SVEUČILIŠTE U RIJECI

FAKULTET INFORMATIKE I DIGITALNIH TEHNOLOGIJA

Saša Sambolek

DETEKCIJA OSOBA NA SLIKAMA BESPILOTNIH LETJELICA U AKCIJAMA TRAGANJA I SPAŠAVANJA U NEURBANIM PODRUČJIMA

DOKTORSKI RAD

Rijeka, 2024.

SVEUČILIŠTE U RIJECI

FAKULTET INFORMATIKE I DIGITALNIH TEHNOLOGIJA

Saša Sambolek

DETEKCIJA OSOBA NA SLIKAMA BESPILOTNIH LETJELICA U AKCIJAMA TRAGANJA I SPAŠAVANJA U NEURBANIM PODRUČJIMA

DOKTORSKI RAD

Mentor: prof. dr. sc. Marina Ivašić-Kos

Rijeka, 2024.

UNIVERSITY OF RIJEKA

FACULTY OF INFORMATICS AND DIGITAL TECHNOLOGIES

Saša Sambolek

PERSON DETECTION IN IMAGES OF UNMANNED AERIAL VEHICLES IN SEARCH AND RESCUE MISSIONS IN NON-URBAN AREAS

DOCTORAL THESIS

Rijeka, 2024.

Mentorica rada: prof. sc. dr. Marina Ivašić-Kos

Doktorski rad obranjen je dana _____ na Fakultetu informatike i digitalnih tehnologija Sveučilišta u Rijeci, pred povjerenstvom u sastavu:

1. _____

2. _____

3. _____

SAŽETAK

Cilj ovog doktorskog rada je razviti inovativni sustav za detekciju osoba na snimkama bespilotnih letjelica u akcijama traganja i spašavanja na neurbanim područjima. Razvijeni sustav testiran je na snimkama bespilotne letjelice neurbanog područja kontinentalnog dijela Hrvatske. Temelji se na uspješnoj implementaciji detektora objekata korištenjem konvolucijskih neuronskih mreža. Za potrebe istraživanja kreiran je namjenski skup označenih slika nazvan SARD. Skup sadrži slike osoba snimljene iz ptičje perspektive na neurbanom području u različitim scenarijima tipičnim za akcije traganja i spašavanja s oznakama osoba. U eksperimentima su korišteni aktualni detektori objekata poput Faster R-CNN, YOLOv4, RetinaNet i Cascade R-CNN te YOLOv8. Nadalje, predložena su dva načina rada: detekcija na snimkama tijekom potrage u realnom vremenu na terenu i analiza prethodno snimljenog materijala. U slučaju naknadne analize snimaka, kako bi zemaljski timovi pristupili traženoj osobi potrebno je znati njenu geolokaciju. Pomoću podataka pohranjenih u slici i koordinate detektirane osobe na slici, predlaže se metoda za određivanje geolokacije tražene osobe. Za osobu koja miruje (najčešći scenarij u akcijama traganja i spašavanja) i koja je detektirana na više slika preporuča se koristiti algoritam mjerenja presjeka, dok u slučaju da se osoba giba ili je detektirana na samo jednoj slici, najbolji rezultati postignuti su korištenjem algoritma koji u obzir uzima i visinu terena na kojem se vrši potraga. Zaključno je prikazan prototip sustava koji sjedinjuje sve navedene cjeline.

Ključne riječi: snimke bespilotnih letjelica, konvolucijske neuronske mreže, prepoznavanje ljudi, YOLO, potraga i spašavanje

ABSTRACT

The goal of this doctoral thesis is to create an advanced system for person detection in drone footage during search and rescue operations in rural areas. The developed system was tested on drone footage of the non-urban area of the continental part of Croatia. It is based on the successful implementation of an object detector using a convolutional neural network. For research purposes, a dataset of images called SARD was created. The set contains bird's-eye images of people in a non-urban area in various scenarios typical of search and rescue operations. Current state-of-the-art object detectors such as Faster R-CNN, YOLOv4, RetinaNet, Cascade R-CNN, and YOLOv8 were used in the experiments. Additionally, two operational modes are suggested: real-time field search and analysis of recorded footage. In the second case, for the ground teams to approach the missed person, it is necessary to know his geolocation. Using metadata stored within the image along with the pixel coordinates of the detected person, we determine the geolocation of the person. When dealing with a stationary person, which is the predominant scenario in search and rescue operations, and is detected across multiple images, it is advisable to employ the intersection measurement algorithm. However, if the person is in motion or is detected in just one image, optimal results are obtained by utilizing an algorithm that factors in the terrain's elevation where the search is conducted. In conclusion, the prototype of the system that unites all the mentioned units is presented.

Keywords: drone imagery, convolutional neural networks, person detection, YOLO, search and rescue

SADRŽAJ

1 UVOD

Hrvatska gorska služba spašavanja (HGSS) bilježi akcije spašavanja tijekom svih godišnjih doba u Hrvatskoj. Akcije zimi najčešće se događaju na planinama, ljeti u nacionalnim parkovima i turističkim mjestima u prirodi, dok su u proljeće i jesen najčešće nesreće penjača i planinara, a sve češće i potrage za izgubljenim sakupljačima šparoga, kestena ili gljiva. Česte su i potrage za starijim stanovništvom, pogotovo za dementnim osobama. Najzahtjevnije potrage su u otežanim vremenskim uvjetima kao što su magla, kiša, snijeg, posebno zimi zbog kratkog dana i hladnoće zbog koje se još dodatno što prije treba pristupiti unesrećenoj ili nestaloj osobi zbog sprječavanja pothlađivanja i što hitnijeg pružanja odgovarajuće zdravstvene pomoći. Statistika pokazuje dramatičan porast intervencija u posljednjih 20 godina, podaci govore da je 1998. godine HGSS intervenirao 96 puta, dok je to u 2018 bilo čak 875 puta. Razlozi uključuju globalne trendove povećane aktivnosti u prirodi, klimatske promjene te sve veći broj posjeta prirodnim ljepotama Hrvatske [1].

Rizične sportske aktivnosti u planinama i prirodi, poput penjanja, često dovode do nesreća s padovima, udarcima i ranjavanjima, dok i pješačke izlete prate rizici poput pokliznuća, iscrpljenosti, dehidracije, gubljenja i slično. Među unesrećenima u planinama i prirodi, većinom su prisutne lakše ozljede poput kontuzija, uganuća, iščašenja te iscrpljenosti, no takve ozljede, bez pravodobne intervencije na nepristupačnom terenu, mogu imati ozbiljne posljedice. Brza intervencija značajno smanjuje štetu po zdravlje i život, skraćuje vrijeme liječenja te pomaže u sprječavanju trajnih invalidnosti. Značajan broj spašenih ljudi bilježi se bez ozljeda, najčešće u situacijama kada se jednostavno izgube na određenom području. Brza i uspješna akcija u potrazi i pronalasku izgubljenih osigurava ne samo spašavanje života i zdravlja, već i sprječava moguće kasnije ozljede ili nesreće.

U svakom slučaju, bez obzira na razlog intervencije, pružanje pomoći i zdravstvene zaštite na nepristupačnim terenima izuzetno je složeno. Posebno kod traganja za nestalom osobom, često je potrebno izvesti zahtjevna pretraživanja velikih i konfiguracijski kompleksnih terena. Važno je naglasiti da je

vrijeme ključni faktor u ovakvim situacijama - kako vrijeme odmiče, vjerojatnost preživljavanja nestale osobe opada, dok se površina koju treba pretražiti eksponencijalno povećava [2]. Akcije traganja i spašavanja zahtijevaju značajan ljudski potencijal i materijalne resurse, uključujući članove gorske službe spašavanja, potražne pse, policiju, zračne snage te sve češće bespilotne letjelice (dronovi).

Bespilotne letjelice su postale standard u većini svjetskih SAR (engl. *Search and Rescue*) službi zbog svoje primjenjivosti u potrazi u urbanim i neurbanim područjima, vodama te u situacijama poput snježnih lavina. Zahvaljujući svojoj kompaktnosti, pokretljivosti, relativno niskoj cijeni i visokoj razlučivosti videozapisa, u stvarnom vremenu omogućuju nadzor većeg područja i prikupljanja informacija o prisutnosti osoba u zoni potrage kao i mogućnost određivanja lokacije osobe koju se traži.

Primjena bespilotnih letjelica značajno povećava vjerojatnost pronalaska osobe tijekom traganja i spašavanja, ubrzavajući proces zahvaljujući brzom pregledu većih površina u jednom letu. Unatoč tomu, udaljeni piloti koji upravljaju bespilotnim letjelicama za vrijeme leta suočavaju se i s izazovima analiziranja snimaka u realnom vremenu na malom ekranu. Osobe za kojima se traga često su male u odnosu na okolinu, zauzimajući samo nekoliko piksela na ekranu, što otežava održavanje dugotrajne koncentracije i pažnje, čak i za obučene udaljene pilote. Nepredvidljive situacije, poput osoba koje su skrivene iza vegetacije ili stijena, dodatno kompliciraju pretragu, posebno u nepovoljnim vremenskim uvjetima poput kiše, magle ili snijega. Gubljenje orijentacije, iscrpljenost, nagla bolest i demencija su česti razlozi nestanka, što dodatno otežava situaciju jer unesrećene osobe često završavaju na neočekivanim mjestima u netipičnim pozama, što može uključivati i ozljede ili neobične položaje.

U operacijama traganja za nestalim osobama, značajnu podršku udaljenom pilotu mogu pružiti metode automatske detekcije osoba. One omogućuju detekciju osoba na snimkama u stvarnom vremenu pružajući informacije o njenoj poziciji, čime doprinose usmjeravanju operacije traganja.

Za detekciju osoba već se uspješno koriste duboke konvolucijske neuronske mreže za detekciju objekata poput Faster-RCNN [3], Cascade R-CNN [4], RetinaNet [5] , SSD [6], YOLOv3 [7], koje postižu visoku točnost na slikama realnih scena poput MS COCO [8], i često nadmašujući ljudske performanse. Ove mreže trenirane su na različitim velikim skupovima podataka poput MS COCO, Pascal VOC [9], ImageNet [10], i postižu izvrsne rezultate u detekciji osoba na sličnim slikama tijekom uobičajenih aktivnosti poput stajanja, hodanja, trčanja ili sjedenja u urbanim scenama.

Kako bi model detekcije postigao što veću točnost, vrlo je važno da skup podataka na kojem se model obučava osigurava slične uvjete onima koji se očekuju prilikom korištenja modela. U akcijama traganja i spašavanja ključan je objekt osoba, međutim kamera je montirana na bespilotnu letjelicu i snimke su iz ptičje perspektive, a takve snimke nisu sadržane u velikim skupovima podataka na kojima su ti modeli obučavani.

Postoje skupovi podataka poput Visdrone [11], Okutama-action [12], UAVDT [13], koji sadrže snimke snimljene bespilotnim letjelicama koje su namijenjene različitim svrhama, uključujući detekciju objekata na slikama i videozapisima, praćenje osoba, prepoznavanje aktivnosti te predviđanje kretanja osoba ili događanja na snimkama. Ipak, ovi skupovi podataka su prilagođeni specifičnim namjenama u urbanim scenama i često ne obuhvaćaju scene koje se pojavljuju u situacijama traganja i spašavanja. Najbliže scenarijima snimljenim bespilotnom letjelicom u traganju i spašavanju su oni koji uključuju ljude u parku dok šetaju ili trče, stoje na trgu, hodaju ulicom ili leže na plaži. Međutim, poze osoba u tim scenama bitno se razlikuju od poza osoba koje se nalaze u situacijama nesreće, gdje su osobe ozlijeđene, iscrpljene ili izgubljene. Iz tog razloga, stvoren je vlastiti skup podataka nazvan SARD [14] koji simulira realne događaje u scenarijima traganja i spašavanja. Slike skupa su označene kako bi korištenjem transfera znanja i finog podešavanja parametara na odabranim arhitekturama dubokih neuronskih mreža, naučeni modeli detektirali osobe u scenama traganja i spašavanja.

Letom na većim visinama skenira se veće potražno područje ali se smanjuje broj piksela koje zauzima osoba na slici/ekranu, također snimke koje se šalju bežično s letjelice na uređaj manje su kvalitete pa je moguće da tražena osoba ne bude detektirana tijekom leta bespilotne letjelice od strane udaljenog pilota i programske podrške za detekciju osoba u stvarnom vremenu. Iz tog razloga je preporučljivo ponoviti traganje/detekciju na snimljenim materijalima, pohranjenim na memorijskoj kartici u letjelici, koji su veće kvalitete što modelu omogućava pouzdanije rezultate.

U slučajevima kada osoba nije detektirana odmah tijekom leta bespilotnom letjelicom već naknadnom pretragom snimljenih materijala, potrebno je odrediti geografske koordinate na kojoj se nalazi detektirana osoba u stvarnom svijetu, kako bi joj na terenu mogli pristupiti zemaljski timovi. Iz dostupnih GPS podataka, metapodataka bespilotne letjelice, karakteristika i pozicije kamere te metapodataka sa snimki bespilotne letjelice određuje se udaljenost od letjelice do detektirane osobe. Iz izračunate udaljenosti predloženi sustav, uz detekciju, određuje geolokaciju te smjer i brzinu kretanja ukoliko je osoba detektirana na više fotografija. Brzina i smjer kretanja važni su kako bi se moglo procijeniti gdje bi se osoba mogla nalaziti u trenutku izlaska na teren. U konačnici sustav predlaže korekciju potražnog područja iz tako dobivenih podataka.

1.1 Cilj i objašnjenje osnovnih pojmova

Cilj istraživanja u okviru doktorskog rada usmjeren je na ispitivanje potencijala metoda dubokog učenja za detekciju i prepoznavanje osoba na snimkama bespilotnih letjelica u operacijama traganja i spašavanja. Proučavanje relevantne literature ukazalo je na puno prostora za napredak u ovom području, posebno u razvoju autonomnih sustava temeljenih na dubokom učenju i neuronskim mrežama, s krajnjim ciljem postizanja pouzdane detekcije i geolokacije traženih osoba.

Osim toga, pregled dostupne literature istaknuo je nedostatak odgovarajućih skupova slika za obuku modela dubokog učenja u detekciji unesrećenih osoba na snimkama iz ptičje perspektive. Iz tog razloga, prepoznata je nužnost

stvaranja novog skupa slika. U sklopu doktorskog rada, stoga je kreiran novi skup slika koje su pripremljene za obuku modela dubokog učenja iz domene traganja i spašavanja.

1.2 Motivacija

Motivacija i znanstvena znatiželja za istraživanjem potencijala metoda dubokog učenja na slikama i videozapisima nastalim tijekom akcija traganja i spašavanja proizlaze iz autorovog osobnog iskustva kao udaljenog pilota bespilotnih letjelica i voditelja potraga u HGSS-u. Stoga, kontinuirano promišljanje dovelo je do ideje o istraživanju primjenjivosti metoda dubokog učenja za detekciju osoba na slikama i videozapisima s bespilotnih letjelica. Implementacija takvih metoda u područje traganja i spašavanja obećava značajan doprinos spašavanju ljudskih života, posebno u izazovnim vremenskim uvjetima i na teškim terenima. Takav sustav značajno bi smanjio troškove, posebice u smislu potrebe za angažmanom ljudskih resursa jer se iz visine može sagledati veća površina terena, a ne predstavlja veliku financijsku investiciju kao kada se za tu svrhu upotrijebi helikopter.

Bespilotne letjelice kao instrumenti u misijama potraga za nestalim osobama značajno proširuju ljudske sposobnosti u tragalačkom pristupu pružajući jedinstveni pogled s visine, omogućuje ljudima nešto što priroda nije pružila - perspektivu iz zraka. Novi izvor informacija i potreba za njihovim što uspješnijim korištenjem snažno je utjecao je razvoj sustava koji korištenjem dostupnih novih tehnologija i metoda dubokog učenja mogu analizirati slike snimljene iz zraka i interpretirati ih.

1.3 Hipoteze i znanstveni doprinosi istraživanja

Kao što je već navedeno, cilj istraživanja u okviru doktorskog rada je ispitivanje mogućnosti primjene modela dubokog učenja za detekciju nestale/ozlijeđene osobe u akcijama traganja i spašavanja uz pomoć bespilotne letjelice u cilju što brže detekcije unesrećene osobe u neurbanom području.

Znanstvene hipoteze su:

H1: Primjena bespilotnih sustava u akcijama traganja i spašavanja doprinosi ranoj detekciji nestalih osoba

H2: Korištenje dubokih neuronskih mreža omogućuje pouzdanu detekciju nestalih osoba u akcijama traganja i spašavanja u neurbanom području

Očekivani znanstveni doprinosi su:

- izrada baze slika i snimaka bespilotnom letjelicom nestalih/ozlijeđenih osoba na neurbanom području pripremljene za obučavanje nadziranog modela strojnog učenja,
- model sustava za detekciju osoba na snimkama snimljenih bespilotnom letjelicom u akcijama traganja i spašavanja,
- metoda za procjenu udaljenosti detektirane osobe od položaja bespilotne letjelice
- prototip sustava za detekciju osoba u akcijama traganja i spašavanja bespilotnim letjelicama.

2 ODABRANI REZULTATI I RASPRAVA

U ovom poglavlju pruža se detaljna rasprava o primijenjenim metodama i postignutim ključnim rezultatima dobivenih istraživanjem. Polazište istraživanja bila je sveobuhvatna analiza potencijala modela dubokog učenja (RAD 1[15]) u kojemu su istraženi izazovi detekcije malih objekata na snimkama iz ptičje perspektive, s fokusom na detekciji igračaka vojnika snimljenih kamerom mobilnog uređaja. Nadalje, obavljen je pregled dosadašnjih radova i skupova podataka koji istražuju slične probleme u detekciji objekata iz zraka ili su fokusirani na primjenu modela dubokog učenja u sličnim scenarijima (RAD 2 [16]). Kreiranjem vlastitog skupa podataka istražila se preciznost i brzina detekcije osoba na snimkama kreiranim pomoću bespilotne letjelice, uz istraživanje robusnosti modela na različite vremenske uvijete i zamućenje uzrokovano gibanjem kamere (RAD 3 [17] i RAD 4 [14]). Primjenom različitih metoda transfera znanja za obučavanje modela dubokog učenja, traži se metoda koja dodatno poboljšava rezultate detekcije osoba (RAD 5 [18]). U konačnici primjenom više algoritama geolokalizacije predlaže se metoda geolokalizacije temeljena na zahtjevnosti potražnog terena i broju detekcija detektirane osobe na slikama snimljenim pomoću bespilotne letjelice korištenjem samo resursa dostupnih u offline načinu rada na neurbanom terenu ([19], RAD 6 [20]).

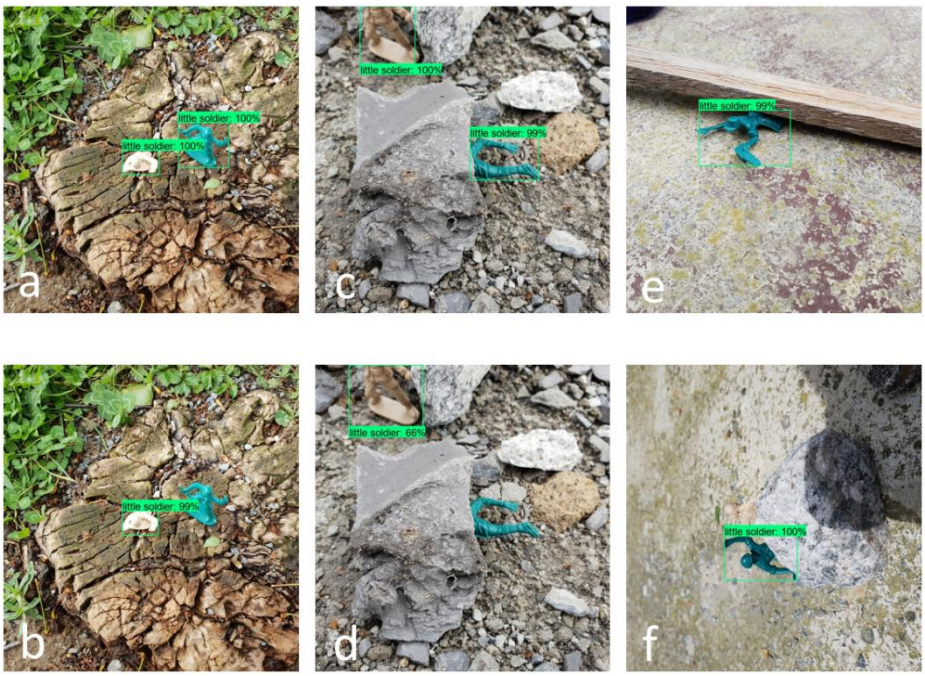
Kao cjelina, objavljeni članci su bili međusobno povezani, ističući usklađenost i doprinos dodatne vrijednosti ukupnom konceptu.

2.1 Konvolucijske neuronske mreže u akcijama traganja i spašavanja

Konvolucijska neuronska mreža (CNN), čija je arhitektura predložena od strane Yanna LeCuna [21], predstavlja specifičnu arhitekturu umjetnih neuronskih mreža. Praktična prednost takve arhitekture leži u smanjenju broja parametara koje koristi CNN u usporedbi s potpuno povezanim neuronskim mrežama, što značajno poboljšava vrijeme obučavanja i smanjuje potrebne podatke za treniranje modela. Nakon što je AlexNet [22] popularizirao duboke neuronske mreže pobjedom na ImageNet natjecanju, duboke konvolucijske neuronske mreže postale su najpopularnija vrsta neuronske mreže za klasifikaciju slika i

probleme detekcije objekata. Modeli računalnog vida koriste se za razne zadatke obrade slike, uključujući klasifikaciju slika, detekciju objekata, segmentaciju slika, generiranje slika i još mnogo toga. Ti modeli često koriste CNN arhitekture i treniraju se na velikim skupovima podataka kako bi naučili prepoznavati obrasce u slikama. Detektori objekata kao vrsta modela računalnog vida specifično su dizajnirani algoritmi za lokalizaciju i prepoznavanje objekta unutar slike ili video zapisa.

Na temelju provedenog pregleda literature i analiziranog potencijala mreža, odabrane su duboke mreže ResNet50 [23], Inception [24] i MobileNet [25] u kombinaciji s detektorima SSD i Faster R-CNN. Odabrani modeli vrednovani su na vlastitom skupu podataka koji simulira različite konfiguracije scena neurbanih područja, složenosti i uvjeta osvjetljenja kao i broja objekata i njihovog položaja (Slika 2.1). U skupu se nalaze snimke igračaka vojnika snimljenih mobilnim telefonom iz ptičje perspektive.



Slika 2.1 Rezultati detekcije igračaka vojnika različitim modelima na složenim scenama skupa podataka [15]

Glavni cilj RAD-a 1 bio je provjeriti jesu li duboke konvolucijske neuronske mreže prikladne za detekciju objekata iz ptičje perspektive. Analiza je pokazala da je

Faster R-CNN najprikladniji za detekciju, dok je problem bio što je ovaj model zahtijevao više vremena i računalne snage u odnosu na ostale testirane modele. Rezultati istraživanja dali su temelje za daljnje istraživanje detekcije nestalih osoba u stvarnom vremenu u akcijama traganja i spašavanja.

RAD 2 daje analizu mogućnosti korištenja bespilotnih letjelica u akcijama traganja i spašavanja te cjeloviti pregled područja vezanog za detekciju osoba na snimkama bespilotnom letjelicom. Također rad je dao opis javno dostupnih skupova podataka te usporedbu najsuvremenijih modela za detekciju osoba na snimkama iz zraka. Zaključak ovoga rada je da je potrebno napraviti namjenski skup podataka snimljen bespilotnom letjelicom prilagođen danom zadatku. Takav skup sadržavao bi osobe u položajima tipičnim za unesrećene ili iznemogle osobe u akcijama traganja i spašavanja koji nisu sadržani u do tada objavljenim skupovima podataka. Također, potrebno je kombinirajući znanja iz postojećih skupova podataka s novim skupom podataka testirati najsuvremenije modele.

2.2 Prikupljanje podataka

2.2.1 SARD skup podataka

U svrhu treniranja i testiranja modela za automatsku detekciju unesrećenih osoba na snimkama i videozapisima snimljenim iz zraka, formirali smo našu bazu podataka nazvanu SARD, koja vjerno simulira stvarne događaje u scenarijima traganja i spašavanja. SARD baza obuhvaća raznovrsne situacije koje uključuju iscrpljene i ozlijeđene osobe, te tipična kretanja ljudi u prirodnom okruženju, kao što su trčanje, hodanje, stajanje, sjedenje ili ležanje. Kako bi uključivala različite vrste terena i pozadinskih elemenata koji mogu utjecati na događaje i scenarije na snimljenim slikama i videozapisima, u SARD bazu su uključene snimke koje obuhvaćaju scene u kojima su osobe smještene na makadamskim putevima, u kamenolomima, niskoj i visokoj travi, sjeni šume i slično.

Snimanje je provedeno tijekom dana, u jesen, pomoću kamere na bespilotnoj letjelici DJI Phantom 4A. Video zapisi su snimljeni u FHD rezoluciji od 1920 x 1080 piksela pri frekvenciji od 50 slika u sekundi. Letjelica je letjela na različitim visinama, varirajući od 5 m do 50 m, s različitim kutovima kamere u rasponu od

45° do 90°. Sve snimke su nastale na području Moslavačke gore, izvan urbanog područja.

Položaji osoba na snimkama obuhvaćaju uobičajne položaje i posture (stojeći, sjedeći, ležeći, hodanje, trčanje) te položaje karakteristične za iscrpljene ili ozlijeđene osobe, rekonstruirane od strane statista prema njihovom vlastitom nahođenju (Slika 2.2). Statisti su bili devet osoba različite životne dobi i spola, od 7 do 55 godina. Osobe su također smještene na različitim lokacijama, od jasno vidljivih do manje uočljivih (oku) lokacija, u šumi, visokoj travi, u sjeni i slično, što dodatno otežava detekciju.

Slika 2.2 Neki od položaja osoba za kojima se traga, slike su izrezane iz skupa podataka snimljenih bespilotnom letjelicom. [14]

Iz snimki ukupne duljine oko 35 min izdvojili smo 1.981 pojedinačni kadar na kojemu se nalaze osobe.

Na odabranim slikama smo ručno označili prisutne osobe kako bismo formirali skup podataka koji će poslužiti za treniranje modela. Ukupno je označeno 6.532 objekata za klasu osoba.

Dimenzije okvira u skupu SARD kreću se od 7 px za najmanju širinu i 8 px za najmanju visinu dok je najveća širina 353 px a najveća visina 337 px. Površinom najmanji označeni objekt je 7 px x12 px dok je najveći 322 px x 231 px, u prosjeku veličina okvir je 47px x 58 px. U skupu je označeno 1883 malih objekata (objekti čija je površina graničnog okvira manja od 32^2), 4180 srednjih objekata (32^2 < površina < 96^2) i 1981 velikih objekata (površina > 96^2)

Skup SARD podijeljen je na skup za obučavanje (train) i skup za testiranje (val) u omjeru 60:40 na način da su slike jednoliko raspodijeljene prema scenama (pozadina, osvjetljenje, poza osoba, kut kamere). Skup za treniranje sadrži 1189 slika, na kojima je označeno 3921 osoba, dok skup za testiranje sadrži 792 slike na kojima je označeno 2611 osoba.

2.2.2 VisDrone skup podataka

VisDrone je skup podataka snimljen bespilotnim letjelicama u različitim scenama usredotočen na četiri osnovna problema u području računalnog vida (otkrivanje objekata u slikama, otkrivanje predmeta u videozapisima, praćenje pojedinačnih objekata i praćenje više objekata).

Skup podataka sastoji se od 263 video isječka i dodatnih 10.209 slika. Videozapisi / slike snimljeni su na različitim platformama bespilotnih letjelica (DJI Mavic, DJI Phantom Series 3, 3A, 3SE, 3P, 4, 4A, 4P) u 14 različitih gradova u Kini. Skup podataka pokriva različite vremenske i svjetlosne uvjete maksimalne razlučivosti videozapisa (3840 x 2160 px) i slike (2000 x 1500 px).

U preuzetom VisDrone skupu (train, val i test) nalazi se 147.747 oznaka koje predstavljaju osobu (osoba/pješak) na 7.482 slike, od toga 125 998 malih, 21.221

srednjih i 528 velikih. Iz VisDrone skupa podataka odabrano je 2.000 slika na kojima se nalaze osobe, objedinivši oznake iz skupa pripadnih anotacija koje se odnose na osobu ili pješaka na jednu klasu: osoba. Na odabranim slikama je označeno 30.641 malih objekata(površina < 32^2), 6.384 srednjih objekata (32^2 < površina < 96^2) i samo 101 veliki objekt (površina > 96^2). Iz statistike skupa vidimo da su snimke u VisDrone skupu podataka napravljene na većim visinama u odnosu na SARD skup.

Skup smo podijelili na skup za obuku koji se sastoji od 1598 slika s 29.797 označenih osoba i na skup za testiranje koji sadrži 402 slike s 7.329 osoba.

VisDrone skup korišten je za istraživanje metoda transfera znanja kako bi se poboljšala detekcija osoba na slikama snimljenim bespilotnim letjelicama za potrebe operacija traganja i spašavanja

2.3 Odabir modela za detekciju

U provedenim istraživanjima testirali smo performanse postojećih jednofaznih i dvofaznih suvremenih detektora (Faster R-CNN, YOLOv4 [26], RetinaNet, i Cascade R-CNN), kako bismo odabrali model koji daje najbolje rezultate na našem skupu podataka i kojeg ćemo dalje koristiti u eksperimentima na scenama traganja i spašavanja.

U eksperimentima tijekom izrade doktorskog rada koristimo nekoliko standardnih metrika za procjenu performansi detektora i metrika koje smo namjenski razvili za detekciju i geolociranje u akcijama traganja i spašavanja, kao što je objašnjeno u nastavku.

Omjer presjeka nad unijom (engl. *Intersection over Union*, skraćeno IoU) tradicionalna je metrika za procjenu performansi detektora koja provjerava detekciju i performanse procjenom odnosa površine preklapanja između predviđenih i referentnih oznaka (engl. *ground truth*) i ukupnom površinom obje oznake. Jednadžba je sljedeća:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \qquad 2.1$$

Više vrijednosti za IoU ukazuju na bolje preklapanje između detekcije i stvarnih podataka.

Odziv (engl. *Racall*) (R) i preciznost (engl. *Precision*) (P) izračunavaju se kao:

$$P = \frac{TP}{TP + FP}, \qquad R = \frac{TP}{TP + FN} \tag{2.2}$$

gdje je TP (engl. *True Positive*) pozitivna detekcija tj. detekcija koja je točna, FP (engl. *False Positive*) je lažno pozitivna detekcija gdje su detektirana područja slike koja ne sadrže osobu, a FN (engl. *False Negative*) je lažno negativna detekcija, tj. ovi objekti postoje na slici ali ih detektor nije detektirao. Preciznost je metrika koja mjeri točnost pozitivnih detekcija, tj. koliko je predviđenih pozitivnih detekcija bilo točno, dok je odziv metrika koja mjeri sposobnost modela da detektira sve relevantne instance u skupu.

Prosječna preciznost (AP) je uobičajena metrika procjene u otkrivanju objekata. AP mjeri prosječnu preciznost u rasponu IoU od 0,5 do 0,95, s intervalima od 0,05. AP uzima u obzir preciznost svake detekcije koja ima preklapanje s referentnom oznakom veće ili jednako od određenog prag IoU (npr. 0,5 ili 0,75), te računa prosječnu vrijednost tih preciznosti kako bi dao cjelokupnu ocjenu performansi modela. U našem slučaju ovu metriku promatramo samo kroz jednu klasu (osoba). U eksperimentu koristimo AP_{50}. To znači da se za procjenu koristi preciznost detekcija koje imaju preklapanje s referentnim oznakama veće ili jednako 0,5, tj. uzimaju se u obzir samo one detekcije koje imaju dovoljno preklapanje s referentnim oznakama kako bi se smatrale relevantnim. Također u nekim eksperimentima prosječnu preciznost promatramo u odnosu na veličinu koju objekt zauzima na slici. Prosječnu preciznost kroz različite veličine objekata prikazujemo kao AP_S za male objekte čija je površina manja od 32^2, AP_M za srednje objekte čija je površina između 32^2 i 96^2, dok je prosječna preciznost za velike objekte dana kao AP_L za objekte čija je površina veća od 96^2.

U akcijama traganja i spašavanja cilj je detektirati sve osobe prisutne na sceni, ali s druge strane važna je i preciznost detektora da se nepotrebno ne troše resursi na lažne detekcije. Iz tog razloga, na temelju postignutih rezultata prosječne preciznosti te odnosa preciznosti i odziva odabran je YOLOv4 detektor

za daljnje istraživanje obzirom da postiže najveću srednju preciznost i uspijeva detektirati najveći broj objekata na slici uz najvišu preciznost. Iz istog razloga predložena je i mjera nazvana ROpti [14] koja se računa kao omjer razlike između stvarnih (TP) i lažnih pozitivnih (FP) detekcija te mogućih detekcija (TP + FN) u skupu podataka:

$$ROpti = \frac{TP - FP}{TP + FN} \qquad 2.3$$

Ova mjera pruža kvantitativnu procjenu performansi modela u smislu smanjenja lažno pozitivnih detekcija i favoriziranju stvarno pozitivnih detekcija, uzimajući u obzir sve moguće detekcije u skupu podataka. Za savršenu preciznost (bez lažno pozitivnih), ROpti je jednako odzivu, a za savršen odziv (bez lažno negativnih), ROpti iznosi 1, što predstavlja savršen rezultat.

Tablica 2.1 Rezultati detekcije osoba modelima obučenim na SARD skupu podataka [14]

Model	**AP**	**AP_{50}**	**AP_S**	**AP_M**	**AP_L**
Cascade R-CNN	0,490	0,881	**0,310**	0,544	0,626
Faster R-CNN	0,501	**0,907**	0,305	0,560	0,650
RetinaNet	0,339	0,733	0,129	0,406	0,531
YOLOv4	**0,530**	0,903	0,295	**0,596**	**0,740**

Rezultata testiranja CNN modela dani su u Tablica 2.1. Najbolji rezultati dobiveni su sa YOLOv4, dok rezultati Cascade R-CNN i Faster R-CNN detektora jako malo zaostaju. Svi detektori najbolje rezultate postižu u slučaju AP_{50} s tim da najbolje rezultate od preko 90% postižu Faster R-CNN i YOLOv4. Što je za detektor kakav je potreban za akcije traganja i spašavanja odličan rezultat iz razloga što za pronalazak osobe nije bitan koliko je veliko preklapanje detekcije u odnosu na referentnu oznaku. U stvarnoj situaciji ako detektiramo i samo nogu osobe, osoba je pronađena.

Primjeri detekcija osobe s treniranim modelima prikazani su na Slika 2.3. Stupci na Slika 2.3 predstavljaju rezultate detekcije i to redom u stupcu A) Cascade R-

CNN(SARD) modela, stupac B) Faster R-CNN(SARD) modela, C) RetinaNet (SARD) i u D) YOLOv4(SARD). Na slikama se pojavljuju svi slučajevi detekcije: pozitivna (detektirana je osoba i granični okvir uključuje više od 50% referentne oznake osobe, negativne (osoba nije detektirana) i lažne pozitivne detekcije (detektor je označio kao osoba dio slike koji ne sadrži osobu).

A) B) C) D)

Slika 2.3 Primjeri detekcije različitih modela: A stupac: Cascade R-CNN(SARD), B stupac: Faster R-CNN(SARD), C stupac: RetinaNet(SARD), D stupac: YOLOv4(SARD) [14].

Prvi red na Slika 2.3 prikazuje slučaj u kamenolomu, jedna osoba se nalazi na hrpi kamenja dok su dvije osobe na prašnjavom putu. Svi detektori uspješno su detektirali osobe na putu, dok su samo Cascade R-CNN(SARD) i YOLOv4(SARD) detektirali i osobu koja sjedi na kamenju. Faster R-CNN(SARD) ima jednu lažnu detekciju i višestruku detekciju osobe na putu.

U drugom redu prikazan je primjer tri osobe s preklapanjem (okulzijom) koje se nalaze na niskoj travi. Osobu desno gore, koja stoji uspješno su detektirali svi detektori. Faster R-CNN(SARD) i u ovom primjeru daje višestruke detekcije osoba koje se preklapaju, dok Cascade R-CNN(SARD) i Retinanet(SARD) imaju

problema s okluzijom i nisu detektirali osobu koja kleči iza osobe koja se kreće. YOLOv4(SARD) uspješno je detektirao sve osobe.

U trećoj sceni snimljenoj sa veće visini u odnosu na prva dva primjera, nalazi se osam osoba. Cascade R-CNN(SARD) detektirao je sedam osoba uz jednu lažnu detekciju. Faster R-CNN(SARD) ima pet točnih detekcija kao i Retinanet(SARD) koji uz to ima i tri lažne detekcije. YOLOv4(SARD) točno je detektirao sve osobe na slici.

U zadnjem slučaju snimljenom sa još veće visine i udaljenosti od objekta u visokoj travi i makadamskom putu nalazi se devetero osoba. Cascade R-CNN(SARD i Faster R-CNN(SARD) točno su detektirali sedmero osoba dok je Retinanet(SARD) detektirao točno njih pet. YOLOv4(SARD) uspješno je detektirao sve osobe na slici.

Iz kvalitativne analize odabranih primjera jasno se pokazuje da je YOLOv4(SARD) bio najuspješniji u detekciju osoba. Međutim postoje i primjeri na kojima YOLOv4(SARD) model nije bio uspješan (Slika 2.4). Najčešći primjeri su detekcija dvije osobe koje stoje jako blizu jedna drugoj ili koje se preklapaju kao jedna osoba (prvi red na slici) i lažne detekcije kada detektor detektira tamnije dijelove vegetacije (drugi red) ili sjene (treći red) kao osobu. Postoje i primjeri koji su nažalost česti u akcijama traganja i spašavanja u kojima je osobu jako teško detektirati čak i istreniranom oku jer se stopila s pozadinom, i detektor je nije uspio detektirati (Slika 2.4, treći red).

Slika 2.4 Pogrešne detekcije YOLOv4(SARD) modela [14].

2.3.1 Rezultati detekcije YOLOv4(SARD) detektora ovisno u ulaznoj veličini mreže

Arhitektura YOLO prilagođava veličinu ulazne slike, čuvajući omjer širine i visine prema rezoluciji definiranoj u .cfg datoteci s težinama, određenoj parametrima „width“ i „height“. Ovi parametri nazivaju se rezolucijom mreže. Transformacija rezolucije ulazne slike u YOLO arhitekturi definirana je kao:

$$Img_{train\_width} = Net_{width},$$
$$Img_{train\_height} = \frac{Net_{width}}{Img_{width}} Img_{height} \quad (2.4)$$

Primjerice, ako je rezolucija ulazne slike 1920 x 1080, a rezolucija mreže je definirana kao širina, Net_{width} = 512, i visina, Net_{height} = 512, YOLO će promijeniti rezoluciju ulazne slike na postavljenu širinu, Net_{width}, čuvajući izvorni omjer

između širine slike, Img_{width}, i visine, Img_{height}. Drugim riječima, 1920 x 1080 će se transformirati u 512 x 288.

Da bi se poboljšala performansa detektora, posebno za detekciju malih objekata, jedna od alternativa bila je korištenje veće rezolucije ulaznih slika i treniranje mreže na većim rezolucijama:

$$Net_{\overline{width}} = Net_{width} + k, k = 32n, n \in \mathbb{N} \quad (2.5)$$

Vrijednosti rezolucije mreže mogu biti višekratnici broja 32, na primjer: 608 x 608 ili 832 x 832, jer YOLO mreža smanjuje ulaznu sliku za faktor 32.

U našem slučaju, YOLOv4 (SARD) model je treniran na rezoluciji mreže od 512 x 512, a naše računalo nije bilo dovoljno snažno da trenira mrežu na višim rezolucijama od te. Stoga smo kao alternativu koristili povećanje rezolucije mreže tijekom testiranja [27].

Kako bismo ispitivali utjecaj promjene rezolucije mreže tijekom testiranja performansi detekcije objekata, testirali smo različite rezolucije mreže ispod i iznad rezolucije na kojoj je model bio treniran: 320 x 320, 416 x 416, 512 x 512, 608 x 608, 832 x 832, 1024 x 1024. Rezolucije mreže od 320 x 320 i 416 x 416 su ispod rezolucije na kojoj je YOLOv4 (SARD) model treniran, dok su rezolucije 608 x 608, 832 x 832 i 1024 x 1024 iznad.

Najbolji rezultat postignut je pri rezoluciji mreže od 832 x 832, što se vidi iz Tablica 2.2. Usporedba rezultata pokazuje da se bolji rezultati detekcije mogu postići povećanjem rezolucije mreže tijekom testiranja. Ovdje su postignuti bolji rezultati pri rezoluciji od 608 x 608 i 1024 x 1024 u usporedbi s rezolucijom od 512 x 512 na kojoj je model treniran. Međutim, rezultati također pokazuju da postoji granica nakon koje se rezultati više ne poboljšavaju, kao u slučaju rezolucije mreže od 1024 x 1024, kada su rezultati počeli opadati.

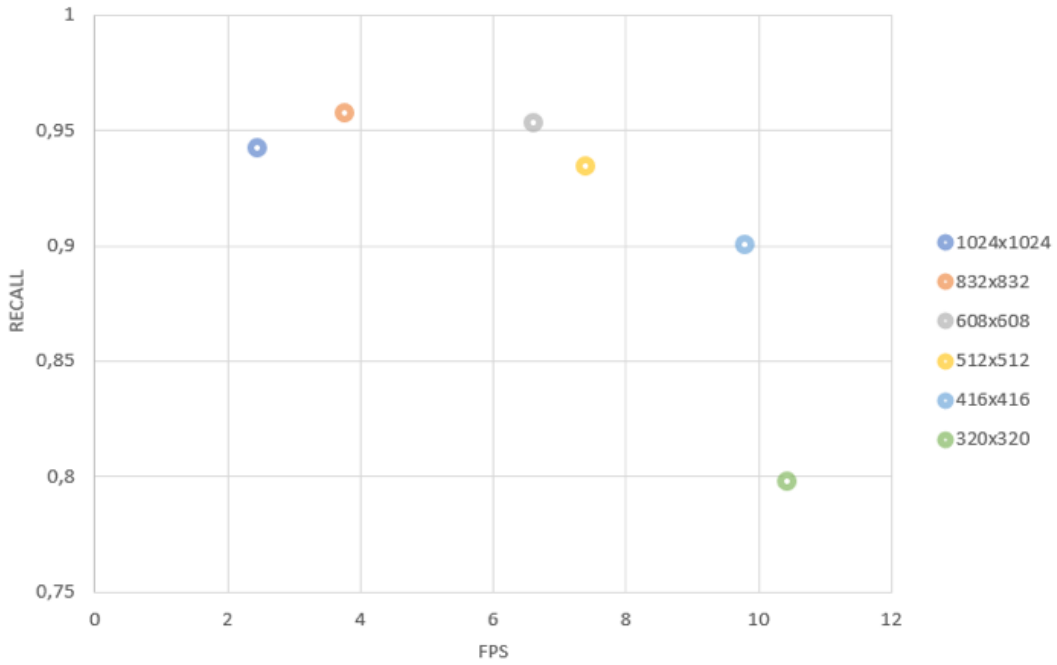
Tablica 2.2 Rezultati detekcije YOLOv4(SARD) detektora ovisno o ulaznoj veličini mreže [14].

Rezolucija mreže	**AP**	**AP_{50}**	**AP_{75}**	**AP_S**	**AP_M**	**AP_L**	**fps**
320x320	0,376	0,764	0,327	0,105	0,457	0,706	**10,43**
416x416	0,503	0,882	0,519	0,247	0,581	0,735	9,79
512x512	0,559	0,915	0,626	0,331	0,627	**0,748**	7,39
608x608	0,581	0,937	0,653	0,382	0,642	0,740	6,64
832x832	**0,597**	**0,948**	**0,680**	**0,443**	**0,646**	0,698	**3,76**
1024x1024	0,572	0,937	0,649	0,436	0,618	0,642	2,46

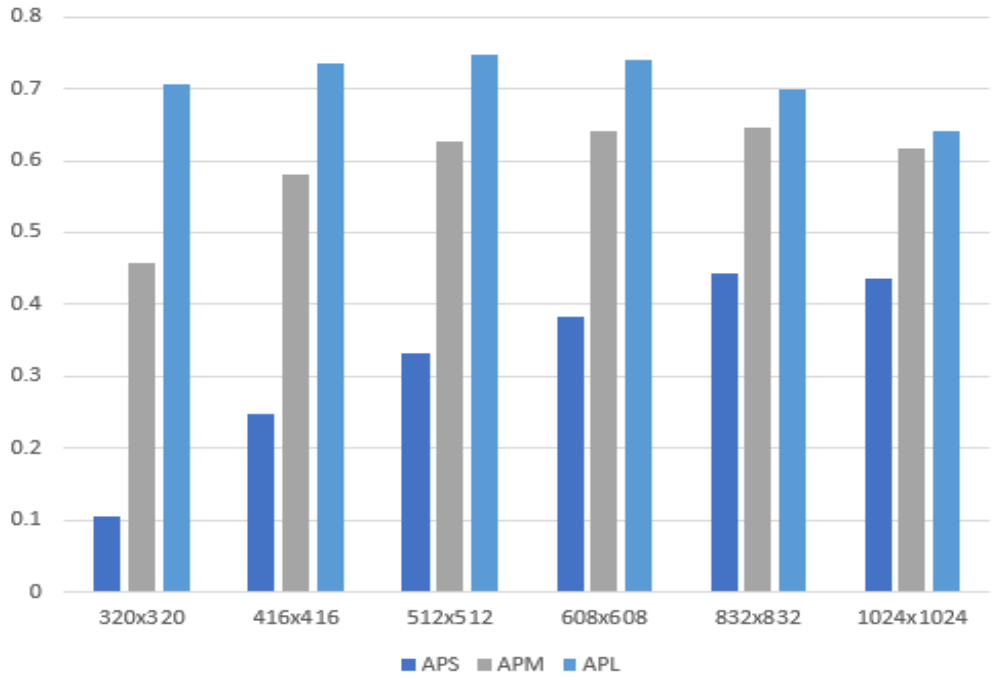
U slučaju testiranja na nižim rezolucijama od rezolucije mreže na kojoj je model treniran, općenito se dobivaju lošiji rezultati, osim u slučaju velikih objekata gdje se postižu samo blago lošiji rezultati. Također, primijećeno je da je brzina zaključivanja otprilike 10 sličica u sekundi (fps) za najnižu rezoluciju mreže, što je 2,5 puta brže nego pri rezoluciji od 832 x 832, gdje se postižu najpreciznji rezultati.

Brzina detekcije važna je za detekcije uživo tijekom leta, pogotovo u operacijama traganja i spašavanja, ali jednako važna je i točnost detekcije i odziv. Stoga je prikazan usporedni graf odnosa između odziva i brzine detekcije za različite rezolucije mreže na Slika 2.5. Na temelju testnih podataka i uzimajući u obzir odnos između brzine detekcije i odziva, odabrali smo rezoluciju mreže od 832 x 832 za daljnja istraživanja.

Slika 2.6 prikazuje rezultate detekcije YOLOv4 (SARD) modela kao ovisnost prosječne preciznosti kroz različite veličine objekata (mali, srednji i veliki) u odnosu na rezoluciju mreže. Najbolja prosječna preciznost od 75% postignuta je s rezolucijom od 512 x 512 piksela za velike objekte, dok su za srednje i male objekte najbolje prosječne preciznosti 44% odnosno 65%, postignute s rezolucijom mreže od 832 x 832 piksela.



Slika 2.5 Odnos između odziva i brzine detekcije za različite veličine mreže [14].



Slika 2.6 Preciznosti YOLOv4 (SARD) detektora kroz različite veličine objekata za različite rezolucije ulazne mreže [14].

Mali objekti s rezolucijom mreže od 832 x 832 piksela postižu čak 6% bolje rezultate u usporedbi s rezolucijom od 608 x 608 piksela. U slučaju srednje velikih

objekata, rezultati su usporedivi, ali s nižom rezolucijom postiže se čak dvostruka brzina detekcije. S obzirom da u našem skupu većinom postoje mali objekti, a obrada može biti i nakon leta na računalima veće računalne snage, rezolucija 832x832 odabrana je kao najprikladnija za naš zadatak detekcije ljudi u scenama traganja i spašavanja.

2.3.2 Robusnost modela na zamućenost zbog gibanja i vremenske uvijete

Kako bi se povećala robusnost modela, korištenjem skupa podataka SARD, računalno je generiran novi skup koji smo nazvali Corr. Skup podataka Corr obuhvaća slike koje simuliraju različite vremenske uvjete (koji su dodani na postojeće slike iz SARD skupa podataka) prisutne u stvarnim scenarijima potrage i spašavanja, kao što su magla, snijeg i led. Također, zamućene slike uključene su u Corr set za simulaciju kretanja kamere pri snimanju.

Testiranje YOLOv4 modela s rezolucijom od 832 x 832, pragom (engl. *threshold*) od 0,25 i IoU od 0,50 izvršeno je na različitim vremenskim uvjetima (snijeg, magla, led) i slikama s zamućenjem zbog pokreta (engl. *motion blur*) na skupu podataka Corr . Korištenjem slika iz SARD testnog skupa, stvorili smo nove skupove u četiri navedene kategorije. Također su iz skupa uklonjene slike ljudi u šumi, što je vrlo teško detektirati. Svaki testni skup za procjenu robusnosti modela za određenu kategoriju sadrži 714 slika.

Rezultati ispitivanja navedeni su u

Tablica 2.3. s obzirom na prosječnu preciznost (AP) i prosječni odziv (AR), uzimajući u obzir preciznost preklapanja objekata (IoU) i veličinu objekata. Rezultati pokazuju nekoliko važnih činjenica.

Značajno smanjenje performansi detekcije dogodio se u slučaju testiranja na slikama s lošim vremenskim uvjetima i zamućenim slikama koje nisu korištene u skupu za treniranje. Na primjer, smanjenje AP_{50} bilo je od 0,948 na 0,59 za snijeg, 0,55 za maglu, 0,63 za led, i 0,67 za zamućenje. Kategorija navedena u Tablici 2.3 predstavlja skup za testiranje, tj. val SARD skup podataka korišten za testiranje, snijeg je isti taj skup u kojem je pomoću računala na slike dodan snijeg

itd. Druga kolona u istoj tablici predstavlja redoslijed skupova podataka na kojima je model za detekciju treniran. Izvorni YOLOv4 model treniran je na COCO skupu podataka i on je označen kao COCO. COCO + SARD predstavlja model YOLOv4 koji je treniran na COCO skupu podatak i nakon toga na SARD skupu podataka.

Nakon što je Corr skup koji sadrži slike s lošim vremenskim uvjetima i zamućenjem korišten za treniranje modela (COCO + SARD + Corr), postignuti su vrlo dobri rezultati za sve kategorije vremenskih uvjeta, slični rezultatima kada su korištene originalne slike iz SARD skupa.

Tablica 2.3 Rezultati detekcije za različite kategorije val skupa s detektorima obučenim na COCO, COCO + SARD i COCO + SARD + Corr skupovima podataka [14].

Kategorija	**Treniran na COCO+**	**AP**	**AP_{50}**	**AP_S**	**AP_M**	**AP_L**	**AR_S**	**AR_M**	**AR_L**
val		0,232	0,402	0,132	0,261	0,413	0,160	0,289	0,473
snijeg	**SARD**	0,325	0,590	0,180	0,358	0,569	0,209	0,398	0,613
magla	**SARD**	0,302	0,550	0,225	0,322	0,404	0,263	0,361	0,457
led	**SARD**	0,359	0,629	0,225	0,393	0,508	0,269	0,433	0,549
zamućenje	**SARD**	0,316	0,678	0,147	0,351	0,581	0,194	0,406	0,628
val	**SARD**	0,597	0,948	0,433	0,646	0,698	0,511	0,703	0,741
snijeg	**SARD + Corr**	0,503	0,885	0,334	0,547	0,651	0,413	0,607	0,700
magla	**SARD + Corr**	0,547	0,916	0,382	0,595	0,653	0,461	0,655	0,698
led	**SARD + Corr**	0,531	0,905	0,367	0,575	0,665	0,443	0,636	0,714
zamućenje	**SARD + Corr**	0,439	0,849	0,244	0,494	0,616	0,320	0,557	0,667
val	**SARD + Corr**	0,555	0,916	0,370	0,615	0,677	0,444	0,675	0,722

Općenito, postignuti rezultati detektora dodatno treniranog na skupu s uključenim lošim vremenskim uvjetima i zamućenjem slike lošiji su na skupu za testiranje po vedrom vremenu, nego u slučaju kada ove transformacije nisu primijenjene, ali

postignuti rezultati daju nam pravo predložiti ovaj model kao pomoć u operacijama traganja i spašavanja zbog znatno poboljšanih rezultata u uvjetima lošijeg vremena. Primjeri rezultata detekcije ovog modela u svim četiri kategorije prikazani su na Slika 2.7.

Slika 2.7 Primjer detekcije YOLOv4 modela obučenog na COCO, SARD i Corr skupu podataka. Gore-lijevo snijeg, gore-desno magla, dolje lijevo led, dolje desno zamućenje zbog pomaka kamere [14].

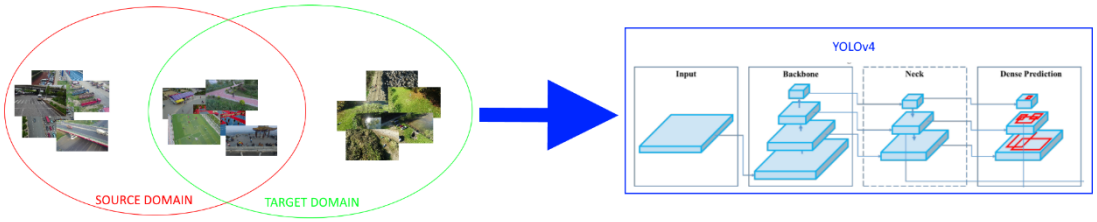
2.3.3 Različite metode transfera znanja

U RADU 5. proučavane su metode transfera znanja kako bi se poboljšala detekcija osoba na slikama snimljenim bespilotnim letjelicama za potrebe operacija traganja i spašavanja. Prilagodili smo YOLOv4 model korištenjem različitih metoda transfera znanja na tri skupa podataka: skup SARD za misije traganja i spašavanja, skup snimaka bespilotnom letjelicom VisDrone u urbanim područjima i skup podataka Corr s sintetski dodanim vremenskim efektima na slikama iz SARD skupa podataka.

Rezultati istraživanja ukazuju na to da se optimalni rezultati detekcije postižu na ciljanom SARD području primjenom mrežnog transfera znanja, kada je skup na kojem se model fino podešava jednako distribuiran kao i skup za testiranje.

Najuspješniji rezultati dobiveni su korištenjem metode mrežnog transfera znanja, koja prenosi značajke naučene na velikim skupovima podataka, te metode transfera znanja zasnovane na instancama, gdje je model treniran na slikama domene koje odgovaraju slikama na kojima će se model testirati (Slika 2.8). Dodatna upotreba sintetičkih instanci slika dodatno je unaprijedila performanse modela.

Također, primijećeno je da su najlošiji rezultati postignuti spajanjem skupova podataka, budući da se u tom slučaju model nije mogao potpuno prilagoditi relevantnim podacima. Unatoč tome, ovakvim spajanjem i povećanjem broja podataka za obučavanje moguće je postići općenitiji model. Osim toga, pokazalo se da redoslijed i način treniranja modela s više skupova podataka nisu zanemarivi faktori.

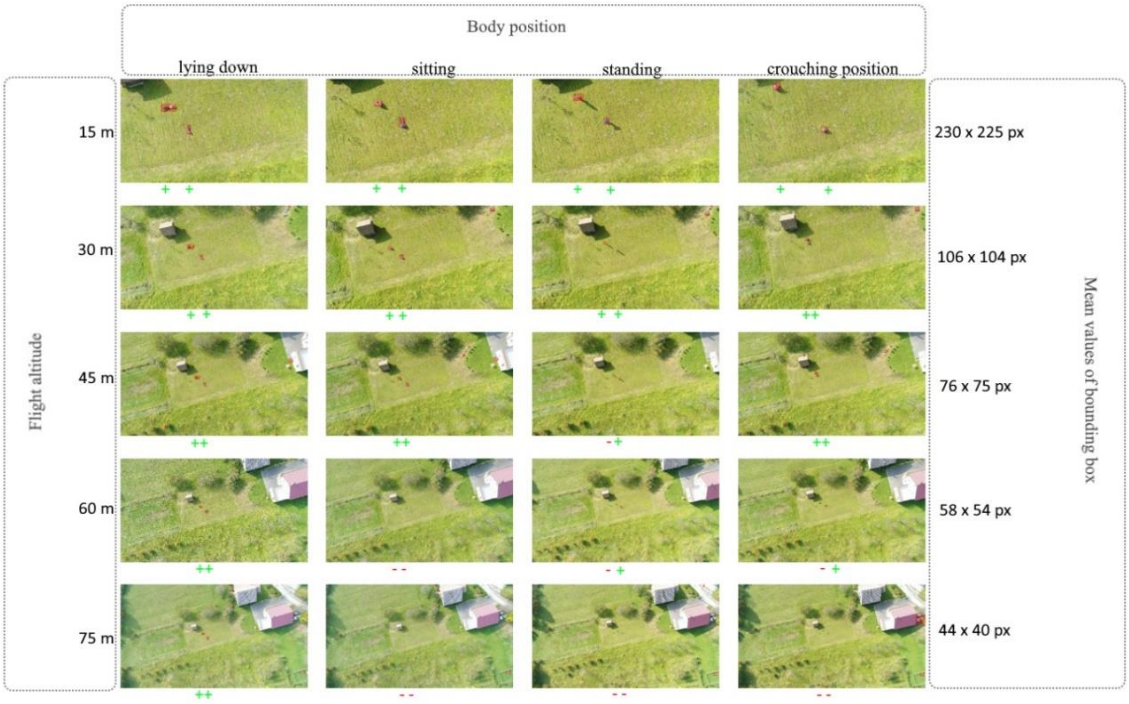


Slika 2.8 Transfer znanja temeljen na instancama. Odabrane su samo relevantne slike za našu domenu na kojima je model treniran. U drugom koraku model se trenira na slikama ciljane domene [18].

2.3.4 Ovisnost detekcije o udaljenosti bespilotne letjelice od tla

Za detekciju osoba na fotografijama iz zraka, visina na kojoj se letjelica nalazi igra ključnu ulogu. Let na većoj visini omogućuje brže prekrivanje područja pretraživanja, dok let na manjoj visini omogućava lakšu detekciju osoba. Najbolja visina leta letjelice ovisi o broju piksela kamere i korištenim lećama. Osoba je prikazana na slikama pomoću piksela, i što je veći broj piksela, osoba je lakša za detektiranje. S DJI Phantom 4 Advanced snimamo slike rezolucije 5472 x 3078

piksela s kutom kamere od 90°, a vidno polje (FOV) prema specifikaciji iznosi 84°. U eksperimentu smo snimili slike dviju osoba (žene i dječaka od osam godina) na različitim visinama (15 m, 30 m, 45 m, 60 m i 75 m). Slika 2.9 prikazuje rezultate detekcije. Vidimo da na visini od 30 m i dalje imamo sve točne detekcije, zbog čega predlažemo da bespilotna letjelica leti na visini na kojoj osoba na slici zauzima okvir dimenzija 100 x 100 piksela.



Slika 2.9 Ovisnost detekcije o položaju tijela osobe i visini snimanja [14].

2.3.5 YOLOv8 detektor

U RAD-u 6. testirane su performanse zadnjeg YOLOv8 modela kroz sve verzije. Ovaj model treniran na manjem dijelu SARD podataka postiže performanse jednake ranijem modelu uz puno veću brzinu obrade podataka.

Tablica 2.4 Performanse pet verzija modela YOLOv8, dodatno treniranih na SARD skupu podataka, pri čemu su najbolji rezultati istaknuti podebljano. Ova provjera izvedena je korištenjem Google Colab platforme [20].

Verzija YOLOv8	AP_{50}	AP	ROpti	Trajanje obrade po slici [ms]
YOLOv8n	0,868	0,549	0,71	**4,6**
YOLOv8s	0,903	0,606	0,76	8,0
YOLOv8m	0,906	0,621	0,77	17,3
YOLOv8l	0,908	0,608	0,78	34,4
YOLOv8x	**0,913**	**0,638**	**0,79**	46,5

U Tablica 2.4 vidimo da je najbolje rezultate postigao YOLOv8x s AP_{50} 91.3% i AP 68.8%, što ga čini najprikladnijim za naknadnu analizu materijala snimljenih tijekom leta bespilotne letjelice jer je točnost u tom slučaju najvažnija. Model YOLOv8n pokazuje značajno najbržu detekciju, svega 4,6 ms po slici, uz postizanje mAP_{50} samo 4,5% niže od najboljih rezultata. Slično tome, model YOLOv8s ostvaruje drugo najbolje vrijeme zaključivanja s gotovo identičnim performansama mAP_{50} kao YOLOv8x. Ovo ga čini posebno prikladnim za primjenu tijekom akcija traženja i spašavanja, gdje, uz preciznost detekcije, brzina zaključivanja u stvarnom vremenu ima ključnu ulogu, a računalni resursi mogu biti ograničeni na manje moćne računalne sustave.

2.4 Geolokacija

U ovom dijelu doktorskog rada prikazani su rezultati dobiveni istraživanjem geolokacijskih metoda poput, pojednostavljenog elipsoidnog modela Zemlje, algoritma koji koristi DEM (engl. *Digital Elevation Model*) i algoritma mjerenja presjeka.

U operacijama traganja i spašavanja, bespilotne letjelice omogućuju pretraživanje terena na dva načina: u stvarnom vremenu (engl. *online*) za vrijeme leta i naknadno na snimljenom materijalu (engl. *offline*). Prilikom online

pretraživanja, udaljeni pilot bespilotne letjelice pregledava teren pomoću ugrađene kamere istovremeno s upravljanjem letjelice. Slike se prikazuju na zaslonu upravljača ili na većem zaslonu ako je dostupan udaljenom pilotu na licu mjesta (često u vozilu). Usporedno, snimke se pohranjuju na memorijsku karticu na letjelici u višoj kvaliteti koje je moguće analizirati naknadno nakon leta. Ako tražena osoba nije pronađena tijekom online pretraživanja, provodi se dodatan pregled snimljenog materijala, poznato kao metoda offline pretraživanja. U tom slučaju ako se na snimkama detektira tražena osoba, kako bi zemaljski tim pristupio traženoj osobi, potrebno je znati njen geografski položaj.

Ulazni podaci za predloženu metodu offline geolokacije osobe na slici snimljenoj tijekom leta bespilotne letjelice su metapodaci koji su pohranjeni uz svaku snimljenu sliku i visina tla na mjestu polijetanja letjelice.

Od mnogih metapodataka zabilježenih tijekom leta bespilotne letjelice za lokalizaciju detektirane osobe, koristili smo njihov podskup koji se sastoji od podataka vezanih uz putanju bespilotne letjelice, identifikaciju slike i parametre kamere u trenutku snimanja fotografije. Upotrijebljeni podaci i specifičan primjer vrijednosti prikazani su u Tablica 2.5.

Tablica 2.5 Metapodaci korišteni za geolokaciju i njihove mjerne jedinice

Varijabla	**Opis**	**Primjer**	**Mjerna jedinica**
Time	Vremenska oznaka točke u letu	2022-09-26 19:35:42	
File_Name	Naziv snimljene slike	DJI_0265.JPG	
Img_Width	Širina snimljene slike	5472	broj piksela
Img_Height	Visina snimljene slike	3648	broj piksela
FOV	Dijagonalno vidno polje kamere	84	stupnjeva

Relative_Altitude	Visina letjelice u letu u odnosu na točku polijetanja	30.1	metar
Gimbal_Pitch_Degree	Nagib kamere	-45.8	stupnjeva
Gimbal_Yaw_Degree	Horizontalni kut kamere	15	stupnjeva
Gimbal_Roll_Degree	Rotacija kamere	0	stupnjeva
GPS_N	Geografska širina	45.5107911388	stupnjeva
GPS_E	Geografska dužina	16.7602712222	stupnjeva

Kako bismo pojednostavili problem detekcije/praćenja osobe na slikama, skup podataka u analiziranom slučaju oblikovan je tako da u slikama koje predstavljaju objekt za detekciju postoji samo jedna osoba.

Tablica 2.6 Izračun koordinata osobe koja stoji na poznatoj lokaciji.

Skup podataka	**Broj snimaka**	**Pojednostavljen elepsoidni model Zemlje**			**DEM**			**Algoritam mjerenja presjeka**		
		MeanError	MaxError	MinError	MeanError	MaxError	MinError	MeanError	MaxError	MinError
PhantomLP1	10	8,96	10,54	7,87				13,45	14,38	12,71
PhantomLP2	10	8,70	11,6	6,212				8,44	8,83	7,59
PhantomVP1	4	18,37	29,26	8,412	10,94	15,83	5,630	4,79	5,45	4,00
PhantomVP2	7	50,49	73,03	14,43	23,60	34,68	7,327	10,53	11,14	10,35
PhantomVP3	9	51,31	98,20	22,82	29,91	66,89	14,76	12,39	14,47	9,73

Za ocjenu metoda lokalizacije i predviđanja kretanja osoba provedeno je nekoliko eksperimenata u realnim uvjetima koji uključuju ove odnose kretanja osobe i bespilotne letjelice:

1. osoba i bespilotna letjelica miruju
2. osoba miruje, a bespilotna letjelica se giba
3. osoba se giba dok bespilotna letjelica miruje/lebdi
4. osoba i bespilotna letjelica se gibaju

Cilj eksperimenta u kojemu osoba i bespilotna letjelica miruju je provjeriti točnost metode tj. koliko je odstupanje od stvarne prostorne koordinate osobe. Eksperiment je izveden na dvije lokacije, terenu bez nagiba (livada) i s nagibom (vinograd). Na livadi su snimljena tri seta dok u vinogradu dva, u svakom setu se nalazi po 10 snimaka. U slučaju snimaka s letjelicom Phantom 4 Advance [28] FOV kamere je 84° dok je rezolucija slike 5472 x 3648 px, letjelica je lebdjela na visini od 30 m, Letjelica Mavic 2 Enterprise Advanced [29] letjela je također na visini od 30 m iznad točke uzlijetanja, rezolucija slika snimljenih ovom letjelicom je 8000 x 6000 dok je FOV kamere 84°. Iz dobivenih rezultata prikazanih u [19] vidljivo je da u slučaju snimaka na livadi srednja pogreška mjerenja iznosi 2,2 m zašto smatramo da je dovoljno precizan rezultat kako bi mogli locirati osobu na terenu. U slučaju vinograda pogreška iznosi 17 m dok se korištenjem DEM algoritma ta pogreška smanjuje na 8 m.

Za eksperiment u kojemu osoba miruje a bespilotna letjelica se giba snimanje je također napravljeno na dvije lokacije. Na livadi su snimljena dva seta dok su u vinogradu snimljena tri. U setu primijenjen je realni scenarij u slučaju potrage za nestalom osobom ovakvim tipom letjelice, što znači da je letjelica letjela iznad terena snimajući slike u vremenskom razmaku od dvije sekunde. I u ovom eksperimentu kao i u prethodnom na livadi dobivamo preciznije rezultate u odnosu na vinogradi kao i korištenjem DEM algoritma, za set Phantom VP1 poboljšanje iznosi 40%, Phantom VP 2 53% i Phantom VP3 46% u odnosu na rezultate kada ne nije koristio DEM.

Slučaj u kojemu se osoba giba dok bespilotna letjelica lebdi na mjestu, snimljen je u dva seta na livadi i dva seta u vinogradu. Ovakva metoda traganja tipična je za velike letjelice gdje se letjelica nalaze na visinama između 100 i 300 m te prostor pregledavaju pomicanjem kamere. U ovom eksperimentu s letjelicom Mavic2 EA dobivena je srednja pogreška od 6,82 m za sve snimljene setove.

U četvrtom eksperimentu giba se i osoba i bespilotna letjelica. Tri su seta snimljena na livadi a dva u vinogradu. U ovom slučaju osoba koja nosi Garmin GPSMAP 78 (kod snimanja s Phantom 4 Advanced) ili GPSMAP 65s ručni

navigacijski uređaj s podrškom za više frekvencijske sustave / višestruki GNSS (kod snimanja s Mavic 2 Enterprise Advanced) koji snima poziciju na kojoj se nalazi osoba svake sekunde i te podatke sprema u .gpx datoteku. Ova datoteka služi nam za određivanje točnosti našeg izračuna. Na isti način praćena je pozicija gibanja osobe u slučaju gibanja osobe dok letjelica miruje. Ovaj realan scenarij, pogotovo u početnoj fazi potrage. Srednja pogreška kreće se od 2,29 m do 13,95 m, što je dobar rezultat ako uzmemo u obzir da terenske ekipe za statistički krug u kojemu je vjerojatnost 75% da se osoba nalazi trebaju pretražiti 19.625.000 m^2.

Peti eksperiment bavi se brzinom gibanja osobe, tj. izračunom iznosa brzine i smjera kretanja osobe što nam služi kao podloga za određivanje novog potražnog područja.

Tablica 2.6 prikazuje rezultate (rada [19] i RAD 6.) procjene udaljenosti između izračunate GPS lokacije osobe pomoću navedena tri algoritma i točne GPS lokacije na kojoj se osoba nalazila. Algoritmi su testirani na pet različitih vlastitih skupova podataka, od kojih su PhantomLP1 i PhantomLP2 snimljena na livadi (ravan teren), dok su tri (PhantomVP1-PhantomVP3) snimljena u vinogradu (nagnuti teren). U skupovima podataka snimljenim na livadi nije primijećeno veće odstupanje između algoritama (npr. razlika srednje pogreške od 4,5 m za skup PhantomLP1), međutim, na terenima s različitim nagibima, algoritam mjerenja presjeka pokazuje značajno bolje rezultate od drugih algoritama. Najprecizniji rezultat ostvaren je u prvom setu snimljenom u vinogradu (PhantomVP1), s prosječnom pogreškom od 4,8 metara. U slučaju modela Zemljinog elipsoida i modela DEM, točnost je provjerena za svaku sliku u skupu podataka.

Na temelju provedenih istraživanja preporuka je da se u akcijama traganja i spašavanja tijekom offline pregleda snimljenog materijala, u slučaju pozitivne detekcije tražene osobe koristi DEM model, ukoliko je osoba detektirana na samo jednoj snimci ili je iz snimaka vidljivo da se osoba giba. To je zato što DEM model pokazuje veću preciznost na terenima s nagibom u usporedbi s pojednostavljenim elipsoidnim modelom. Kod detekcije nepokretne osobe,

detektirane na više slika, predlaže se korištenje algoritma mjerenja presjeka, čime se postižu najbolji rezultati.

Kada se osoba detektira na dvije ili više slika [19], iz dobivenih lokacija moguće je odrediti brzinu osobe ako se osoba giba. Ova informacija sužava područje potrage i skraćuje vrijeme potrebno zemaljskim timovima da pronađu osobu. Uz iznos brzine određuje se i smjer gibanja kao azimut između početnog i konačnog položaja na kojemu je osoba detektirana. Iz navedenog pomaka osobe kreira se novo potražno područje u koje se šalju zemaljski timovi za traganje.

Tablica 2.7 Detektirane brzine kretanja osobe [19].

Skup podataka	**Broj snimaka**	**Detektirana brzina (m/s)**	**Brzina prema Garmin GPSMAP 78/ Garmin GPSMAP 65s (m/s)**
Phantom LM 1 DEM	13	1,172	1,085
Phantom LM 2 DEM	20	1,639	0,799
Phantom LM 3 DEM	10	1,199	1,257
Mavic LM 1 DEM	10	1,088	1,344
Mavic LM 2 DEM	5	1,172	1,259
Mavic LM 3 DEM	7	1,617	1,284
Mavic LM 4 DEM	3	1,141	1,166
Phantom VM 1 DEM	6	2,571	1,096
Phantom VM 2 DEM	5	1,053	1,192
Mavic VM 1 DEM	5	2,062	1,029
Mavic VM 2 DEM	6	0,421	0,854
Mavic VM 3 DEM	5	0,563	0,371
Mavic VM 4 DEM	12	1,386	1,250

Tablica 2.7 prikazuje usporedbu detektirane i referentne brzine kretanja detektirane osobe na slici. Detektirana brzina predstavlja srednju brzinu određenu kao kvocijent prijeđenog puta (udaljenost između dvije detektirane GPS koordinate) i vremena (vrijeme proteklo između nastanka slika). Garmin brzina određena je korištenjem točaka izmjerenih pomoću ručnog GPS uređaja (u slučaju Phantom skupa podataka to je GPSMAP 78 dok je za Mavic2 EA skup

korišten GPSMAP 65). Broj snimaka u skupu podataka predstavlja broj slika na kojima je osoba detektirana.

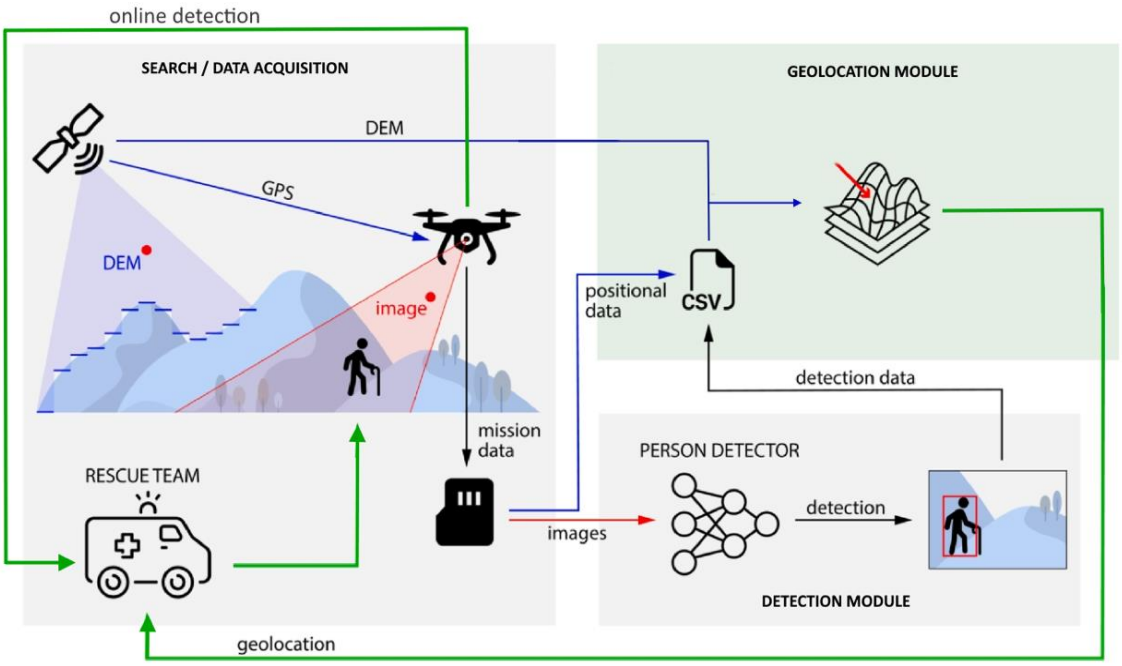
Paralelno s ovim istraživanjima, u suradnji s članovima tima Laboratorija za računalni vid, Fakulteta informatike i digitalnih tehnologija Sveučilišta u Rijeci, rađena su istraživanja određivanja geolokacije pomoću raycast metode koja daje rezultate velike preciznosti [30][31]. Nakon niza eksperimenata na terenima različitih konfiguracija i složenosti, korištenjem prilagođenog 3D generatora terena i raycast metodu, zajedno s detektorom osoba temeljenim na dubokoj neuronskoj mreži obučenoj na našem prilagođenom skupu podataka, definirali smo metodu za geolokaciju detektiranih osoba. Naša metoda prevladava probleme s kojima su se suočavale prethodne metode i postiže visoku pouzdanost, čak i uz samo 4 uzastopne detekcije. Također, kratko vrijeme obrade omogućuje učinkovitu analizu podataka snimljenih tijekom leta bespilotne letjelice, dokazavši da se predložena metoda može uspješno koristiti u stvarnim SAR misijama.

2.5 Prototip sustava za detekciju i geolokalizaciju

Sustav za detekciju i geolokalizaciju osoba u akcijama traganja i spašavanja primjenom bespilotnih letjelica zamišljen je kao sustav koji ima integriran YOLOv8m detektor objekata podešen za detekciju osoba snimljenih bespilotnom letjelicom u scenama traganja i spašavanja i algoritam za geolociranje na snimkama iz zraka. Sustav pomaže pilotu bespilotnog sustava da pronađe nestalu osobu na način da označi detektirane osobe na ekranu tijekom leta ili na snimkama tijekom naknadne pretrage na snimljenome materijalu. Jedan od ciljeva je da ovakav sustav bude jednostavan za korištenje, posebno u otežanim uvjetima izvan urbanog područja.

Na Slika 2.10 vidimo prikaz sustava za traganje i spašavanje. Sustav je podijeljen u tri modula: modul u kojemu se vrši potraga i sakupljanje terenskih podataka, modul za detekciju i modu za geolokalizaciju. U fazi online pretrage terena , ukoliko udaljeni pilot bespilotne letjelice uoči nestalu osobu, šalje spasilačkom

timu lokaciju koju očita na upravljaču letjelice te tim pristupa osobi i završava dio potrage. Paralelno uz pretragu terena prikupljaju se podaci (slike i metapodaci). Sateliti povremeno snimaju digitalne karte nadmorske visine (DEM) i opskrbljuju bespilotnu letjelicu GPS podacima tijekom leta. Metapodaci uključuju položaj i orijentaciju bespilotne letjelice i podatke kamere. Slike i metapodaci snimaju se na SD memorijsku karticu i postaju dostupni za offline obradu nakon povratka bespilotne letjelice u bazu.

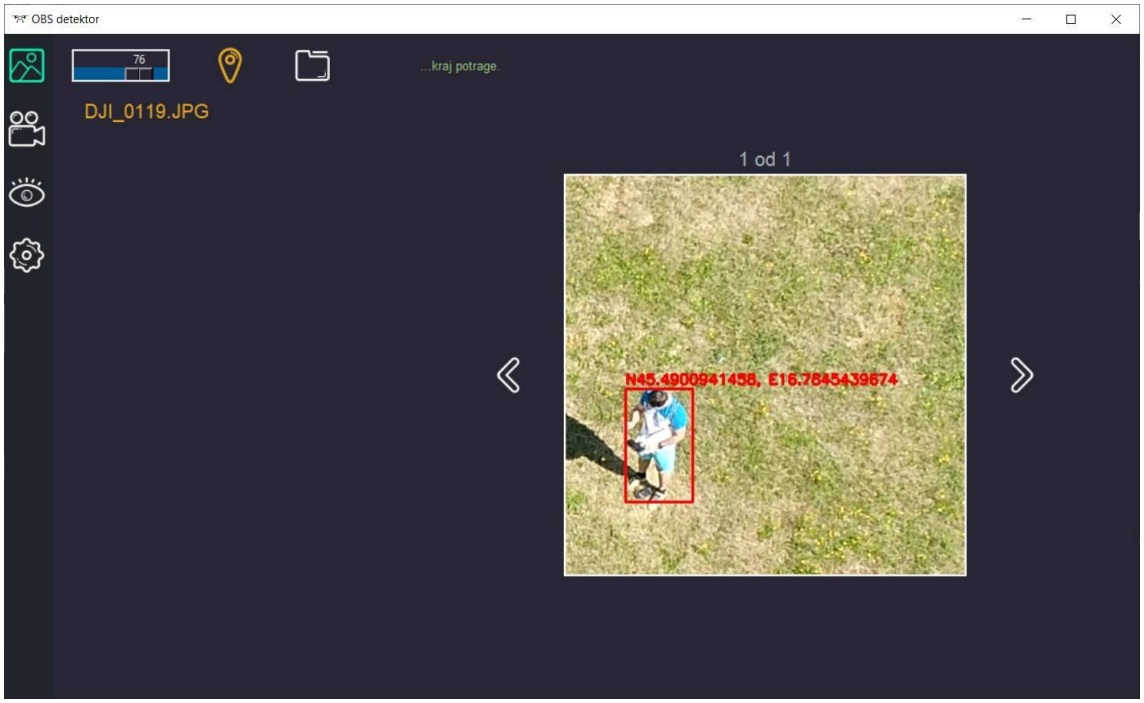


Slika 2.10 Prikaz sustava za traganje i spašavanje

U fazi offline obrade podataka za analizu slika snimljenih tijekom leta bespilotne letjelice, koristi se modul za automatsku detekciju osoba. Taj modul koristi prethodno obučen i fino podešen model Yolov8x duboke konvolucijske neuronske mreže za automatsku detekciju osoba, posebice ozlijeđenih i nestalih osoba. U geolokacijskom modulu, dobiveni podaci detekcije kombiniraju se s podacima o položaju bespilotne letjelice u jednoj CSV datoteci i DEM podacima terena na kojemu je vršena potraga. Korištenjem tih podataka i ranije predloženim algoritmima određuje se položaj detektirane osobe. Podaci o

geolokaciji prosljeđuju se spasilačkom timu koji pristupa osobi na zemlji i uspješno završava SAR misiju.

Da bismo demonstrirali i testirali primjenu predloženih metoda, razvijen je prototip desktop aplikacije (Slika 2.11) namijenjen za upotrebu u operacijama traganja i spašavanja. Tijekom terenskog rada, upravljač bespilotne letjelice povezan je putem HDMI kabela (ili bežično putem WiFi-ja) s računalom na kojem se izvršava detektor, koji obavlja detekciju na slikama dobivenim iz videa tijekom leta (Slika 2.12). Drugi dio aplikacije namijenjen je offline detekciji i detekciji na snimljenom videu ili na snimkama napravljenim tijekom leta na terenu.



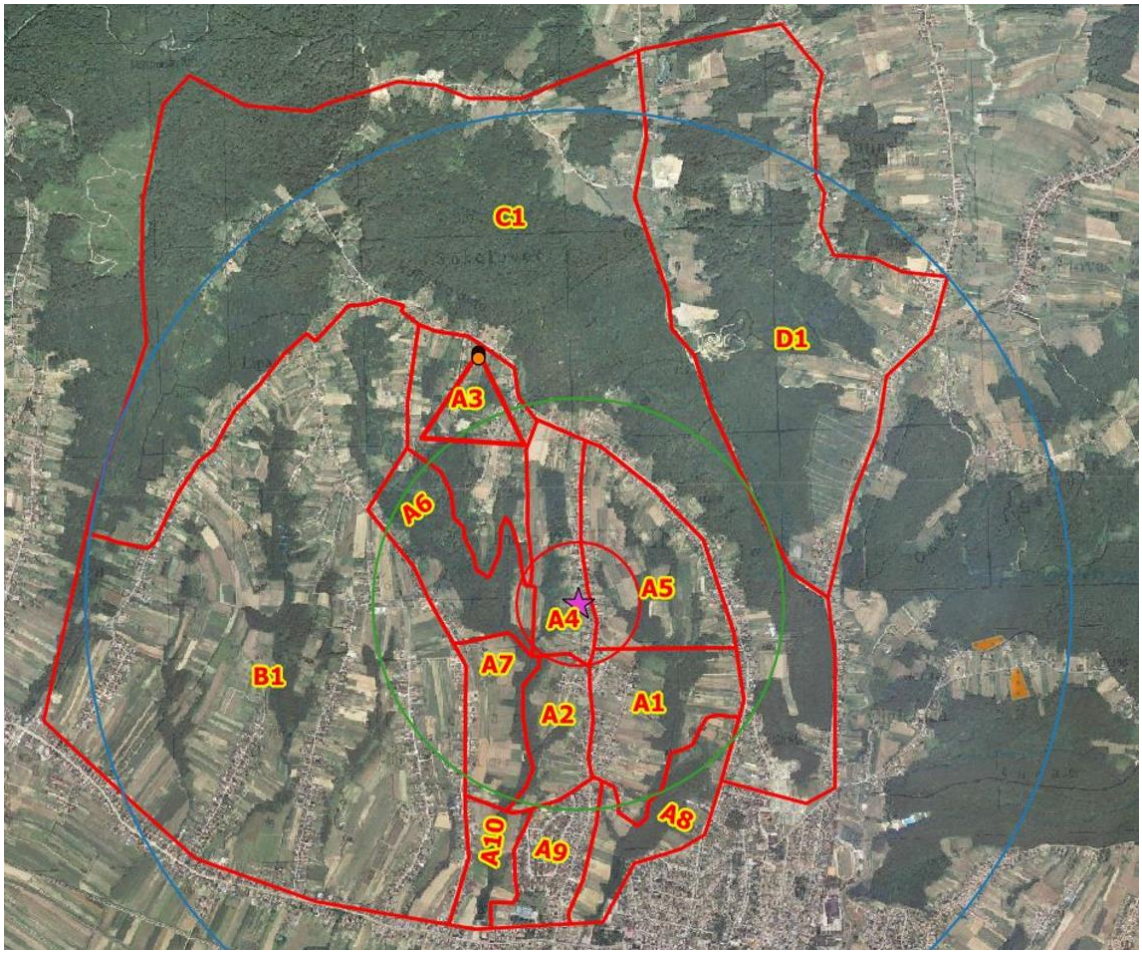
Slika 2.11 Korisničko sučelje aplikacije "OBS detektor". Aplikacija prikazuje detektiranu osobu zajedno s GPS koordinatama lokacije osobe.


a)

b)

Slika 2.12 Komponente sustava spremnog za pretragu terena. a) Kontroler bespilotne letjelice Phantom 4 Advanced je povezan s računalom putem HDMI kabela, na kojem je prikazana slika s ekrana kontrolera. b) Kontroler bespilotne letjelice Mavic 2 Enterprise Advanced povezan je s računalom bežičnim putem.

Na Slika 2.13 prikazan je primjer početne faze potrage za nestalom dementnom osobom na području Moslavine. Krugovi na slici označavaju statistička područja dosadašnjih pronalaska osoba istog tipa, poput demencije, djeteta, planinara, gljivara i sl. Prvi krug, prema statističkim podacima, predstavlja područje s 25% vjerojatnosti pronalaska nestale osobe (što iznosi 300 m), zeleni krug s 50% vjerojatnosti (radijus 1000 m), dok je plavi krug s 75% vjerojatnosti za pronalazak nestale osobe (radijus 2400 m). Crvenom linijom označena je zona subjektivne pretrage, koju će zemaljski timovi pretraživati, a zatim je podijeljena na zone označene slovima A, B, C i D. Zona A dodatno je podijeljena na segmente A1 - A10, gdje se šalju timovi za potragu. Zvjezdica na Slika 2.13 označava IPP (engl. *Initial Planning Point*), koji je obično točka posljednjeg viđenja ili zadnja poznata lokacija nestale osobe. U slučaju pronalaska nestale osobe u offline načinu traganja, njena lokacija na karti se izračunava (žuta točka na slici 16 unutar segmenta A3) te na temelju izračunate brzine, proteklog vremena od nastanka slike i azimuta kretanja osobe, kreira se novi segment pretrage (crveni trokut na Slika 2.13)



Slika 2.13 GIS u akcijama traganja i spašavanja [19].

3 ZAKLJUČAK

Ovaj doktorski rad istražuje mogućnosti, uspješnost i pouzdanost metoda umjetne inteligencije, konkretno dubokih konvolucijskih neuronskih mreža u detekciji osoba na snimkama iz zraka napravljenim pomoću bespilotnih letjelica na neurbanim područjima. U tu svrhu odabran je YOLOv4 model koji je postigao odlične rezultate prepoznavanja osoba na RGB snimkama iz zraka. Glavni eksperiment, a to je detekcija osoba u ne tipičnim pozama osoba, proveden je na prilagođenom vlastitom skupu podataka SARD snimljenom na području Moslavačke gore. Uz navedeni skup podataka kreiran je i skup podataka koji je služio za procijene algoritama za određivanje geolokacije prepoznate osobe.

Yolov4 duboka konvolucijska neuronska mreža trenirana na MS COCO skupu podataka korištena je kao osnovna mreža koja je dodatno trenirana na SARD skupu podataka i sa proširenim Corr skupom podataka. Iako je osnovni model na skupu SARD imao prosječnu preciznost od 23% poslužio je kao dobra polazna točka za treniranje novog modela koji je dodatno treniran na snimkama bespilotne letjelice.

Model YOLOv4(SARD) postigao je preciznost od 59,7% što je 37% bolji rezultat od osnovnog modela tj. ovaj model točno je prepoznao je 2512 osoba od mogućih 2611 u skupu uz samo 88 netočnih prepoznavanja. Kako bi se provjerio rad modela u otežanim uvjetima model je testiran na Corr skupu podataka. Corr skup podataka sadrži slike koje dodatno simuliraju različite vremenske uvjete koji se mogu dogoditi u stvarnim situacijama traganja i spašavanja kao što su magla, snijeg i led. Također, zamućene slike su uključene u Corr set koje simuliraju zamućenje nastalo zbog kretanja kamere tijekom snimanja iz zraka. Korištenjem SARD i Corr skupova podataka postignuti su vrlo dobri rezultati za sve kategorije vremenskih uvjeta.

Rezultati istraživanja transfera znanja pokazuju da su najuspješniji rezultati dobiveni korištenjem metode mrežnog transfera znanja, koja prenosi značajke naučene na velikim skupovima podataka, te metode transfera znanja zasnovane na instancama, gdje je model treniran na slikama domene koje odgovaraju slikama na kojima će se model testirati. Što znači da skup podataka kojim se

obučava konvolucijska neuronska mreža treba odgovarati području u kojemu će se vršiti potraga tj. učinkovitije je imati dva odvojena modela (primjerice, jedan za kontinentalnu Hrvatsku i drugi za obalni dio) umjesto jednog koji bi pokrivao čitavo područje Republike Hrvatske.

Posljednja verzija YOLOv8 modela pokazala je iznimne performanse unatoč tome što je model trenirana na dijelu SARD skupa podataka. Verzija modela YOLOv8x postigla je srednje preciznosti od 63,8%, odnosno 91,3% za AP_{50} što ju čini najprikladniju za naknadnu analizu snimaka. Model YOLOv8s ostvaruje slične performanse srednje preciznosti od 60,6% i 90,3% AP_{50} ali značajno kraće vrijeme zaključivanja od 8 ms za jednu sliku (YOLOv8x treba 46,5 ms) što ga čini posebno prikladnim za primjenu tijekom akcija traženja i spašavanja.

Ispitivanjem preciznosti geolokalizacije detektiranih osoba na zračnim snimkama provedeno je korištenje dva modela bespilotnih letjelica. Rezultati su pokazali da se primjenom algoritma presjeka postiže točnost unutar radijusa od 5 metara na terenima s nagibom, što se smatra dovoljno preciznim za lokalizaciju osoba od strane ljudskih timova na terenu.

Osim spomenutih doprinosa, eksperimenti iz ovog rada pokazali su da kombinacija bespilotnih letjelica i duboke neuronskih mreža otvara širok spektar novih mogućnosti implementacije u različitim područjima.

3.1 Znanstveni doprinos

Doktorski rad temeljen je na predviđenim znanstvenim doprinosima, koji su prethodno navedeni u odjeljku 1.3. U ovom kontekstu, želimo ih ponovno istaknuti, pružajući dodatna pojašnjenja za svaki pojedinačni doprinos uz dokaze o ostvarenim rezultatima.

Sukladno navedenom, očekivani znanstveni doprinosi su bili:

- ***izrada baze slika i snimaka bespilotnom letjelicom nestalih/ozlijeđenih osoba na neurbanom području pripremljene za obučavanje nadziranog modela strojnog učenja.***

U radovima RAD 3 i RAD 4 predstavljena je baza slika snimaka bespilotnom letjelicom (SARD) s pripadajućim proširenjem u smislu Corr skupa podataka, te je ista korištena u eksperimentalnom dijelu istraživanja u okviru radova objavljenih tijekom ovog doktorskog studija. Kreirana baza javno je objavljena i koristi se u znanstvenoj zajednici za definiranje novih modela za detekciju osoba i druga povezana istraživanja.

- ***model sustava za detekciju osoba na snimkama snimljenih bespilotnom letjelicom u akcijama traganja i spašavanja***

  Model za detekciju osoba na snimkama snimljenim bespilotnom letjelicom predstavljen je u RAD 4 i RAD 6, dok je u RAD 5 predstavljeno kako postojeći model tehnikama transfera znanja dodatno učiniti robusnijim i preciznijim u smislu detekcije osoba na snimkama snimljenim bespilotnom letjelicom.

- ***metoda za procjenu udaljenosti detektirane osobe od položaja bespilotne letjelice***

  U radu [19] opisana je metoda za geolokalizaciju detektirane osobe, što se pokazalo korisnijim timovima na terenu u usporedbi s udaljenošću detektirane osobe od letjelice. Svi algoritmi korišteni za određivanje geolokacije koriste neku od metoda za određivanje udaljenosti detektirane osobe od položaja letjelice. Uz poznatu GPS lokaciju letjelice i njenu orijentaciju, izračunava se azimut detektirane osobe koji, uz udaljenost, omogućuje dobivanje GPS koordinate osobe.

- ***prototip sustava za detekciju osoba u akcijama traganja i spašavanja bespilotnim letjelicama***

  Prototip sustava predstavljen je u poglavlju 2.5 te je testiran u realnim operacijama gorske službe spašavanja. Sustav poput ovog, uz detekciju i geolokalizaciju može pomoći u planiranju akcije traganja i spašavanja npr.

kroz segmentaciju potražnog područja na područja koja su pokrivena šumom, livadama, vodom i sl. Također ovakvi sustavi mogu na temelju detekcije osobe u pokretu predlagati novo potražno područje što može dovesti u bržeg pronalaska nestale osobe. Ove primjene su također testirane na terenu u realnim okolnostima.

3.2 Buduća istraživanja

Znanstveni doprinosi, proizašli iz ovog doktorskog rada kao početne točke, usmjereni su na eksperimentiranje i istraživanje mogućnosti primjene detektora objekata temeljenih na dubokim konvolucijskim neuronskim mrežama u akcijama traganja i spašavanja za detekciju nestale osobe. U daljnjem radu, fokus će biti na obučavanju modela dubokog učenja za detekciju osoba na RGB i termalnim snimkama. S obzirom na rastuću dostupnost bespilotnih letjelica s termo vizijskim kamerama, ovakav pristup pretrazi mogao bi dodatno ubrzati pronalazak, posebice u dijelu godine kada je vegetacija u mirovanju.

U daljnjem istraživanju namjeravamo temeljito ispitati robusnost modela na različite vremenske uvjete, posebice noćno snimanje. Također, planiramo provesti eksperimente s više skupova podataka kako bismo poboljšali robusnost i sposobnost generalizacije našeg modela. Dodatno, planiramo istražiti mogućnosti stalnog obučavanja modela kako bismo model prilagodili novim situacijama.

U cilju nastavka istraživanja koja su provedena u ovom doktorskom radu prijavljena su dva projekta, od kojeg je jedan dobio financiranje, a za drugi čekamo rezultate evaluacije:

- NPOO projekt transfera tehnologije pod nazivom: ***„SAR-DAG: Sustav nadzora za pomoć u akcijama traganja i spašavanja temeljen na umjetnoj inteligenciji“*** čiji je cilj razvoj sustava SAR-DAG za detekciju i geolokaciju osoba u neurbanim područjima pomoću modela računalnog vida i bespilotne letjelice koji će, u realnom vremenu ili naknadno offline, automatski detektirati unesrećene

osobe kao i njihovu preciznu lokaciju i/ili smjer te brzinu kretanja. Čekamo rezultate evaluacije.

- UNIRI projekt, „SAR-DAG: ***Automatska detekcija i geolokacija osoba snimljenih dronom u akcijama traganja i spašavanja***“, uniri-iskusni-drustv-23-278, kojem je cilj poboljšati rezultate detekcije i geolokacije osoba snimljenih dronom u različitim realnim uvjetima.

4 SAŽETAK RADOVA

4.1 RAD 1. Detekcija igračaka vojnika snimljena iz ptičje perspektive pomoću konvolucijskih neuronskih mreža / Detection of toy soldiers taken from a bird's perspective using convolutional neural networks

Ovaj rad opisuje upotrebu dva različita pristupa dubokom učenju za detekciju objekata kako bi se prepoznao vojnik igračka. Koristimo snimke igračaka vojnika u različitim pozama pod različitim scenarijima kako bismo simulirali izgled osoba na snimci snimljenoj bespilotnom letjelicom. Snimke iz ptičje perspektive danas se masovno koriste u potrazi za nestalim osobama u neurbanim područjima, graničnoj kontroli, kontroli kretanja životinja i slično. Usporedili smo single-shot multi-box detektor (SSD) s MobileNet ili Inception V2 kao okosnicom, SSDLite s MobileNet i Faster R-CNN u kombinaciji s Inception V2 i ResNet50. Rezultati pokazuju da Faster R-CNN uspješnije detektira male objekte kao što su vojnici od SSD-a, a vrijeme treninga Faster R-CNN-a mnogo je kraće od SSD-a.

Dostupno na: https://link.springer.com/chapter/10.1007/978-3-030-33110-8_2

Sambolek, S., and Ivašić-Kos, M., 2019. Detection of toy soldiers taken from a bird's perspective using convolutional neural networks. In ICT Innovations 2019. Big Data Processing and Mining: 11th International Conference, ICT Innovations 2019, Ohrid, North Macedonia, October 17–19, 2019, Proceedings 11 (pp. 13-26). Springer International Publishing.

4.2 RAD 2. Detekcija objekata na slikama s bespilotnih letjelica: kratki pregled napretka / Detecting objects in drone imagery: a brief overview of recent progress

Detekcija objekata na snimkama bespilotnih letjelica (dronova) zahtjevan je zadatak i nedovoljno istražen problem koji u posljednje vrijeme dobiva sve više pažnje u istraživačkoj zajednici. Kod snimanja bespilotnom letjelicom ne mijenjaju se samo vrijeme i svjetlosni uvjeti, već se mijenjaju i visina i kut snimanja jer položaj kamere nije fiksan tijekom snimanja. Rad ima za cilj opisati mogućnosti korištenja bespilotnih letjelica u operacijama traganja i spašavanja te dati cjelovit pregled područja vezanog za detekciju osoba na snimkama bespilotnih letjelica. Rad uključuje opis javno dostupnih skupova podataka i usporedbu najsuvremenijih modela detekcije osoba na snimkama bespilotnih letjelica te završava prijedlogom budućih istraživanja.

Dostupno na: https://ieeexplore.ieee.org/abstract/document/9245321

Sambolek, S., and Ivašić-Kos, M., 2020. Detecting objects in drone imagery: a brief overview of recent progress. In 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO) (pp. 1052-1057). IEEE.

4.3 RAD 3. Detekcija osoba na slikama s bespilotne letjelice / Person Detection in Drone Imagery

Korištenje bespilotnih letjelica u operacijama traganja i spašavanja postalo je standard gotovo svugdje u svijetu. Poseban izazov u tim operacijama je automatska detekcija osoba u različitim terenima, situacijama, položajima tijela, vremenskim uvjetima te s različitih visina snimanja tijekom leta bespilotne letjelice. Ovaj rad istražuje točnost detekcije ljudi na slikama snimljenim bespilotnom letjelicom na postojećim skupovima podataka VisDrone, Okutama-Action te na prilagođenom skupu podataka SARD, izrađenom kako bi simulirao scene traganja i spašavanja. Kao detektor korišten je Faster R-CNN s FPN kao osnovom, prethodno obučen na COCO skupu podataka. Detektor osoba dodatno je treniran na SARD skupu podataka koji sadrži 1,981 slika i na podskupu VisDrone skupa podataka. Nakon transfera znanja postignuto je značajno poboljšanje rezultata detekcije osoba na slikama snimljenim bespilotnom letjelicom, posebno u pogledu srednje preciznosti i odziva.

Dostupno na: https://ieeexplore.ieee.org/abstract/document/9243737

Sambolek, S., and Ivasic-Kos, M., 2020. Person detection in drone imagery. In 2020 5th International Conference on Smart and Sustainable Technologies (SpliTech) (pp. 1-6). IEEE.

4.4 RAD 4. Automatska detekcija osoba u operacijama traganja i spašavanja pomoću dubokih konvolucijskih neuronskih mreža / Automatic Person Detection in Search and Rescue Operations Using Deep CNN Detectors

Zbog sve većeg broja ljudi koji se bave raznim adrenalinskim aktivnostima ili avanturističkim turizmom i borave u planinama i drugim teško pristupačnim mjestima, raste potreba organizacije operacija traganja i spašavanja (SAR) radi pružanja pomoći i zdravstvene skrbi ozlijeđenima. Cilj SAR operacije je pretražiti najveće područje teritorija u najkraćem mogućem vremenu i pronaći izgubljenu ili ozlijeđenu osobu. Danas su dronovi (bespilotne letjelice ili UAV) sve više uključeni u operacije traženja, budući da mogu brzo snimiti veliko, kontrolirano područje. Međutim, detaljna analiza velike količine snimljenog materijala ostaje problem. Čak i za stručnjaka nije lako pronaći tražene osobe koje su relativno male u odnosu na područje gdje se nalaze, često skrivene vegetacijom ili stopljene s tlom te u neobičnim pozama zbog padova, ozljeda ili iscrpljenosti. Stoga je automatska detekcija osoba i objekata na slikama i videozapisima snimljenim bespilotnim letjelicama u ovim operacijama vrlo značajna. U ovom radu istraživana je pouzdanost postojećih detektora najnovije generacije poput Faster R-CNN, YOLOv4, RetinaNet i Cascade R-CNN na skupu podadataka VisDrone i prilagođenom skupu podataka SARD, izrađenom kako bi simulirao scenarije spašavanja. Nakon treniranja modela na odabranim skupovima podataka, rezultati detekcije uspoređeni su. Zbog visoke brzine, preciznosti i malog broja lažnih detekcija, detektor YOLOv4 odabran je za daljnje ispitivanje. Analizirani su rezultati modela YOLOv4 u vezi s različitim veličinama mreže, različitim točnostima detekcije te postavkama transfera znanja. Također je ispitana robusnost modela na vremenske uvjete i zamućenje uzrokovano pokretom. Rad predlaže model koji se može koristiti u SAR operacijama zbog izvrsnih rezultata u detekciji ljudi u scenarijima traganja i spašavanja.

Dostupno na: https://ieeexplore.ieee.org/abstract/document/9369386

Sambolek, S., and Ivasic-Kos, M., 2021. Automatic person detection in search and rescue operations using deep CNN detectors. IEEE Access, 9, 37905-37922.

4.5 RAD 5. Metode transfera znanja za treniranje detektora osoba na slikama s bespilotnih letjelica / Transfer Learning Methods for Training Person Detector in Drone Imagery

Duboke neuronske mreže ostvaruju izvrsne rezultate na različitim zadacima računalnog vida, ali obučavanje tih modela zahtijeva velike količine označenih slika koje često nisu dostupne. Kao alternativno rješenje za postizanje boljih rezultata i veće sposobnosti generalizacije modela, a bez potrebe za velikim brojem podataka, koristi se pristup transfera znanja, odnosno prilagodbe prethodno naučenih modela zadatku koji je pred nama.

Cilj ovog rada je poboljšati rezultate detekcije ljudi u scenama traganja i spašavanja primjenom detektora YOLOv4. Budući da je izvorni skup podataka SARD za treniranje detektora ljudi u scenama traganja i spašavanja ograničen, razmatraju se različiti pristupi transfera znanja. Dodatno, koristi se skup podataka VisDrone koji sadrži slike s bespilotnih letjelica u urbanim područjima kako bi se povećao skup podataka za treniranje i time poboljšali rezultati detekcije osoba.

Dostupno na: https://link.springer.com/chapter/10.1007/978-3-030-82196-8_51

Sambolek, S., and Ivašić-Kos, M., 2022. Transfer learning methods for training person detector in drone imagery. In Intelligent Systems and Applications: Proceedings of the 2021 Intelligent Systems Conference (IntelliSys) Volume 2 (pp. 688-701). Springer International Publishing.

4.6 RAD 6. Detekcija osoba i procjena geolokacije na zračnim slikama s bespilotnih letjelica: Eksperimentalni pristup / Person Detection and Geolocation Estimation in UAV Aerial Images: An Experimental Approach

Upotreba bespilotnih letjelica u operacijama traganja i spašavanja postala je neophodna za pomoć u pronalasku i spašavanju nestale ili ozlijeđene osobe, budući da smanjuje vrijeme i troškove pretrage, povećava područje nadzora i sigurnost ekipe za spašavanje. Detekcija ljudi na zračnim slikama izazovna je i zamorna zadaća kako za obučene ljude, tako i za algoritme detekcije zbog varijacija u položaju, preklapanju, razmjeru, veličini i lokaciji na kojoj se osoba može nalaziti na slici, kao i zbog loših uvjeta snimanja, smanjene vidljivosti, zamućenosti zbog kretanja i slično. U ovom radu, generički model detekcije objekata YOLOv8, prethodno obučen na COCO skupu podataka, prilagođava se na prilagođenom SARD skupu podataka koji se koristi za optimizaciju modela za detekciju osoba na zračnim slikama planinskih krajolika snimljenih bespilotnom letjelicom. Različiti modeli algoritama obitelji YOLOv8 prilagođeni SARD skupu eksperimentalno su testirani, a pokazano je da model YOLOv8x postiže najvišu srednju prosječnu preciznost (mAP@0,5:0,95) od 63,8%, uz vrijeme zaključivanja od 4,6 ms, što pokazuje potencijal za stvarnu upotrebu u operacijama traganja i spašavanja. Testirali smo tri algoritma za geolokaciju u stvarnim uvjetima te predložili izmjene i preporuke za korištenje u misijama traganja i spašavanja kako bi se odredila geolokacija osobe snimljene bespilotnom letjelicom nakon automatske detekcije s modelom YOLOv8x.

Dostupno na:
https://www.scitepress.org/PublicationsDetail.aspx?ID=ptBwsiKhlSk=&t=1

Sambolek Saša and Marina Ivašić-Kos. "Person Detection and Geolocation Estimation in UAV Aerial Images: An Experimental Approach." Proceedings of the 13th International Conference on Pattern Recognition Applications and

Methods - ICPRAM; ISBN 978-989-758-684-2; ISSN 2184-4313, SciTePress, pages 785-792. DOI: 10.5220/0012411600003654

4.7 Ostali radovi koji su rezultat istraživanja u okviru doktorata

4.7.1 Lokalizacija osobe i određivanje udaljenosti pomoću raycast metode / Person localization and distance determination using the raycast method

Korištenjem bespilotnih letjelica u akcijama traganja i spašavanja (SAR), detekcija nestalih osoba moguća je tijekom ili nakon leta analizom snimljenog materijala. Međutim, jednako važan je i proces lokalizacije osobe kako bi spašavatelji mogli brzo pristupiti osobi koja treba pomoć. Predlažemo upotrebu metode raycastinga za precizno određivanje lokacije osobe i njezine udaljenosti od bespilotne letjelice, koristeći niz monokularnih slika snimljenih letjelicom. Predloženu metodu smo testirali in silico, koristeći proceduralni simulator prilagođen specifičnim uvjetima leta, uključujući i situacije s vjetrom. Naši rezultati ukazuju da višestruki raycasting rješava problematične telemetrijske podatke te da postoji optimalan broj iteracija potrebnih za preciznu lokalizaciju, ovisno o telemetrijskom šumu specifičnom za svaku bespilotnu letjelicu.

Dostupno na: https://ieeexplore.ieee.org/abstract/document/9566329

Paulin, G., Sambolek, S., & Ivasic-Kos, M. (2021, September). Person localization and distance determination using the raycast method. *In 2021 6th International Conference on Smart and Sustainable Technologies (SpliTech)* (pp. 1-5). IEEE.

4.7.2 Primjena metode raycast za geolokalizaciju osoba i određivanje udaljenosti pomoću slika snimljenih bespilotnom letjelicom u realnim scenarijima traganja i spašavanja na kopnu / Application of raycast method for person geolocalization and distance determination using UAV images in Real-World land search and rescue scenarios

Ljudi vole provoditi vrijeme u divljini iz mnogo razloga. No, povremeno se izgube ili ozlijede, a njihovo preživljavanje ovisi o brzom pronalaženju i spašavanju. Nakon dojave o nesreći, pokreće se akcija traganja i spašavanja (SAR), mobilizirajući sve dostupne resurse. Uključivanjem bespilotnih letjelica u SAR operacije omogućena je primjena računalnog vida za automatsku detekciju osoba na snimkama iz zraka. Pri pretraživanju bespilotnom letjelicom, prednost se daje fotografijama koje obuhvaćaju veće površine unutar jedne slike, što smanjuje vrijeme pretraživanja. No, s takvim fotografijama dolazi i promjena mjerila, što otežava lokalizaciju osobe u stvarnom svijetu i određivanje udaljenosti od letjelice. Kako bismo riješili ovaj izazov, inspirirani našim prethodnim simulacijama, istražili smo primjenu metode raycasta za geolokalizaciju osoba i određivanje udaljenosti u stvarnim scenarijima. U ovom radu predstavljamo sustav koji precizno geolocira osobe automatski detektirane na offline obrađenim slikama snimljenim tijekom SAR misije. Nakon niza eksperimenata na terenima različitih konfiguracija i složenosti, korištenjem prilagođenog 3D generatora terena i raycastera, zajedno s detektorom osoba temeljenim na dubokoj neuronskoj mreži obučenoj na našem prilagođenom skupu podataka, definirali smo metodu za geolokaciju detektiranih osoba pomoću raycast metode. Naša metoda prevladava probleme s kojima su se suočavale prethodne metode i postiže visoku pouzdanost, čak i uz samo 4 uzastopne detekcije. Također, kratko vrijeme obrade omogućuje učinkovitu analizu podataka snimljenih tijekom leta bespilotne letjelice, dokazavši da se predložena metoda može uspješno koristiti u stvarnim SAR misijama. Predložili smo i novu metriku procjene (ErrDist) za geolokalizaciju osoba te dali preporuke za korištenje predloženog sustava u stvarnim scenarijima.

Dostupno na:
https://www.sciencedirect.com/science/article/abs/pii/S0957417423019978

Paulin, G., Sambolek, S., & Ivasic-Kos, M. (2024). Application of raycast method for person geolocalization and distance determination using UAV images in Real-World land search and rescue scenarios. *Expert Systems with Applications*, 2024, 237, 121495. https://doi.org/10.1016/j.eswa.2023.121495.

4.7.3 *Određivanje geolokacije osobe detektirane na slici snimljenoj bespilotnom letjelicom / Determining the geolocation of a person detected in an image taken with a drone*

Detekcija osoba pomoću bespilotnih letjelica postaje sve popularnija u operacijama traganja i spašavanja, upravljanju katastrofama ili praćenju osoba. Međutim, detekcija i geolociranje osobe koja stoji ili se kreće u neurbanom području na temelju slika iz zraka ima problema s malim objektima, složenim scenama i senzorima niske točnosti. Za rješavanje ovih problema, ovaj rad razvija okvir za detekciju i geolociranje osobe te određivanje smjera i brzine kretanja ako je osoba detektirana na više fotografija, korištenjem monokularne kamere, GPS prijamnika i senzora ugrađenih u letjelici. Prvo, metoda temeljena na YOLOv4 modelu dubokog učenja, trenirana na skupu podataka SARD, korištena je za detekciju osoba zbog svoje učinkovitosti i djelotvornosti u detekciji malih objekata u složenim scenama. Zatim je predstavljena metoda pasivne geolokacije za izračunavanje GPS koordinata osobe. Na kraju, na temelju dobivenih podataka, sustav predlaže novo potražno područje. Predloženi sustav testiran je pomoću dva drona DJI Phantom 4 Advanced i DJI Mavic 2 Enterprise Advanced. Eksperimentalni rezultati pokazuju da se na ovaj način može detektirati i geolocirati osobu sa zadovoljavajućom točnošću čak i u slučaju nagnutog terena, koristeći DEM datoteke područja pretraživanja. Sustav pokazuje svoju sposobnost u stvarnom svijetu, sugerirajući njegovu potencijalnu primjenu u operacijama traganja i spašavanja.

Dostupno na: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4373987

Poslano na recenziju: 1. ožujka 2023. još nije stigla recenzija

Sambolek, Sasa and Ivasic-Kos, Marina, Determining the Geolocation of a Person Detected in an Image Taken with a Drone. Available at SSRN: https://ssrn.com/abstract=4373987 or http://dx.doi.org/10.2139/ssrn.4373987

LITERATURA

[1] HGSS, “ Hgss presjek i statistika akcija. ”, https://www.hgss.hr/hgss-presjek-i-statistika-akcija/. [4.8.2024].

[2] Koester, Robert J. " Lost Person Behavior: A Search and Rescue Guide on Where to Look—For Land, Air, and Water ", *dbS Productions: Charlottesville*, VA, USA, 2008.

[3] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards real-time object detection with region proposal networks." *IEEE transactions on pattern analysis and machine intelligence 39*, no. 6 (2016): 1137-1149..

[4] Cai, Zhaowei, and Nuno Vasconcelos. "Cascade r-cnn: Delving into high quality object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition, pp.* 6154-6162. 2018.

[5] Lin, Tsung-Yi, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal loss for dense object detection." *Proceedings of the IEEE international conference on computer vision*, pp. 2980-2988. 2017.

[6] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single shot multibox detector." *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pp. 21-37. Springer International Publishing, 2016.

[7] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint arXiv*:1804.02767 (2018).

[8] Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft coco: Common objects in context." *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740-755. Springer International Publishing, 2014. [9] Everingham, Mark, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. "The pascal visual object classes challenge: A retrospective." *International journal of computer vision* 111 (2015): 98-136.

[10] Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang et al. "Imagenet large scale visual recognition challenge." *International journal of computer vision* 115 (2015): 211-252.

[11] Zhu, Pengfei, Longyin Wen, Dawei Du, Xiao Bian, Qinghua Hu and Haibin Ling. “Vision Meets Drones: Past, Present and Future.” *ArXiv abs*/2001.06303 (2020): n. pag.

[12] Barekatain, Mohammadamin, Miquel Martí, Hsueh-Fu Shih, Samuel Murray, Kotaro Nakayama, Yutaka Matsuo, and Helmut Prendinger.

"Okutama-action: An aerial view video dataset for concurrent human action detection." *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 28-35. 2017.

[13] Yu, Hongyang, Guorong Li, Weigang Zhang, Qingming Huang, Dawei Du, Qi Tian, and Nicu Sebe. "The unmanned aerial vehicle benchmark: Object detection, tracking and baseline." *International Journal of Computer Vision* 128 (2020): 1141-1159.

[14] Sambolek, Sasa, and Marina Ivasic-Kos. "Automatic person detection in search and rescue operations using deep CNN detectors." *Ieee Access* 9 (2021): 37905-37922.

[15] Sambolek, Saša, and Marina Ivašić-Kos. "Detection of toy soldiers taken from a bird's perspective using convolutional neural networks." *ICT Innovations 2019. Big Data Processing and Mining: 11th International Conference, ICT Innovations 2019, Ohrid, North Macedonia, October 17–19, 2019, Proceedings 11*, pp. 13-26. Springer International Publishing, 2019.

[16] Sambolek, Saša, and Marina Ivašić-Kos. "Detecting objects in drone imagery: a brief overview of recent progress." *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, pp. 1052-1057. IEEE, 2020.

[17] Sambolek, Sasa, and Marina Ivasic-Kos. "Person detection in drone imagery." *2020 5th International Conference on Smart and Sustainable Technologies (SpliTech)*, pp. 1-6. IEEE, 2020.

[18] Sambolek, Saša, and Marina Ivašić-Kos. "Transfer learning methods for training person detector in drone imagery." *Intelligent Systems and Applications: Proceedings of the 2021 Intelligent Systems Conference (IntelliSys) Volume 2*, pp. 688-701. Springer International Publishing, 2022.

[19] Sambolek Saša and Marina Ivašić-Kos. "Determining the Geolocation of a Person Detected in an Image Taken with a Drone," doi: http://dx.doi.org/10.2139/ssrn.4373987.

[20] Sambolek Saša and Marina Ivašić-Kos. "Person Detection and Geolocation Estimation in UAV Aerial Images: An Experimental Approach." *Proceedings of the 13th International Conference on Pattern Recognition Applications and Methods* - ICPRAM; ISBN 978-989-758-684-2; ISSN 2184-4313, SciTePress, pages 785-792. DOI: 10.5220/0012411600003654

[21] LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE 86*, no. 11 (1998): 2278-2324.

[22] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in*

neural information processing systems 25 (2012).

[23] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.

[24] Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision." *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818-2826. 2016.

[25] Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv*:1704.04861 (2017). [26] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." *arXiv preprint arXiv*:2004.10934 (2020).

[27] “Yolo-v4 and Yolo-v3/v2 for Windows and Linux.” https://github.com/AlexeyAB/darknet. [8.4.2024].

[28] “Phantom 4 Advanced - Specs.” https://www.dji.com/hr/phantom-4-adv/info#specs. [5.5.2021]..

[29] “Mavic 2 Enterprise Advanced - Specs - DJI.” https://www.dji.com/hr/mavic-2-enterprise-advanced/specs [16.12.2022].

[30] Paulin, Goran, Sasa Sambolek, and Marina Ivasic-Kos. "Person localization and distance determination using the raycast method." *2021 6th International Conference on Smart and Sustainable Technologies (SpliTech)*, pp. 1-5. IEEE, 2021.

[31] Paulin, Goran, Sasa Sambolek, and Marina Ivasic-Kos. "Application of raycast method for person geolocalization and distance determination using UAV images in Real-World land search and rescue scenarios." *Expert Systems with Applications* 237 (2024): 121495.

POPIS SLIKA

POPIS TABLICA

ŽIVOTOPIS

Saša Sambolek, rođen 1982. godine u Kutini (Hrvatska). Osnovnu i srednju školu završio je u Kutini. 2009. godine na Prirodoslovno matematičkom fakultetu u Zagrebu, na Fizičkom odsjeku stekao je zvanje profesora fizike i informatike. Radi u srednjoj školi. Godine 2014. započeo je doktorski studij informatike na Fakultetu informatike i digitalnih tehnologija Sveučilišta u Rijeci. Oženjen, otac troje djece.

Područje znanstvenog istraživanja usmjereno je na umjetnu inteligenciju i računalni vid.

Popis objavljenih radova:

1. Sambolek, Saša, and Marina Ivašić-Kos. "Detection of toy soldiers taken from a bird's perspective using convolutional neural networks." ICT Innovations 2019. Big Data Processing and Mining: 11th International Conference, ICT Innovations 2019, Ohrid, North Macedonia, October 17–19, 2019, Proceedings 11. Springer International Publishing, 2019.
2. Sambolek, Saša, and Marina Ivašić-Kos. "Detecting objects in drone imagery: a brief overview of recent progress." 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO). IEEE, 2020.
3. Sambolek, Sasa, and Marina Ivasic-Kos. "Person detection in drone imagery." 2020 5th International Conference on Smart and Sustainable Technologies (SpliTech). IEEE, 2020.
4. Sambolek, Sasa, and Marina Ivasic-Kos. "Automatic person detection in search and rescue operations using deep CNN detectors." *IEE Access* 9 (2021): 37905-37922.
5. Sambolek, Saša, and Marina Ivašić-Kos. "Transfer learning methods for training person detector in drone imagery." Intelligent Systems and Applications: Proceedings of the 2021 Intelligent Systems Conference (IntelliSys) Volume 2. Springer International Publishing, 2022.
6. Sambolek Saša and Marina Ivašić-Kos. "Person Detection and Geolocation Estimation in UAV Aerial Images: An Experimental

Approach." Proceedings of the 13th International Conference on Pattern Recognition Applications and Methods - ICPRAM; ISBN 978-989-758-684-2; ISSN 2184-4313, SciTePress, pages 785-792.

7. Paulin, Goran, Sasa Sambolek, and Marina Ivasic-Kos. "Person localization and distance determination using the raycast method." 2021 6th International Conference on Smart and Sustainable Technologies (SpliTech). IEEE, 2021.
8. Paulin, Goran, Sasa Sambolek, and Marina Ivasic-Kos. "Application of raycast method for person geolocalization and distance determination using UAV images in Real-World land search and rescue scenarios." *Expert Systems with Applications* 237 (2024): 121495., Elsevier
9. Sambolek, Saša, and Marina Ivašić-Kos. "Person Detection and Geolocation Estimation in UAV Aerial Images: An Experimental Approach." Proceedings of the 13th International Conference on Pattern Recognition Applications and Methods - ICPRAM; ISBN 978-989-758-684-2; ISSN 2184-4313, SciTePress, pages 785-792. DOI: 10.5220/0012411600003654

II. UKLJUČENE PUBLIKACIJE

RAD 1. DETECTION OF TOY SOLDIERS TAKEN FROM A BIRD'S PERSPECTIVE USING CONVOLUTIONAL NEURAL NETWORKS

Ovaj rad je objavljen kao: Sambolek, Saša, and Marina Ivašić-Kos. "Detection of toy soldiers taken from a bird's perspective using convolutional neural networks." ICT Innovations 2019. Big Data Processing and Mining: 11th International Conference, ICT Innovations 2019, Ohrid, North Macedonia, October 17–19, 2019, Proceedings 11, pp. 13-26. Springer International Publishing, 2019.

This research was supported by Croatian Science Foundation under the project IP-2016-06-8345 "Automatic recognition of actions and activities in multimedia content from the sports domain" (RAASS) and by the University of Rijeka under the project number 18-222-1385.

https://link.springer.com/chapter/10.1007/978-3-030-33110-8_2

1. Introduction

Convolutional neural network (CNN) [21] is a particular architecture of artificial neural networks, proposed by Yann LeCun in 1988. The key idea of CNN is that the local information in the image is important for understanding the content of the image so a filter is used when learning the model, focusing on the image, part by part, as a magnifying glass. The practical advantage of such approach is that CNN uses fewer parameters than fully-connected neural networks, which significantly improves learning time and reduces the amount of data needed to train the model.

Recently, after AlexNet [22] popularized deep neural networks by winning ImageNet competitions, convolutional neuronal networks have become the most popular model for image classification and object detection problems. Image classification predicts the existence of a class in a given image based on a model that is learned on a set of labeled images. There are several challenges associated with this task, including differences between objects of the same class, similarities between objects of different classes, object occlusions, different object sizes, various backgrounds. The appearance of an object on the image might change due to lighting conditions, position (height, angle) of the camera and distance from the camera and similar [19]. The detection of an object beside the prediction of the class to which the object belongs, provides information about its location in the image, so the challenge is to solve both the classification and location task. The detected object is most often labeled with the bounding box [23], but there are also detectors that segment objects at the pixel level and mark the object using its silhouette or shape [14, 5].

Some of today's most widely used deep convolution neural networks are Faster R-CNN, RFCN, SSD, Yolo, RetinaNet. These networks are unavoidable in tasks such as image classification [22] and object detection [26], analysis of sports scenes and activities of athletes [6], disease surveillance [25], surveillance and detection of suspicious behavior [2019], describing images [17], development and management of autonomous vehicles in robotics [10], and the like.

In this paper, we have focused on the problem of detecting small objects on footage taken by the camera of a mobile device or drones from a bird's eye view. These footages are today widely used when searching for missing persons in non-urban areas, border control, animal movement control, and the like.

In [1], drones were used to locate missing persons in search and rescue operations. Authors have used HOG descriptors [8]. In [3] the SPOT system is de-scribed. It uses an infrared camera mounted on an Unmanned Aerial Vehicle and Faster R-CNN to detect villains and control animals in images. A modified MobileNet architecture was used in [9] for body detection and localization in the sea. Images were shot both with an optical camera and a multi-spectral camera. In [33] YOLO was used for detection of objects on images taken from the air. In [2424], three models of deep neural networks (SSD, Faster R-CNN, and RetinaNet) were analyzed for detection tasks on images collected by crewless air-craft. The authors showed that RetinaNet was faster and more accurate when detecting objects. The dependence analysis of Faster R-CNN, RFCN, and SSD speed and precision in case of running on different architectures was given in [18].

In this paper, we will approximate the problem of detecting small objects on bird-eye viewings or drone shots with the problem of detecting toy soldiers captured by the camera of a mobile device.

The rest of the paper is organized as follows: in Section II. we will present the architecture of CNN networks, ResNet50, Inception and MobileNet with Faster R-CNN and SSD localization methods that are used in our research. We have examined their performance on a custom toy soldiers' dataset. The comparison of the detector performance and discussion are given in Section III. The paper ends with a conclusion and the proposal for future research.

2. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are adapted to solve the problems of high-dimensional inputs and inputs that have many features such as in cases of image processing and object classification and detection. The CNN network

consists of a convolution layer, after which the network has been named, activation and pooling layers, and at the end is most often one or more fully connected layers.

The convolution layer refers to a mathematical operator defined over two functions with real value arguments that give a modified version of one of the two original functions. The layer takes a map of the features (or input image) that convolves with a set of learned parameters resulting in a new two-dimensional map. A set of learned parameters (weights and thresholds) are called filters or kernels. The filter is a 2D square matrix, small in size compared to the image to which it is applied (equal depths as well as the input). The filter consists of real values that represent the weights that need to be learned, such as a particular shape, color, edge in order to give the network good results.

The pooling layer is usually inserted between successive convolution layers, to re-duce map resolution and increase spatial invariance - insensitivity to minor shifts (rotations, transformations) of features in the image as well as to reduce memory requirements for the implementation of the network. Along with the most commonly used methods (arithmetic mean and maximum [44]), there are several pooling methods used in CNN, such as Mixed Pooling, Lp Pooling, Stochastic Pooling, Spatial Pyramid Pooling and others [13].

The activation function propagates or stops the input value in a neuron depending on its shape. There is a broader range of neuron activation functions such as linear activation functions, jump functions, and sigmoidal functions. The jump functions and sigmoidal functions are a better choice for neural networks that perform classification while linear functions are often used in output layers where unlimited output is required. Newer architectures use activation functions behind each layer. One of the most commonly used activation functions in CNN is the ReLU (Rectified Linear Unit). In [13], the activation functions used in recent works are presented: Leaky Relu (LReLU), Parametric ReLU (PReLU), Randomized ReLU (RReLU), Exponential Linear Unit (ELU) and others.

A fully connected layer is the last layer in the network. The name of the fully connected layer indicates its configuration: all neurons in this layer are linked to

all the outputs of the previous layer. Fully connected layers can be viewed as special types of convolution layers where all feature maps and all filters are 1 x 1.

Network hyperparameters are all parameters needed by the network and set before the network provides data for learning [2]. The hyper-parameters in convolution neural networks are learning rate, number of epochs, network layers, activation function, initialization weight, input pre-processing, pooling layers, error function.

Selecting the CNN network for feature extraction plays a vital role in object detection because the number of parameters and types of layers directly affect the memory, speed, and performance of the detector. In this paper, three types of networks have been selected for feature extraction: ResNet50, Inception, and MobileNet.

2.1 ResNet

ResNet50 is a 50-layer Residual Network. There are other variants like ResNet101 and ResNet152 also [15]. The main innovation of ResNet is the skip connection. The skip connection in the Fig. 1 is labeled “identity.” It allows the network to learn the identity function that allows passing the input through the block without passing through the other weight layers. This allows stacking additional layers and building a deeper network, as well as overcoming the vanishing gradient problem by allowing network to skip through layers if it feels they are less relevant in training. Vanishing gradients often occurs in deep networks if no adjustment is performed because during backpropagation gradient gets smaller and smaller and can make learning difficult.

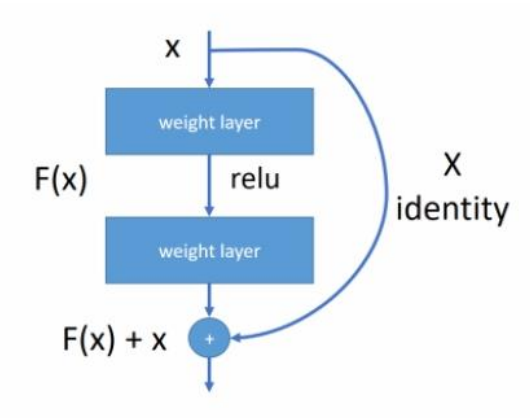


Figure 1. A residual block, according to [15]

2.2 Inception

GoogLeNet has designed a module called Inception that approximates a sparse CNN with a normal dense construction, Fig. 2. The idea was to keep a small number of convolutional filters taking into account that only a small number of neurons are effective. The convolution filters of different sizes (5x5, 3x3, 1x1) were used to capture details on varied scales. In the versions Inception v2 and Inception v3, the authors have proposed several upgrades to increase the accuracy and reduce the computational complexity [28, 29].

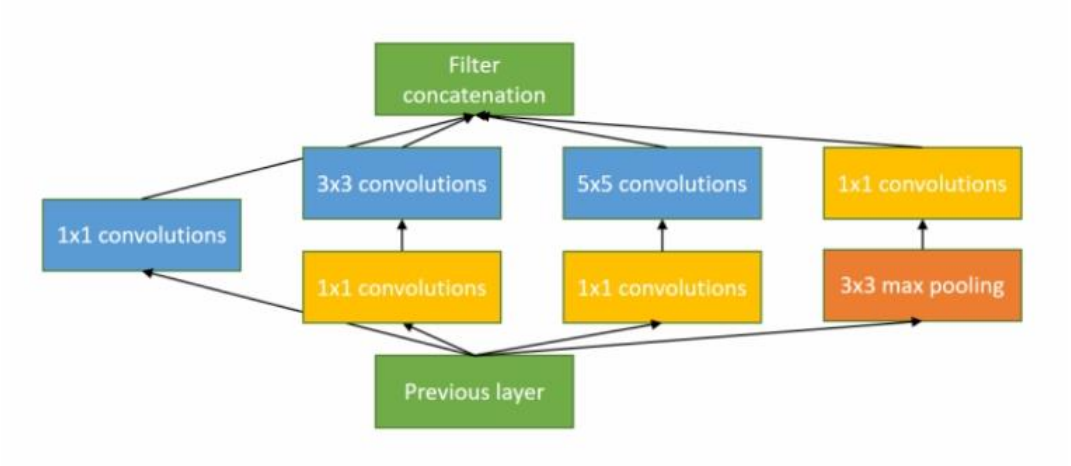


Figure 2. Inception module, according to [28]

2.3 MobileNet

MobileNet is a lightweight architecture designed for use in a variety of mobile applications [16]. It filters the input channels by running a single convolution on

each color channel instead of combining all three channels and flattening them all.

2.4 Faster R-CNN

In the earlier version of R-CNN [11] and Fast R-CNN [12], region proposals are generated by selective search (SS) [31] rather than using convolutional neural network (CNN). The SS algorithm was the "bottleneck" in region proposal process, so in the Faster R-CNN a separate convolution network (RPN) is used to propose regions. The RPN network then checks which location contains the object. Appropriate locations and bounding boxes are sent to the detection network that determines the class of the object and returns the bounding box of that object. This kind of design has speed up the object detection.

2.5 Single Shot Detector

The Single Shot Detector (SSD) method for objects detection uses deep network that omits the stage of bounding box proposal and allows features extraction without losing accuracy. The approach assumes that potential objects can be located within the predefined bounding box of different size and side ratios centered in each location of feature map. The network for each bounding box determines the probability measure for the presence of each of the possible categories and adjusts the position of the box to frame the object better. In order to overcome the problems inherent in the difference in object sizes, the network makes decisions by combining prediction from several feature maps of different dimensions [23].

3. Comparison of SSD and Faster RCNN detection performance on scenes of toy soldiers

We have tested and compared the accuracy of the object detector for a class of person (toy soldier) at different scene configurations, changing the number of objects, their position, background complexity and lighting conditions. The goal is to select the appropriate model for future research on the detection of missing persons in rescue operations.

We used publicly available pre-trained models with corresponding weights learned on the Microsoft's common object and context (COCO) dataset [7] by transfer learning and fine-tune the model parameters on our data set.

We used the Tensorflow implementation [30] of the CNN model and the Python programming language in the Windows 10 x64 environment. All models were trained on a laptop with the i5-7300HQ CPU and the Ge-Force GTX 1050Ti 4GB GPU. The number of epochs and the training time differs among models and depends on loss. The parameters of each model have not been changed and were equal to the parameters of the original model.

3.1 Data Preprocessing

The data set contains 386 images shot by a mobile device camera (Samsung SM-G960F) at a 2160x2160px resolution, without using a tripod. Each image contains multiple instances of toy soldiers, taken under different angles and different lighting conditions with a different background type from a uniform to complex (such as grassy surfaces). The images are divided into a learning and test set in a cutoff of 80:20, and their resolution is reduced to 720x720px. In total, there are 798 toy soldiers in the images, of which 651 are in learning set and 147 in test.

The LabelImg tool was used to plot bounding box and create responsive XML files with stored xmin, xmax, ymin, ymax position for each layout. Images and corresponding XML files are then converted to TFRecord files that are implemented in the Tensorflow environment. TFRecord files merge all the images and notes into a single file, thus reducing the training time by eliminating the need of opening each file.

3.2 Methods

SSD with MobileNet

This method uses SSD for detection while the MobileNet network is used as a feature extractor. The output of MobileNets is processed using the SSD. We have tested the detection results of two versions of the MobileNet network (V1 and V2), referred to as ssd_mobilenet_v1 and ssd_mobilenet_v2. Both networks were pre-

trained (ssd_mobilenet_v1_coco_2018_01_28 and ssd_mobilenet_v2_coco_2018_03_29) on COCO dataset of 1.5 million objects (80 categories) in 330,000 images. We trained the network using toy soldier's images width bounding box as input to the training algorithm. The network parameters include: prediction dropout probability 0.8, kernel size 1 and a box code size set to 4. The root mean square propagation optimization algorithm is used for optimizing the loss function with learning rate of 0.004 and decay factor 0.95. At the non-maximum suppression part of the network a score threshold of 1×10^{-8} is used with an intersection of union threshold of 0.6, both the classification and localization weights are set to 1. Ssd_mobilenet_v1 was trained for 17,105 steps and ssd_mobilenet_v2 for 10,123 steps.

SSD with Inception-V2

The combination of SSD and Inception-V2 is called SSD-Inception-V2. In this case, SSD is used for detection while Inception-V2 extracts features. We trained the network using predefined ssd_inception_v2_coco_2018_01_28 weights. The training process uses similar hyperparameters as SSD with MobileNet, except in this case of the kernel size that is set to 3. The network was trained for 6,437 steps.

SSDLite with MobileNet-V2

SSDLite [27] is a mobile version of the regular SSD, so all regular convolutions with detachable convolutions are replaced (depthwise followed by 1 × 1 projection) in SSD layers. This design is in line with the overall design of MobileNet and is considered to be much more efficient. Compared to SSD, it significantly reduces the number of parameters and computing costs. We trained the network using pre-trained ssdlite_mobilenet_v2_coco_2018_05_09 weights. Similar hyperparameters were used as before, and the network was trained for 14,223 steps.

Faster R-CNN with ResNet50

Faster R-CNN detection involves two phases. The first phase requires a region proposal network (RPN) that allows simultaneous prediction of object anchors

and confidence (objectiveness) from some internal layers. For this purpose, a residual network with a depth of 50 layers (ResNet50) is used. The grid anchor size was 16 x 16 pixels with scales [0.25, 0.5, 1.0, 2.0], a non-maximum-suppression-IoU threshold was set to 0.7, the localization loss weight to 2.0, objectiveness weight to 1.0 with an initial crop size of 14, kernel size was 2 with strides set to 2. The second phase requires information from the first phase to predict the class label and the bounding box. We trained the network using pre-trained faster_rcnn_resnet50_coco_2018_01_28 weights. The IoU threshold for prediction score was set to 0.6; the momentum (SGD) optimizer for optimizing the loss functions has initial learning rate set to 0.0002 and momentum value 0.9. The network was trained for 12,195 steps.

Faster R-CNN with Inception-V2

Faster R-CNN uses the Inception V2 feature extractor to get features from the input image. The middle layer of the Inception module uses the RPN network component to predict the object anchor and confidences. As in previous cases, the network was trained with pre-trained fast-er_rcnn_inception_v2_coco_2018_01_28 weights. Simi-lar hyperparameters were used as in case of Faster R-CNN with ResNet50 and the learning process lasted for 33,366 steps.

3.3 Results and discussion

We compared the results of the SSD model and the Faster RCNN object detector based on CNNs on our toy soldiers test set concerning mean average precision (mAP) [32]. A detection is considered as true positive when more than half of the area belonging to the soldier is inside the detected bounding box. Detectors performance are also evaluated in terms of recall, precision and F1 score.

$$F1 = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{(\text{Recall} + \text{Precision})} \quad (1)$$

Fig. 3. shows a comparison of results of models that were additionally learned on our learning set with original models trained on the COCO dataset. The results show a significant increase in the average precision of all models after training on our dataset. The best results of over 96% were achieved with the faster_rcnn

network. The implementation of faster_rcnn with Resnet50 proved to be somewhat successful than architecture with the Inception Network. Faster_rcnn has also shown the best classification results concerning F1 score and Recall [19], Fig. 4. All classification result of all models in terms of precision, recall and F1 score are shown in Fig. 4.

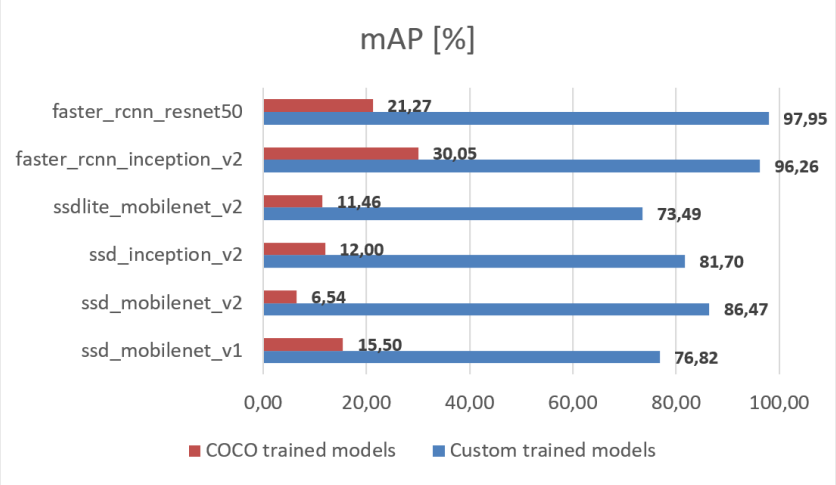


Fig. 3. Comparison of the evaluation result of the toy soldier's detection

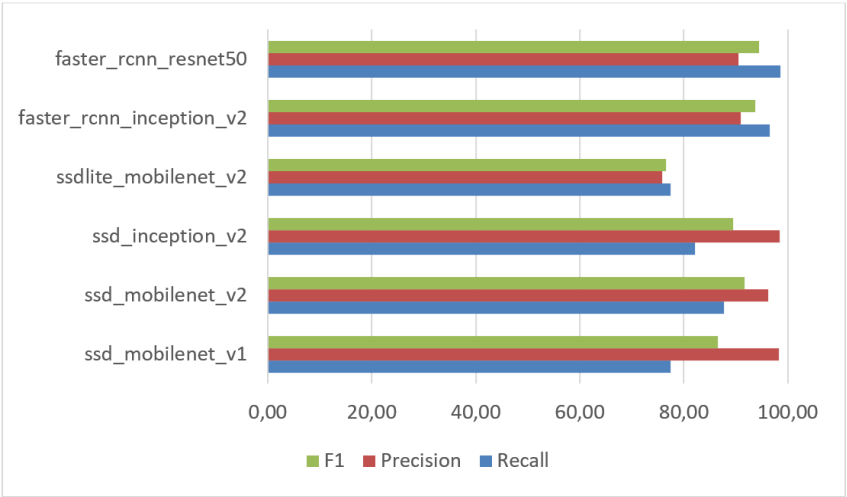


Fig. 4. Comparison of the results of the trained model detection concerning the F1, Precision and Recall metrics

Fig. 5. shows the time required to train the model on our learning set. The least amount of time was needed to learn the faster_rcnn model. The longest, more than 3.5 times longer than learning the fast_rcnn model, was needed to learn the ssdlitle-mobilenet_v2 model.

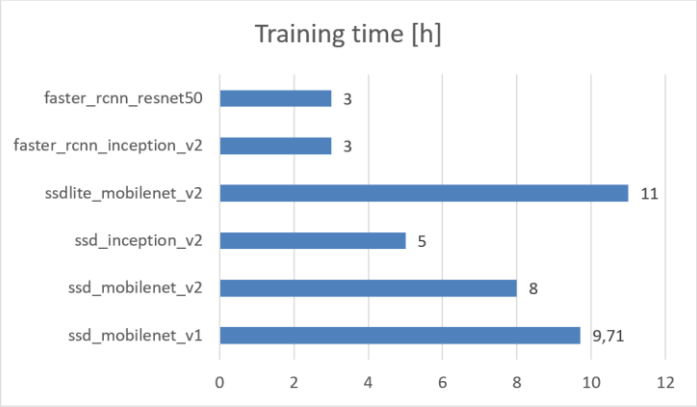


Fig. 5. Comparison of model learning time on the custom dataset

Model performances are additionally presented in two scenarios: simple and complex. The simple scenario has a uniform background color and up to 8 visible objects near the camera, which may overlap. A complex scenario is considered when the number of objects in a scene is equal to and greater than 9, away from the camera and with occlusions. A Fig. 6. shows an example of the detection results in the case of a simple scenario. The images marked A through F show the same scenarios with a uniform background with a wooden pattern and five soldiers in different poses such as walking with a gun, shooting, crawling and lying down.

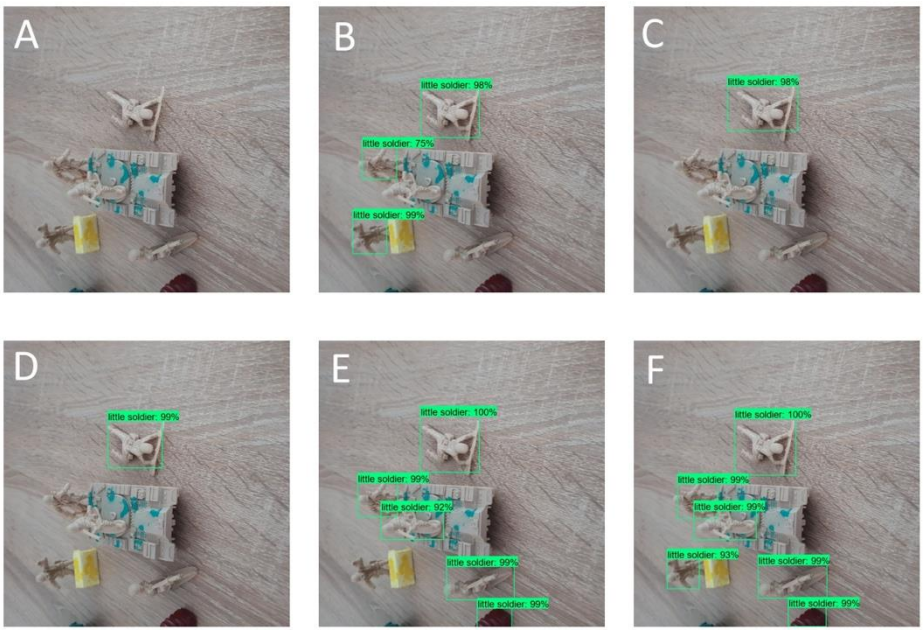


Figure 6. The detection results in the case of a simple scenario

In all the images, the results of individual models are indicated in the following way:

- A – ssd_mobilenet_v1,
- B – ssd_mobilenet_v2,
- C – ssdlite_mobilenet_v2,
- D – ssd_inception_v2,
- E – faster_rcnn_resnet_50,
- F – faster_rcnn_inception_v2.

In figure A, no soldier was detected, B has 3 of 5 true positive (TP) detections, C and D only one detected soldier, while E has 4 TP with one false positive (FP) detection, and F all positive detections with one FP.

In the case of a uniform background with higher contrast to soldiers, as in Fig. 7. all models have detected with greater success in comparison to the previous case, even though in this example, we have a higher number of soldiers and at a greater distance. There were 11 soldiers on the scene, but no model detected a soldier on a tank of the identical color. The best results were achieved in E and F images with 10 successful detections, then model B with 7, and then models A and C follow. The sequence of the success of the model is similar to the one in the previous example.

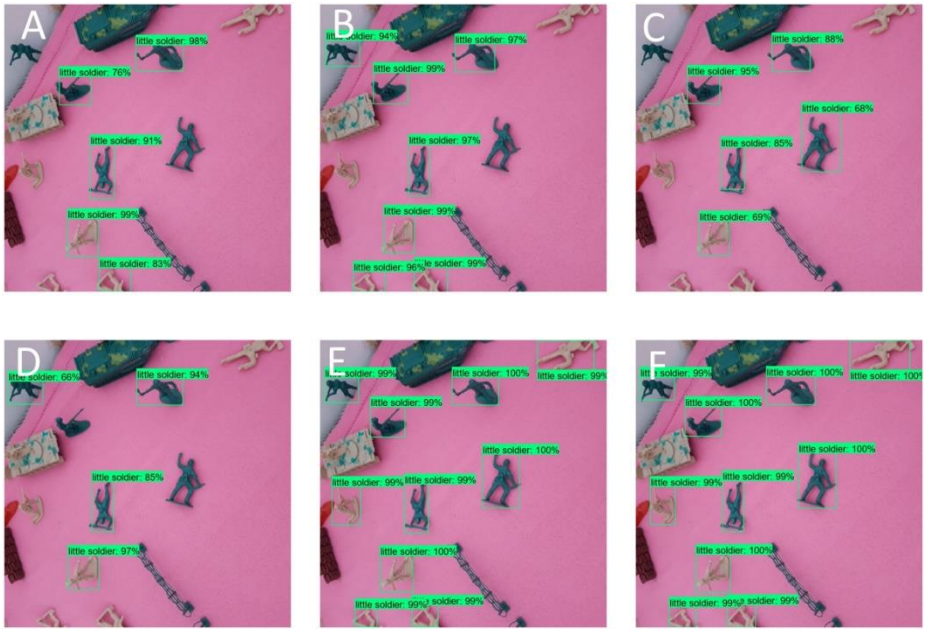


Figure 7. The case of a uniform background with higher contrast to soldiers

Fig. 8. shows a complex scene in which soldiers are partially covered with grass. The camera's position is not as in the previous cases from the top, but from the side. The models in Figures A and D did not have any detection, while B detected almost the entire image as a soldier. C has one positive and two false detections, E has repeatedly detected the same object, but with different rectangle size and has a false detection, and F has an accurate, true positive detection.

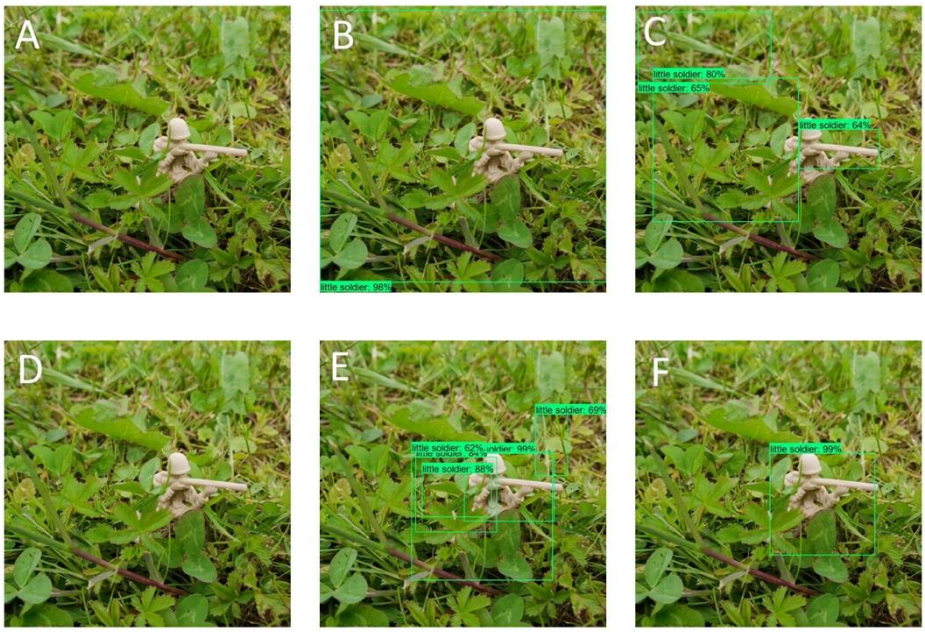


Figure 8. A complex scene in which soldiers are partially covered with grass

Fig. 9. shows two scenes with a camera positioned from a bird's perspective at a greater distance than in earlier cases and with 8 soldiers on a uniform blue background. Model A detected only one soldier (Fig. 9.a) whereas B, C, and D had no detection (Fig. 9.b). Model E detected all soldiers (9.d), while F detects all soldiers plus three false detections (9.e).

In the second scene recorded from a greater distance on the grass, Fig. 9.c and 9.f, only F detects one soldier out of 7 possible. Examples show that all models have prob-lems with object detection when an object is less than 50px in height or width, espe-cially when the contrast of the subject and background is not significant, and when the background is more complex than in the case of grass.

Fig. 10 is an example of a scene with two soldiers with a cluttered background. E and F models detected both soldiers with a probability of detection of 100% (Fig. 10.a, Fig. 10.c, and Fig. 10.e), while model B detected only one soldier (Fig. 10.b, Fig. 10.d.) and other models failed to detect anything. Fig. 10.e shows a higher contrast between the soldiers and the background, but this did not help models A, B, C, D to have a successful detection. Fig. 10.f shows the occlusion of soldiers; however, models B, D, E, and F were able to detect them.

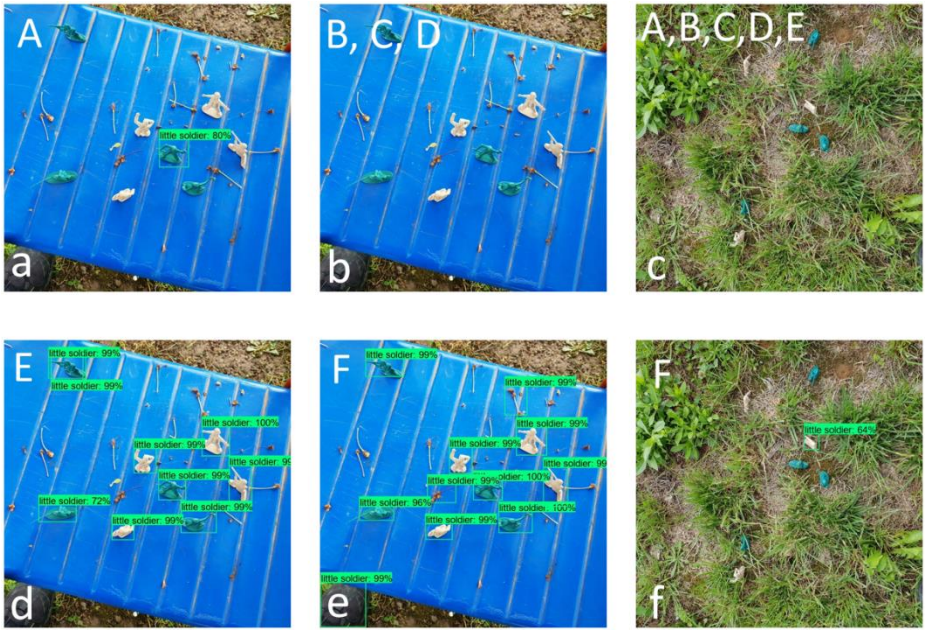


Figure 9. Two scenes with a camera positioned from a bird's perspective

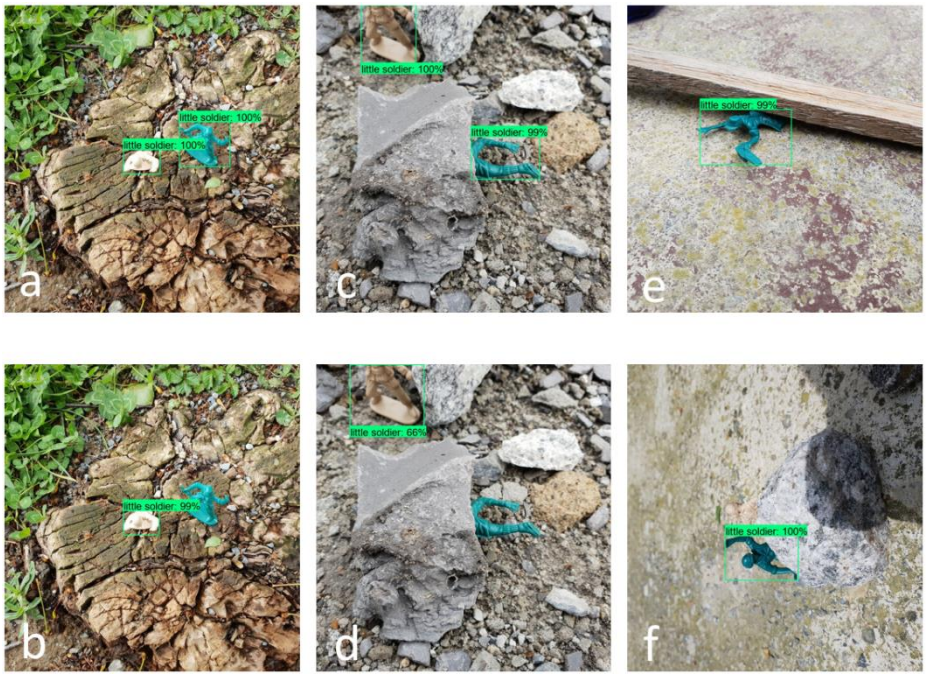


Figure 10. A scene with a cluttered background and the occlusion of soldiers

4. Conclusion

Recordings taken from the air today are used mainly in search of missing persons, in mountain rescue, in the control of animal movement, and the like. The ability to automatically detect persons and objects on the images taken from a bird's perspective would greatly facilitate the search and rescue of people or the control of people and animals.

CNN networks have proven successful in classification and object detection tasks on general-purpose images, and in this paper, we have tested their performance in detecting toy soldiers taken from the bird's eye view. On the custom dataset, we compared the performance of ResNet50, Inception, and MobileNet networks with Faster RCNN and SSD methods of localization. The analysis of the obtained results shows that Faster RCNN is more suitable for detection because it detects toy soldiers more successfully. The configuration with the Inception network is more successful than the configuration with ResNet50. The problem with this method is that it requires more time and computation power.

The examples also show the background effect on detection accuracy. With a uniform background and higher color contrast, detection of all models is significantly successful than in case of detection at a greater distance, on the grass, and with semi-hidden objects. In future work, we will try to find a way to solve this problem.

This paper provides a promising base ground for further research in real-time detection of missing persons in search and rescue operations. We plan to investigate the further use of different detection methods (speed, accuracy) on the android system.

References

[1] Andriluka, M., Schnitzspan, P., Meyer, J., Kohlbrecher, S., Petersen, K., Von Stryk, O., ... & Schiele, B.: Vision-based victim detection from unmanned aerial vehicles. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1740-1747 (2010)

[2] Bengio, Y.: Practical recommendations for gradient-based training of deep architectures. In Neural networks: Tricks of the trade pp. 437-478 (2012)

[3] Bondi, E., Fang, F., Hamilton, M., Kar, D., Dmello, D., Choi, J., ... & Nevatia, R.: Spot poachers in action: Augmenting conservation drones with automatic detection in near real-time. In Thirty-Second AAAI Conference on Artificial Intelligence. (2018)

[4] Boureau, Y., Ponce, J., & LeCun, Y.: A theoretical analysis of feature pooling in vision algorithms. In Proc. International Conference on Machine learning (Vol. 28), (2010)

[5] Burić, M., Pobar, M., Ivašić-Kos, M.: Ball Detection using Yolo and Mask R-CNN, In 2018 International Conference on Computational Science and Computational Intelligence (CSCI'18), pp. 319-323, Las Vegas (2018)

[6] Burić, M., Pobar, M., Ivašić-Kos, M.: Adapting YOLO Network for Ball and Player Detection, Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM, SciTePress, pp. 845-851, Prag (2019)

[7] Cocodataset, http://cocodataset.org/, last accessed 2019/5/25

[8] Dalal, N., & Triggs, B.: Histograms of oriented gradients for human detection. In international Conference on computer vision & Pattern Recognition, pp. 886-893 (2005)

[9] Gallego, A. J., Pertusa, A., Gil, P., & Fisher, R. B.: Detection of bodies in maritime rescue operations using unmanned aerial vehicles with multispectral cameras. Journal of Field Robotics.

[10] Gao, H., Cheng, B., Wang, J., Li, K., Zhao, J., and Li, D.: Object Classification Using CNN-Based Fusion of Vision and LIDAR in Autonomous Vehicle Environment, in IEEE Transactions on Industrial Informatics, vol. 14, no. 9, pp. 4224-4231 (2018)

[11] Girshick, R., Donahue, J., Darrell, T., & Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580-587 (2014)

[12] Girshick, R.: Fast R-CNN. In Proceedings of the IEEE international conference on computer vision, pp. 1440-1448 (2015)

[13] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... & Chen, T.: Recent advances in convolutional neural networks. Pattern Recognition, 77, pp. 354-377 (2018)

[14] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN, 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2980-2988, Venice (2017)

[15] He, K., Zhang, X., Ren, S., & Sun, J.: Deep residual learning for image recognition. In Proc. of the IEEE conference on computer vision and pattern recognition, pp. 770-778 (2016).

[16] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications, (2017)

[17] Hrga, I., Ivašić-Kos, M.: Deep Image Captioning: An Overview, In IEEE MIPRO, Opatija (2019)

[18] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Murphy, K.: Speed/accuracy trade-offs for modern convolutional object detectors.

IEEE Conference on computer vision and pattern recognition, pp. 7310-7311 (2017)

[19] Ivašić-Kos, M., Ipšić, I., Ribarić, S.: A knowledge-based multi-layered image annotation system. Expert systems with applications. 42, pp. 9539-9553 (2015)

[20] Ivasic-Kos, M., Kristo, M., Pobar, M.: Human detection in thermal imaging using YOLO, In ICCTA 2019, Istanbul (2019)

[21] Johnson, J., Karpathy, A.: Convolutional Neural Networks, Stanford Computer Science, https://cs231n.github.io/convolutional-networks, last access 2019/3/11.

[22] Krizhevsky, A., Sutskever, I., Hinton, G. E.: Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pp. 1097-1105 (2012)

[23] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C.: Ssd: Single shot multi-box detector. In European conference on computer vision, pp. 21-37 (2016)

[24] Radovic, M., Adarkwa, O., Wang, Q.: Object recognition in aerial images using convolutional neural networks. Journal of Imaging (2017)

[25] Ramcharan, A., McCloskey, P., Baranowski, K., Mbilinyi, N., Mrisho, L., Ndalahwa, M., & Hughes, D.: Assessing a mobile-based deep learning model for plant disease surveillance (2018)

[26] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pp. 91-99 (2015)

[27] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L. C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510-4520 (2018)

[28] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A.: Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9 (2015)

[29] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z.: Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818-2826 (2016)

[30] Tensorflow object detection models zoo, https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md, last accessed 2019/5/25

[31] Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W.: Selective search for object recognition. International journal of computer vision,104(2), 154-171 (2013)

[32] Visual Object Classes Challenge 2012 (VOC2012), http://host.robots.ox.ac.uk/pas-cal/VOC/voc2012/, last accessed 2019/5/25

[33] Wang, X., Cheng, P., Liu, X., & Uzochukwu, B.: Fast and Accurate, Convolutional Neural Network Based Approach for Object Detection from UAV. In IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society, pp. 3171-3175 (2018)

RAD 2. DETECTING OBJECTS IN DRONE IMAGERY: A BRIEF OVERVIEW OF RECENT PROGRESS

Ovaj rad je objavljen kao: Sambolek, Saša, and Marina Ivašić-Kos. "Detecting objects in drone imagery: a brief overview of recent progress," 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 2020, pp. 1052-1057, doi: 10.23919/MIPRO48935.2020.9245321.

Radi jasnoće, rad je preoblikovan, inače je sadržaj isti kao i objavljena javno dostupna verzija rada.

http://www.mipro.hr/LinkClick.aspx?fileticket=YNSS8zv%2fWnU%3d&tabid=196&language=hr-HR (stranice 1350 – 1355)

This research was supported by Croatian Science Foundation under the project IP-2016-06-8345 “Automatic recognition of actions and activities in multimedia content from the sports domain” (RAASS) and the project IP-2018-01- 7619 “A Knowledge-based Approach to Crowd Analysis in Video Surveillance (KACAVIS) and by the University of Rijeka (project number 18-222).

https://ieeexplore.ieee.org/abstract/document/9245321

1. Introduction

With the development of technology, unmanned aerial vehicles (UAVs, drones) equipped with cameras find their application in industry, agriculture, surveillance, and search and rescue operations. Detecting objects on drone images is an extremely useful and still under researched problem.

When flying at low altitudes, drone records more details on objects of interest, while larger altitudes cover a larger area. Detecting objects in drone imagery creates greater challenges than traditional panorama detection. One of the reasons is the change in shooting height which significantly affects the size of the desired object or change of the shooting angle [1]. In a single video, in a very short time, a drone can record an object from the front, side, or a bird's eye view. Changes in lighting (day, night) and weather (sunny, cloudy, foggy, or rainy) drastically affect the visibility and display of an object [1]. In addition to all of the above, the challenge in detecting and monitoring is also posed by the rapid movements of the camera, the occlusion, and the relative movement between the camera and the object.

As objects captured by a drone are often too small to be detected by the human eye on a drone control screen, object detection needs to be automated. In recent years, considerable progress has been made in detecting objects using deep learning (convolutional neural networks). The most popular deep learning-based detectors are Faster R-CNN [1], SSD [4], YOLO [5], and RetinaNet [6] trained on datasets like PASCAL VOC [7] or MS COCO [8]. However, it turns out that they are not equally successful when applied to drone-recorded images.

In order to improve the results of object detection on drone imagery, it is necessary to include these images in a training set. However, until recently, there were no publicly available datasets recorded by the drones. With datasets like Campus [9], UAV123 [10], CARPK [11], Okutama-action [12], UAVDT [13] and VisDrone [14], images taken with the drones are available, but there are tailored

to specific issues such as parking control, traffic monitoring, or movement of people across pedestrian zones.

This paper aims to provide a simple but comprehensive overview of object detection on imagery recorded by drones to study existing databases and models that could be used in the detection of persons in search and rescue operations.

During the search and rescue operation, it is important to find the missing person as quickly as possible, as the survival of the missing person declines exponentially over time [15]. The weather and light conditions vary greatly between different search and rescue operations, which is an additional challenge in detecting a missing or injured person.

Today, almost all search and rescue services have integrated the use of drones in their search. This is due to the increasing availability of drones with quality high-resolution cameras. It takes approximately 25 seconds for a video analyst to detect a victim on a drone recording [20]. The benefit of video analytics is knowing the context of the image and predicting where the person may be based on previous experiences, but the analyst focuses only on a small portion of the image so the assistance of an automated detector can be of great use.

The rest of the paper is organized as follows: in Section II. we will present public available drones datasets. An overview of methods using drones in search and rescue missions is given in Section III with an overview of the state-of-the-art object detector algorithms in drone imagery. The paper ends with a conclusion and a proposal for future research.

2. Public available drones datasets

A comparison of publicly available sets of images taken with a drone and prepared for deep learning tasks is given in Table 1.

Table 1. Comparison of publicly available drone datasets

	Campus	UAV123	CARPK	Okutama-action	UAVDT	VisDrone
Year	2016	2016	2017	2017	2018	2018

D			x	x	x	x
S		x			x	x
M	x				x	x
Frames	929,5k	112 578	1 448	77 365	80 000	189 473
Boxes	19,5k	110k	90k	422,1	841,5k	2 500k
Categories	6	6	1	1	1	10
Resolution	1400x1904	1280x720	1280x720	3840x2160	1080x540	3840x2160

A. Campus

The Campus is a large dataset containing images and videos of different classes such as pedestrians, bicyclists, cars, skateboarders, golf carts, buses, taken inside the campus from a bird's eye view (see Figure 1. a).

The footage was taken with a 4K drone-mounted camera (3DR solo) flying at a height of approximately 80 meters. The dataset contains about 19,000 objects at a resolution of 1400 x 1904 px, [9].

B. UAV123

The UAV123 dataset contains a set of scenes ranging from urban landscapes, roads, fields, and beaches with objects such as cars, trucks, boats, and persons. Persons are additionally tagged for object tracking. Activities such as walking, cycling, swimming, car driving are also labeled.

The data are divided into 3 sub-groups [10]:

- 103 video clips taken with the DJI S1000 drone tracking different objects between 5m and 25m in height, 720p, and 4K resolution at 30 and 96 fps.
- 12 videos shot with images of lower quality and resolution
- 8 synthetic video clips recorded using a drone simulator of the Unreal4 Game Engine.

C. CARPK

The CARPK is according to authors [11], the first and largest database of drone recordings that supports object counting, and provides the bounding box annotations. More specifically, there are 89 777 tagged cars on the dataset. The

cars were shot by drones in four different parking lots. The dataset is tailored to deep learning algorithms for object count and localization scenarios.

Figure 1. Examples of the scenes captured in a) Campus, b) CarPK

D. Okutama-action

The Okutama-action database contains 43 fully labeled drone video clips for training and testing models when detecting multiple simultaneous actions within different categories (reading, handling, carrying different items), [11].

The videos were recorded using two DJI Phantom 4 drones in 4K baseball court and at 30 fps at 10m to 45m height, while the camera angle is 45 or 90 degrees. The dataset for each video contains metadata such as camera angle, speed, and height. The shots were taken with two different lighting conditions (sunny and cloudy)

E. UAVDT

The UAVDT consists of 100 drone videos in multiple urban locations such as streets, squares, intersections, etc [11]. The videos were shot at 1080 x 540 px with 30 fps in different weather conditions (day, night, fog), and in three different altitude ranges (low: 10m to 30m, medium: 30m to 70m high: more than 70m) and different camera views (front view, side view, and bird's eye view). An example of the scene captured in the UAVDT dataset is given in Figure 2.a.

F. VisDrone

VisDrone is a set of data shot in different scenes focusing on four basic problems in the field of computer vision (object detection in images, object detection in videos, single object tracking and multiple object tracking).

The dataset consists of 263 video clips and an additional 10.209 images [11]. Videos/images were recorded on different drone platforms (DJI Mavic, DJI Phantom Series 3, 3A, 3SE, 3P, 4, 4A, 4P) in 14 different cities in China. The dataset covers different weather and light conditions of maximum video resolution (3840 x 2160 px) and images (2000 x 1500 px).

Figure 2. Examples of the scenes captured in a) UAVDT, b) VisDrone

Each of the datasets presented here is important for the development of computer vision research in the field of UAV images. However, it is clear from the descriptions of each image database and examples that they are intended for a specific task and tailored to a particular problem. For a specific problem, such as searching and rescuing people, there are missing appropriate scenarios where people in non-standard poses appear (e.g. injured persons during a fall), so they need to be recorded and included in the set.

3. Computer vision tasks in search and rescue operations

Detection of people in images and videos plays a significant role in various applications, but in this section, we focus on search and rescue applications using drone recordings. The search and rescue problem can be divided into four application areas: in combat, on water, in urban and non-urban areas [16]. The use of drones in search and rescue operations has been discussed in [18][19][20] [17]. In this review, we will focus on non-urban areas and water areas.

In [20] image segmentation and contrast enhancement were applied and then convolution neural networks (multiple single shot detector SSD) for the detection of persons ranging from 5 to 50 px, on drone imagery. They also used a 3D game editor to generate synthetic search and rescue datasets.

The Inception model with the Support Vector Machine (SVM) classifier is used in [21] for detecting people trapped in an avalanche using drone imagery. In [23] the focus is on detecting humans on sea recorded using an unmanned aerial vehicle equipped with a multi-spectral camera. A modified MobileNet convolution neural network architecture is used for detection.

In [24], authors have developed a system for detecting people and action recognition on the Okutama-action dataset while calculating GPS locations. For object detection, a model that was upgraded to MobileNetv2 and called POINet was used. Another example of the use of GPS signals in search and rescue actions is given in [25]. The assumption is that the injured person has a mobile device switched on, so a GSM radio signal of the mobile device is used to log the position of the injured person from the strength of the single and GPS position of the drone.

A platform for the detection of persons in water with the Tiny YOLO V3 architecture integrated on the NVIDIA Jetson X1 computer was introduced at [26]. The model was trained on the COCO dataset and swimmer's custom dataset recorded with a drone equipped with a GoPro camera in HD resolution. For the detection of sea surface objects, the use of a drone thermal camera and a real-time onboard algorithm was proposed in [27] to detect and track objects on the ocean surface.

The strategy of using semi-supervised and supervised machine learning approaches for the classification of aerial imagery and object detection along with the suggestion of hardware and software architecture for the UAV platform is given in [28]. An algorithm for planning a search path for a UAV and using unmanned ground vehicles (UVG) to verify the identity of an object detected by the UAV is given in [29]. In [30] the authors classify drone imagery on human and non-human images and provide classification results using several CNN

architectures. According to [31], it was the first paper that applies multiple object visual tracking to aerial imagery for search and rescue purposes, invariant to scale, translation and rotation, and with the ability to re-identify persons. Person detection is based on color and depth data and the use of the Human Shape Validation Filter that uses the locations of human joints obtained from the Convolutional Pose Machine [32]. The purpose of the filter is to study the shape of the human skeleton on detections to avoid false detections.

According to the results of the Vision Meets Drones competition, VisDrone 2019 [33], the Cascade R-CNN [34] model and models derived from it are most commonly used to detect objects such as pedestrians, cars and bicycles in largescale benchmark dataset covering a wide range of aspects including location (taken from 14 different cities), environment (urban and country), objects (pedestrian, vehicles, bicycles, etc.), and density (sparse and crowded scenes).

Cascade R-CNN is a multi-stage object detector framework, which aims to increase the quality of detection by constantly increasing the intersection over union (IoU) thresholds [35]. Cascade R-CNN was used in different applications including agricultural, aerial photography, fast delivery, and surveillance, followed by CenterNet [36] and RetinaNet [37].

CenterNet is a one-stage highly efficient detector for exploring the visual patterns within each bounding box. For detecting an object, this approach uses a triplet, rather than a pair, of keypoints. Paying attention to the center information, RetinaNet has a feature pyramid network (FPN) [37] attached to its backbone to generate multi-scale pyramid features. Then, pyramid features go into classification and regression branches, whose weights can be shared across different levels of the FPN. The focal loss is applied to compensate for the accuracy drop, which improves performance. The most used detectors in the VisDrone competition, as the backbone mainly use ResNet- 101, ResNet 101, ResNet 50, and SEResNeXt50.

The performance that object detectors achieve on images captured with a drone is much lower than that achieved on images that are not a bird's eye view in different application domains [38, 40, 40]. The top three detectors in the VisDrone

2019 competition (DPNet-ensemble, RRNet [41], and ACM-OD) in the image detection category reach an average precision (AP) of about 29% with an IoU> 50%. For person detection a maximum of 16% AP is achieved (BetterFPN, 16.45% AP, DPNet ensemble [35], 15.97% AP, ACM-OD [35], 15.50% AP).

Slightly lower object detection results than in the case of images were achieved in the object detection category on video [44]. The three best results were achieved by the following algorithms: DBAI-Det with 29.22% AP, AFSRNet with 24.27% AP, and HRDet+ with 23.03% AP. As in the case with the images, positive detection was counted if IoU is greater than or equal to 50%. In the case of detection of persons, the best results were achieved with DBAI-Det, VCL-CRCNN, and AFSRNet, with pedestrian detection results being different from the detection of a person in general.

The success of the best algorithms is attributed to the combination of many recently proposed powerful networks, including DCNv2 [45], FPN, and Cascade R-CNN, and detection performance is significantly enhanced by the benefits of anchor-based RetinaNet and anchorless FSAF.

4. Conclusion

The paper provides an overview of the object detection on imagery recorded by unmanned aerial vehicles (drones). The first part of the paper gives an overview of the current state of publicly available datasets with their characteristics and appropriate tasks. The following section shows the research activities using drones in search and rescue operations and computer vision methods for missing person detection. Finally, the models that currently show the best detection results on images made by unmanned systems are listed.

In future work, it is necessary to create a dataset of drone recordings for better detection of injured persons. Such a dataset would contain people in atypical poses that are not contained in existing datasets. Combining knowledge transfer from existing datasets and the new custom set, it is necessary to test the state-of-the-art models and analyze their performance in a new set of scenes characteristic for search and rescue operations. If necessary, adaptation

and enhancement of existing models will be proposed to achieve the best possible detection results for disabled and missing persons in non-urban and off-water areas.

References

[1] S. Sambolek, M. Ivasic-Kos, Detection of Toy Soldiers Taken from a Bird's Perspective Using Convolutional Neural Networks, ICT Innovations 2019, Ohrid. Springer Communications in Computer and Information Science

[2] M. Kristo, M. Ivašić-Kos, Thermal Imaging Dataset for Person Detection; Proceedings of 42nd International ICT Convention – MIPRO 2019, Opatija, Hrvatska: Mipro, 2019. str. 1316-1321

[3] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards realtime object detection with region proposal networks," in Advances in neural information processing systems, 2015, pp. 91-99.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, A. C. Berg, "SSD: Single shot multi-box detector," in European conference on computer vision, Springer, Cham, 2016, pp. 21-37.

[5] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779-788.

[6] T. Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, "Focal loss for dense object detection," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980-2988.

[7] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, "The pascal visual object classes challenge: A retrospective," International journal of computer vision, 2015, 111(1), 98-136.

[8] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In ECCV, 2014.

[9] A. Robicquet, A. Sadeghian, A. Alahi, S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in

European conference on computer vision, Springer, Cham, 2016, pp. 549-565.

[10] M. Mueller, N. Smith, B. Ghanem, “A benchmark and simulator for UAV tracking,” in European conference on computer vision, Springer, Cham, 2016, pp. 445-461.

[11] M. R. Hsieh, Y. L. Lin, W. H. Hsu, “Drone-based object counting by spatially regularized regional proposal network,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4145-4153.

[12] M. Barekatain, M. Martí, H. F. Shih, S. Murray, K. Nakayama, Y. Matsuo, H. Prendinger, “Okutama-action: An aerial view video dataset for concurrent human action detection,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 28-35.

[13] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, Q. Tian, “The unmanned aerial vehicle benchmark: Object detection and tracking,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 370-386.

[14] P. Zhu, L. Wen, X. Bian, H. Ling, Q. Hu, “Vision meets drones: A challenge,” arXiv preprint arXiv:1804.07437, 2018.

[15] R. J. Koester, Lost Person Behavior: A Search and Rescue. DBS Productions LLC, 2008.

[16] S. N. A. M. Ghazali, H. A. Anuar, S. N. A. S. Zakaria, Z. Yusoff, “Determining position of target subjects in maritime search and rescue (MSAR) operations using rotary-wing unmanned aerial vehicles (UAVs),” in 2016 International Conference on Information and Communication Technology (ICICTM), IEEE, 2016, pp. 1-4.

[17] P. Doherty, P. Rudol, “A UAV search and rescue scenario with human body detection and geolocalization,” in Australasian Joint Conference on Artificial Intelligence, Springer, Berlin, Heidelberg, 2007, pp. 1-13.

[18] M. A. Goodrich, B. S. Morse, C. Engh, J. L. Cooper, J. A. Adams, “Towards using unmanned aerial vehicles (UAVs) in wilderness search

and rescue: Lessons from field trials," Interaction Studies, 2009, 10(3), 453-478.

[19] S. Waharte, N. Trigoni, "Supporting search and rescue operations with UAVs," in 2010 International Conference on Emerging Security Technologies, IEEE, 2010, pp. 142-147.

[20] C. A. Baker, S. Ramchurn, W. T. Teacy, N. R. Jennings, "Planning search and rescue missions for UAV teams," in Proceedings of the Twenty-second European Conference on Artificial Intelligence, IOS Press, 2016, pp. 1777-1778.

[21] K. Yun, L. Nguyen, T. Nguyen, D. Kim, S. Eldin, A. Huyen, E. Chow, "Small target detection for search and rescue operations using distributed deep learning and synthetic data generation," in Pattern Recognition and Tracking XXX (Vol. 10995, p. 1099507), International Society for Optics and Photonics, 2019.

[22] M. Bejiga, A. Zeggada, A. Nouffidj, F. Melgani, "A convolutional neural network approach for assisting avalanche search and rescue operations with UAV imagery," Remote Sensing, 2017, 9(2), 100.

[23] A. J. Gallego, A. Pertusa, P. Gil, R. B. Fisher, "Detection of bodies in maritime rescue operations using unmanned aerial vehicles with multispectral cameras," Journal of Field Robotics, 2019, 36(4), 782-796.

[24] R. Geraldes, A. Gonçalves, T. Lai, M. Villerabel, W. Deng, A. Salta, H. Prendinger, "UAV-based situational awareness system using deep learning," IEEE Access, 2019, 7, 122583-122594.

[25] S. O. Murphy, C. Sreenan, K. N. Brown, "Autonomous unmanned aerial vehicle for search and rescue using softwaredefined radio," in 2019 IEEE 89th Vehicular Technology Conference VTC2019-Spring, 2019, pp. 1-6. IEEE.

[26] E. Lygouras, N. Santavas, A. Taitzoglou, K. Tarchanidis, A. Mitropoulos, A. Gasteratos, "Unsupervised human detection with an embedded vision system on a fully autonomous UAV for search and rescue operations," Sensors, 2019, 19(16), 3542.

[27] F. S. Leira, T. A. Johansen, T. I. Fossen, “Automatic detection, classification and tracking of objects in the ocean surface from UAVs using a thermal camera,” in 2015 IEEE aerospace conference, IEEE, 2015, pp. 1-10.

[28] J. Sun, B. Li, Y. Jiang, C. Y. Wen, “A camera-based target detection and positioning UAV system for search and rescue (SAR) purposes,” Sensors, 2016, 16(11), 1778.

[29] Z. Kashino, G. Nejat, B. Benhabib, “Aerial wilderness search and rescue with ground support,” Journal of Intelligent & Robotic Systems, 2019, 1-17.

[30] T. Marasović, V. Papić, “Person classification from aerial imagery using local convolutional neural network features,” International Journal of Remote Sensing, 2019, 1-19.

[31] A. Al-Kaff, M. J. Gómez-Silva, F. M. Moreno, A. de la Escalera, J. M. Armingol, “An appearance-based tracking algorithm for aerial search and rescue purposes,” Sensors, 2019, 19(3), 652.

[32] S. E. Wei, V. Ramakrishna, T. Kanade, Y. Sheikh, “Convolutional pose machines,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4724-4732.

[33] D. R. Pailla, “VisDrone-DET2019: the vision meets drone object detection in image challenge results, 2019.

[34] Z. Cai, N. Vasconcelos, “Cascade R-CNN: Delving into highquality object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6154-6162.

[35] M. Burić, M. Pobar, M. Ivasic-Kos, Adapting YOLO Network for Ball and Player Detection; Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM, Prag, Češka: SciTePress, 2019. str. 845-851

[36] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, “Centernet: Keypoint triplets for object detection,” in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 6569-6578.

[37] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, “Feature pyramid networks for object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117-2125.

[38] M. Ivasic-Kos, M. Krišto, M. Pobar, Person Detection in Thermal Videos Using YOLO; Proceedings of SAI Intelligent Systems Conference IntelliSys 2019: Intelligent Systems and Applications, Cham: Springer, 2019. str. 254-267

[39] M. Pobar, M. Ivasic-Kos, Detection of the leading player in handball scenes using Mask R-CNN and STIPS, Proc. SPIE 11041, Eleventh International Conference on Machine Vision (ICMV 2018), Muenchen: SPIE, 2018

[40] M. Ivasic-Kos, M. Pobar; Building a labeled dataset for recognition of handball actions using Mask R-CNN and STIPS, 7th European Workshop on Visual Information Processing EUVIP, Tampere, Finska: IEEE, 2018. str. 1-6

[41] C. Chen, Y. Zhang, Q. Lv, S. Wei, X. Wang, X. Sun, J. Dong, “RRNet: A hybrid detector for object detection in drone-captured images,” in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2019.

[42] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, Z. Zhang, “MMDetection: Open MMLab detection toolbox and benchmark, 2019, arXiv preprint arXiv:1906.07155.

[43] B. Singh, M. Najibi, L. S. Davis, “SNIPER: Efficient multi-scale training,” in Advances in Neural Information Processing Systems, 2018, pp. 9310-9320.

[44] P. Zhu, D. Du, L. Wen, X. Bian, H. Ling, Q. Hu, T. Peng…, “VisDrone-VID2019: “The vision meets drone object detection in video challenge results,” 2019.

[45] X. Zhu, H. Hu, S. Lin, J. Dai, “Deformable convnets v2: More deformable, better results,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9308-9316.

RAD 3. PERSON DETECTION IN DRONE IMAGERY

Ovaj rad je objavljen kao: Sambolek, Saša, and Marina Ivašić-Kos. "Person Detection in Drone Imagery," 2020 5th International Conference on Smart and Sustainable Technologies (SpliTech), Split, Croatia, 2020, pp. 1-6, doi: 10.23919/SpliTech49282.2020.9243737.

This research was supported by Croatian Science Foundation under the projects IP-2016-06-8345 “Automatic recognition of actions and activities in multimedia content from the sports domain” (RAASS) and IP-2018-01-7619 “A Knowledge-based Approach to Crowd Analysis in Video Surveillance (KACAVIS) and by the University of Rijeka (project number 18-222).

https://ieeexplore.ieee.org/abstract/document/9243737

1. Introduction

In the case of searching for a missing person, it is of great importance to find the person in the shortest possible time as this increases the likelihood of survival.

In the past few years, unmanned aerial vehicles (drones) have been included in the search and rescue operations in addition to existing resources such as search dogs, human resources, helicopters. During a drone flight, the operator must simultaneously operate the drone and search for the missing person, who, due to the distance, is generally small in size, very often in a lying or crouching position, in inaccessible terrain, obscured by vegetation, which further complicates the detection of missing persons. Ground forces can check the terrain well, but they progress very slowly and have a small view field, especially in the case of dense vegetation so the assistance of the aircraft is necessary.

An ideal search and rescue system would be one that would include drones that could autonomously fly and detect objects of interest in real-time, and then alarms ground teams and forwards them the location and the image of detected objects. At lower altitudes, the drone can capture more details about objects of interest, while at higher altitudes it covers a larger area but the objects are extremely small on them.

The drone footage is being analyzed by video analysts today. In [1] is described that the human video analyst was able to detect the victim within 25 seconds in the drone recording (4K image, with target size 5 – 50 pixels), focusing on the small part of the image that, according to previous experience, is the most likely to be the person being sought. High concentration is required for that task and the help of an automated detector can be of great benefit.

In recent years, considerable progress has been made in automatic object detection in images using deep learning (convolutional neural networks). However, it has been shown that popular detectors such as SSDs [2], YOLO [3], and RetinaNet [4] do not achieve equally good detection results from a bird's eye view or on images captured by drones [5].

Automatic detection of objects on drone imagery poses greater challenges than the same task on stationary camera images. One reason is the change in shooting height, which causes a significant change in the size of the object, a change in the shooting angle and the position of the object towards the camera, and a change in perspective. In the case of a search and rescue operation, the visibility of the object is also affected by changes in lighting (daytime, nighttime) and weather conditions (sunny, cloudy, foggy or rainy). With all of the above, the challenge of detecting an object captured by a drone is very often very small object size that is hard to see in a cluttered background with frequent occlusions.

In this paper, the performance of a popular state-of-the-art object detector, a Faster R-CNN for detecting persons in drone-captured images was investigated.

Two publicly available sets of images taken with a drone and prepared for deep learning tasks, the VisDrone and Okutama – Action datasets have been selected. Each of these datasets includes scenes designed for a specific task and tailored to a specific problem. For a specific problem, such as a search and rescue operation, they do not have proper scenarios with people lying in the grass, crouching behind a stone, leaning against a tree or other atypical poses for urban scenes, so our own custom set of images called SARD (Search And Rescue Dataset) was created.

The rest of the paper is organized as follows: in Section II, an overview of the drone-related research is given with an emphasis on image datasets and commonly used detection methods. Section III describes the experiment and training of the Faster RCNN model for person detection on the custom dataset SARD containing typical scenes for the rescue operation and two public datasets of drone imagery. Obtained results and discussion are given in Section IV. The paper ends with a conclusion and a proposal for future research.

2. Related work

The detection of persons in drone images and videos is of increasing relevance and has a significant role to play in the safety of persons and the surveillance in urban and non-urban areas.

A. Datasets

A prerequisite for the use of models in various applications, and so is in the field of UAV imaging, is the preparation of appropriate image databases used for supervised model learning. Publicly available datasets that have contributed to the development of computer vision research in the field of drone images are Campus [6], UAV123 [7], CARPK [8], Okutama – Action [9], UAVDT [10], and VisDrone [11].

Each of the image databases is intended for a specific purpose and is tailored to a specific problem. They usually contain different classes taken from a bird's eye view that are present in urban scenes such as pedestrians and skateboarders on the streets or squares, cyclists, cars, buses, and trucks on roads, crossings, or parking lots [6]. There are also examples containing non-urban landscapes such as fields and beaches with objects such as boats and bathers [11]. In some cases, activities of the people such as walking, running, reading, hugging, and the like are also indicated [9].

In this work, VisDrone, and Okutama – Action datasets have been used, so they were described in more detail.

1) VisDrone

The VisDrone dataset contains 288 videos and 10,209 images captured on different drone platforms (DJI Mavic, DJI Phantom Series 3, 3A, 3SE, 3P, 4, 4A, 4P) in 14 different cities in China. The set covers different weather and light conditions of maximum video resolution (3840 x 2160 px) and images (2000 x 1500 px). Within the set are 10 categories of objects (pedestrian, person, car, van, bus, truck, motor, bicycle, awning-tricycle, and tricycle).

2) Okutama-Action

The dataset contains 43 drone-recorded video clips for training and testing models to detect multiple simultaneous actions within different categories, human to human interaction: handshaking, hugging, human to object interaction: reading, drinking, pushing/pulling, carrying, calling, non-interaction: running, walking, lying, sitting, standing. Using the open-source tool to annotate objects

VATIC [12], they manually annotate every tenth frame, and the tags were linearly interpolated to 30 fps.

The videos were shot using two DJI Phantom 4 drones on a baseball court in 4K resolution with 30 fps, at a height of 10 m to 45 m, with a camera angle of 45 or 90 degrees. The dataset for each video contains metadata such as camera angle, speed, and height. The shots were taken with two different lighting conditions (sunny and cloudy).

Analyzing the available databases of drone images, the conclusion was that there is still no publicly available dataset containing scenes captured by a drone during search and rescue operations, so in this paper, our dataset for this purpose has been created.

B. Methods used to detect persons in rescue operations

In recent years, drones have been increasingly used, and methods for the automatic detection of drone imaging objects have been increasingly developed. We are particularly interested in detection methods used to detect persons in search and rescue operations. One of the earlier works is [13] where drones are used to find injured persons in search and rescue operations using HOG descriptors [14].

The advantages of using deep learning for computer vision tasks using drones are presented in [15], where authors have analyzed three models (SSD, Faster R-CNN, and RetinaNet) and showed that RetinaNet is faster and more accurate model than others analyzed, in object detection task on drones imagery.

In [16] multi-spectral and visible-spectrum cameras are used, with modified MobileNet architecture to detect and localize bodies in the sea. The upgraded version of the MobileNetv2 model and the Okutama-Action dataset is used in [17] for person detection. In [18] for detection of persons in the water, a Tiny YOLO V3 Architecture integrated on NVIDIA Jetson TX1 computer is used. The model was trained on a COCO dataset and a custom swimmer's dataset recorded with an unmanned aerial vehicle.

The use of drones to detect avalanche casualties is described in the [19], where the Inception model with the Support Vector Machine classifier is used for detection.

The YOLO detector was used to detect aircraft in real-time on videos obtained from the UAV during the flight [20] while the aircraft were grounded. The YOLO detector has also proven to be a good solution for people detection from a bird's eye view in quite demanding shooting conditions [21] with a large number of objects on the scene [22], with occlusion among people and indoors [23].

In [24] image segmentation, contrast enhancement, and convolution neural networks are applied for the detection of persons (range 5 to 50 px) on drone imagery. They have also used ARMA3 a 3D game editor to generate synthetic search and rescue datasets and data augmentation (flip, rotation, zoom in/out). An approach that increases a relatively modest set of real-world data with synthesized images has also been applied in [25] to influence the improved performance of object detectors. The size and position of the persons or object in general in the synthesized images should be adjusted to the actual situations, e.g. in these works persons was set on 5-30 px.

For search and rescue operations to be carried out even when there is no more daylight, the use of IR light should also be considered. A Yolo detector was used to detect humans on thermal images recorded at night in [26] and in [27] to recognize humans while sneaking through the woods and animals during bad weather. In [28] an infrared camera was mounted on an unmanned aerial vehicle to detect poachers and control animal movements using Faster R-CNN.

The [29] describes applying multiple object-based visual tracking to aerial imagery for search and rescue purposes. Person detection was based on color and depth information and the use of the Human Shape Validation Filter that uses the locations of the human joints detected by the Convolutional Pose Machine [30] to avoid false detections. During the tracking of persons, the method used must be invariant for the scale, movement, and rotation of the object and also that has the ability to re-identify persons. For that purpose, in [31] a DeepSort method was used to track people on the sports field. When monitoring objects,

especially when the objects are very far from the cameras and often in occlusion, as is the case with drone imagery, satisfactory results are not yet achieved.

Something that certainly goes in favor of solving this challenge is more precise object detection.

3. Experiment setup

The experiment aimed to detect people in scenes appropriate to search and rescue cases.

For detection, Faster R-CNN [32] was decided to use, which has become the de facto standard after proving to be a multi-purpose detector that enables high accuracy of detecting small and large objects [33]. The original implementation of the Faster R-CNN model was used with Feature Pyramid Network - FPN [34] as a backbone without changing the hyperparameters of the model. The model was trained on the COCO [35] dataset. According to results reported in [36], average precision (AP) of the faster_rcnn_R_50_FPN_3x model for person detection on COCO (val2017) dataset was 54.46%.

Figure 1. Some example of drone images from VisDrone dataset [11] (top), Okutama-Action [9] (middle) and SARD dataset (bottom)

Our goal was to apply the knowledge from the pre-trained model and features and weights learned on a COCO-dataset for person detection to the new but related problem of person detection on images captured by drones. The goal was

to use transfer learning to overcome the isolated learning paradigm for only one task and to avoid learning models from scratch.

The key motivation was that learning a deep learning model for a complex task requires a large amount of data that is not easy to collect and can be very time-consuming and arduous to label and prepare data for supervised learning. An additional motivation to use transfer learning was to learn a model that goes beyond specific tasks and tries to use knowledge from pre-trained models to solve new problems and to avoid the bias problems the most models have, that can be successfully used only on the specific domain for which they were specialized.

Three datasets have been used in this experiment: VisDrone, Okutama - Action, and our dataset SARD.

From the VisDrone dataset, 2,000 images containing person class (Fig 1, top row) were selected. Objects that represent a person are labeled either as pedestrians or as persons in the VisDrone dataset. The set was divided into two subsets, a training set containing 1,598 images with 29,797 labeled persons, and a test set containing 402 images with 7,329 person objects. A model trained on images from the VisDrone dataset is called a CV model.

A custom dataset has been built and prepared, referred to as SARD, containing images recorded by the DJI Phantom 4A drone in the area of Moslavacka Gora, Croatia (Fig. 1, bottom row). The footages were taken in a non-urban area along the road, lake, meadow, quarry, forest. The flight altitude of drones during the shooting was 5 m to 50 m, with a camera angle of 45° to 90° and lens FOV 84°. Different people were recorded while performing various actions such as walking, running, sitting, lying down according to scenarios depicting the injured person. The aim was to capture different situations in which the people being searched may find themselves.

The dataset was obtained from 8 videos in 1920px x 1080px resolution, 50fps with a total of 115,767 frames, by selecting 1,981 images and manually tagging the person on them. The set was divided into two subsets, a training set

containing 1,579 images with 5,160 tagged individuals and a test set of 402 images containing 1,317 tagged persons. To prepare ground truth data, the boxes to each person in the images using the LabelImg tool was ticked.

A model trained on the SARD dataset is called CS.

Besides, the data from the VisDrone dataset and the SARD dataset have been merged to train the model that is referred to as CVS.

The modes were trained on a laptop with an i5-7300HQ CPU and GeForce GTX 1050Ti 4GB GPU on Ubuntu 18.04.4 64-bit. Detectron2, the open-source object detection system from Facebook AI Research, was used as the software. The CV model was trained in 36,000 iterations for 5 hours on the VisDrone subset, and the CS model 5.5 hours on the SARD dataset. The CV model was additionally trained for 5.5 hours on the SARD dataset (CVS label).

For additional testing of the generality of CV, CS, and CVS models, images from the Okutama - Action dataset have been used (Fig.1, middle row). The set consists of 290 selected frames with 2,066 persons that were manually labeled. The image resolution was reduced to 1280px x 720px for this experiment.

4. Results and Discussion

The model performance was compared concerning average precision (AP). The detections are considered true positive when the intersection over union (IoU) of the detected bounding box and the ground truth box exceed the threshold of 0.5. The IoU is defined as the ratio of the intersection of the detected bounding box and the ground truth (GT) bounding box and their union (Fig. 2)

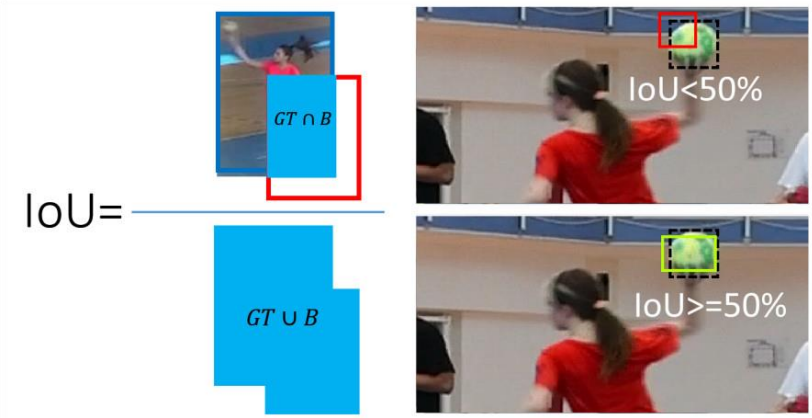


Figure 2. Visual representation of intersection over union (IoU) criteria equal to or greater than 50% [21]

First, we have tested all the models on the SARD dataset. The original model trained on the COCO dataset, with no additional training (referred to as COCO model) achieved AP of 36.84%, much lower than reported on the COCO dataset. The CV model, re-trained on images from the VisDrone dataset achieved 35.88% AP for person detection on SARD data. That is even lower than the original model and represents a negative knowledge transfer probably because images in the VisDrone dataset used for re-training were taken at higher altitudes than in the SARD dataset on which the model was tested.

The CS and CVS models were both re-trained using images from the SARD training dataset, and were more successful, achieving 95,84% AP and 96.40% AP, respectively. The huge difference in detection results is due to the large difference in training sets compared to the test set. In the COCO dataset, there are no images from a bird's eye view and in the VisDrone dataset, the distance of the person from the camera is much greater. On the other hand, images from the SARD training set had an important impact on more accurate adjustment of feature maps and better detection results, since were shot under the same conditions, at the same distance, and from the same perspective as in the case of the SARD test dataset. The graphical representation of the results on the SARD dataset is shown in Fig. 3 (blue columns).

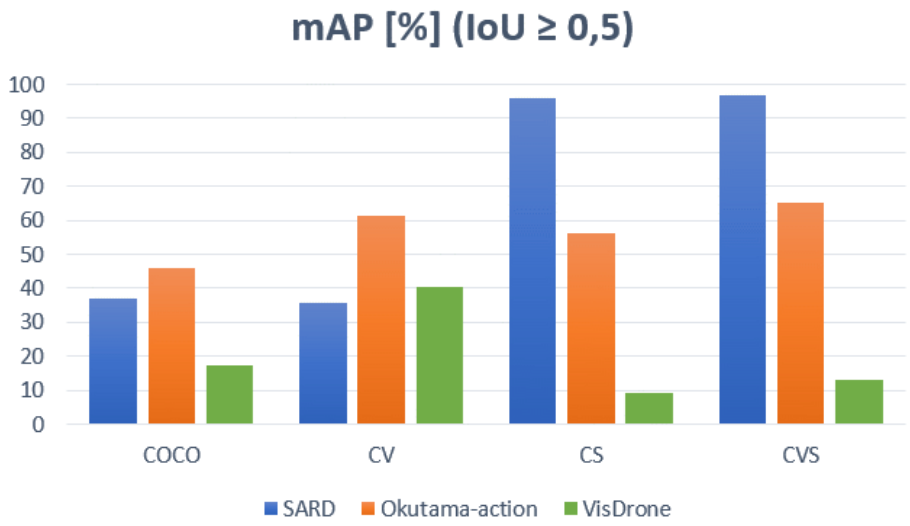


Figure 3. Person detection mAP results of COCO, CV, CS, and CVS models on SARD, Okutama-action and VisDrone test datasets

In the next case, we have tested all the models on the VisDrone dataset. The person detection results on the VisDrone test set are shown in Fig. 3 (green columns) and are as follows: COCO has an AP of 17.37%, CS: 9%, CV: 40.3% and CSV: 12.88%.

All the models not trained on the images of the VisDrone training set (COCO, CS, CSV), achieve significantly worse results than in the first case. The probable reason is that the images in the VisDrone dataset were taken from a much higher shooting height and the objects are tiny, so models that did not have such examples in the learning set cannot detect them.

Finally, all the models were tested on selected images from the Okutama - Action database. This dataset is not used for the training of any of the models. The results are shown in Fig. 3 (orange columns) and are as follows: COCO: 45.97%, CV: 61.31%, CS: 56.12%, and CVS: 65.33%. The best results were achieved by models that had images from the VisDrone database in the training set. The CV model achieved more than 15% better accuracy, and CVS almost 20% better accuracy than the base COCO model. This shows that the initial model is much better trained for detecting persons in drone images after transfer learning on drone datasets.

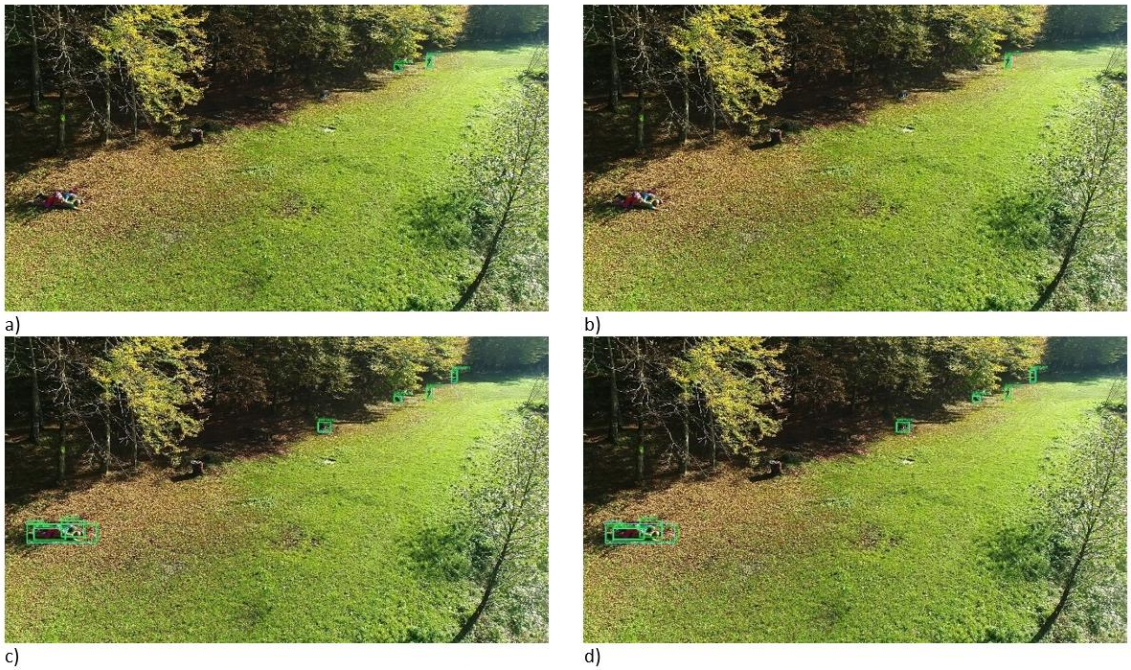


Figure 4. Detection results of: a) COCO, b) CV, c) CS, d) CVS models

The performances of the COCO, CV, CS, and CVS models on different test datasets in terms of the average precision and recall are shown in Fig. 5.

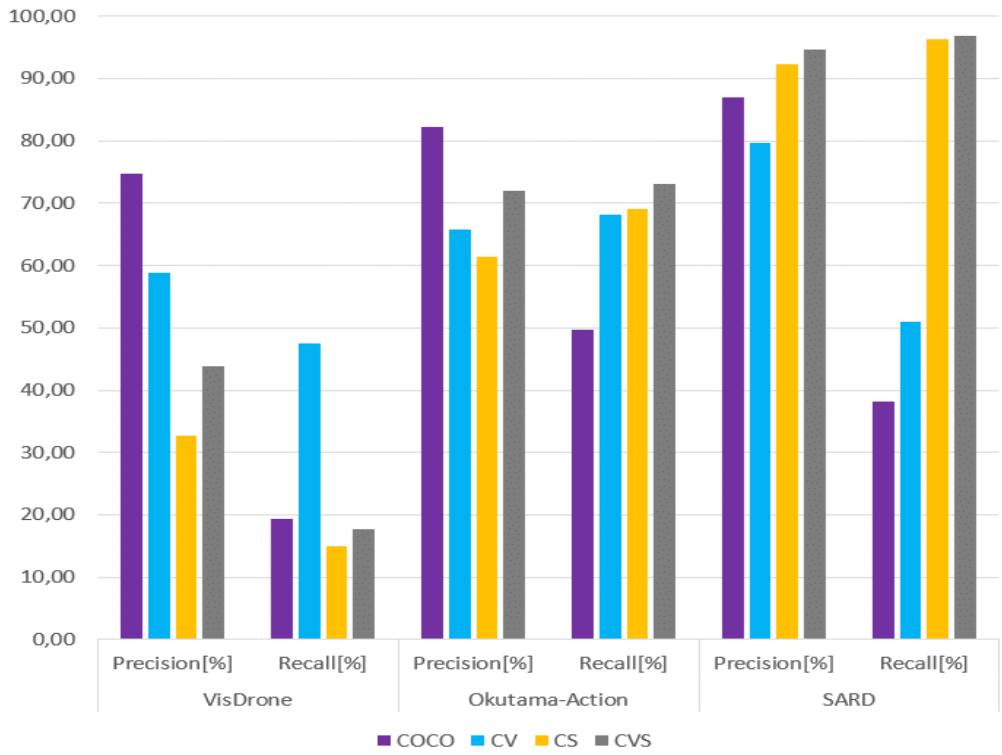


Figure 5. Average precision and recall of COCO, CV, CS and CVS models on different test datasets

Overall, the highest precision and recall of over 90% is achieved by the CS and CVS models on the SARD dataset. That provides a promising base ground for further research when we can investigate the results of precision and recall in the case when the IoU decreases because the goal is to find the lost person and not to detect it completely. On Okutama – Action dataset the best result of 82.15% of precision has the COCO model that was also the best with respect of the highest precision on VisDrone dataset (74.65%) but with a rather low recall of 18%. CV models on the VisDrone dataset get a precision of 58.76%, but with the highest recall of 48%. CS and CSV performed much better on the Okutama - Action dataset in terms of both precision and recall.

A Fig. 5. shows an example of detection results for all four models. There are seven people in the scene, one standing, one running, and five lying down (three on each other - an occlusion example). COCO model has detected running kid and one person lying down, CV model only running kid while CS and the CVS models have detected all persons on the image.

Figure 6. Detection results from bird's perspective of; a) COCO, b) CV, c) CS, d) CVS models

In a case with a camera positioned from a bird's perspective Fig. 6., COCO and CV models have detected the same two pedestrians, while CS and CVS models have detected all persons on the image.

5. Conclusion

Recordings taken from the drones today are used mainly in search of missing persons, in mountain rescue, in the border control, and the like. The ability to automatically detect persons and objects on the images taken from a bird's perspective would greatly facilitate the search and rescue of the people.

In this paper, we have tested the performance of the Faster R-CNN detector for a person detection task on three datasets: SARD, custom dataset built to simulate search and rescue operations, and freely available drone datasets Okutamaaction and VisDrone. In experiment we have used publicly available Faster R-CNN model implementation with corresponding weights learned on the COCO data set.

We have additionally trained the Faster R-CNN model on VisDorone and SARD datasets to fine-tune the model parameters for person detection on drone-captured images. In the experiment, we showed a positive impact of transfer learning so that the model that was re-trained on SARD images and VisDrone images achieved the best results of person detection in drone-captured images concerning both mAP precision and recall metrics.

In future work, we will expand our database with additional drone imagery and focus on changes in detector architecture to achieve even better results in object detection.

References

[1] K. Yun, L. Nguyen, T. Nguyen, D. Kim, S. Eldin, A. Huyen, E. Chow, "Small target detection for search and rescue operations using distributed deep learning and synthetic data generation," in Pattern Recognition and Tracking XXX (Vol. 10995, p. 1099507), 2019.

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, A. C. Berg, “SSD: Single shot multi-box detector,” in European conference on computer vision, Springer, Cham, 2016, pp. 21-37.

[3] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779-788.

[4] T. Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, “Focal loss for dense object detection,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980-2988.

[5] D. R. Pailla, “VisDrone-DET2019: the vision meets drone object detection in image challenge results, 2019.

[6] A. Robicquet, A. Sadeghian, A. Alahi, S. Savarese, “Learning social etiquette: Human trajectory understanding in crowded scenes,” in European conference on computer vision, Springer, Cham, 2016, pp. 549-565.

[7] M. Mueller, N. Smith, B. Ghanem, “A benchmark and simulator for UAV tracking,” in European conference on computer vision, Springer, Cham, 2016, pp. 445-461.

[8] M. R. Hsieh, Y. L. Lin, W. H. Hsu, “Drone-based object counting by spatially regularized regional proposal network,” in Proc. of the IEEE International Conference on Computer Vision, 2017, pp. 4145-4153.

[9] M. Barekatain, et. al. “Okutama-action: An aerial view video dataset for concurrent human action detection,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 28-35.

[10] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, Q. Tian, “The unmanned aerial vehicle benchmark: Object detection and tracking,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 370-386.

[11] P. Zhu, L. Wen, X. Bian, H. Ling, Q. Hu, “Vision meets drones: A challenge,” arXiv preprint arXiv:1804.07437, 2018.

[12] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently scaling up crowdsourced video annotation. International Journal of Computer Vision, 101(1):184–204, 2013.

[13] M. Andriluka, P. Schnitzspan, J. Meyer, S. Kohlbrecher, K. Petersen, O. Von Stryk, B. Schiele, “Vision-based victim detection from unmanned aerial vehicles,” in 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 1740-1747.

[14] N. Dalal, B. Triggs, “Histograms of oriented gradients for human detection,” in International Conference on computer vision & Pattern Recognition, 2005, pp. 886-893.

[15] X. Wang, P. Cheng, X. Liu, B. Uzochukwu, “Fast and Accurate, Convolutional Neural Network Based Approach for Object Detection from UAV,” in IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society, 2018, pp. 3171-3175.

[16] A. J. Gallego, A. Pertusa, P. Gil, R. B. Fisher, “Detection of bodies in maritime rescue operations using unmanned aerial vehicles with multispectral cameras,” Journal of Field Robotics.

[17] R. Geraldes, A. Gonçalves, T. Lai, M. Villerabel, W. Deng, A. Salta, H. Prendinger, “UAV-based situational awareness system using deep learning,” IEEE Access, 2019, 7, 122583-122594.

[18] E. Lygouras, N. Santavas, A. Taitzoglou, K. Tarchanidis, A. Mitropoulos, A. Gasteratos, “Unsupervised human detection with an embedded vision system on a fully autonomous UAV for search and rescue operations,” Sensors, 2019, 19(16), 3542.

[19] M. Bejiga, A. Zeggada, A. Nouffidj, F. Melgani, “A convolutional neural network approach for assisting avalanche search and rescue operations with UAV imagery,” Remote Sensing, 2017, 9(2), 100

[20] M. Radovic, O. Adarkwa, Q. Wang, “Object recognition in aerial images using convolutional neural networks,” Journal of Imaging, 2017.

[21] M. Pobar, M. Ivasic-Kos, “Active Player Detection in Handball Scenes Based on Activity Measures,” Sensors 20 (5), 1475

[22] M. Buric, M. Pobar, M. Ivasic-Kos, Object Detection in Sports Videos. In Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 21–25 May 2018.

[23] M. Buric, M. Pobar, M. Ivasic-Kos, “Adapting YOLO network for a ball and player detection,” Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2019), 2019/2, pp. 845-851

[24] K. Yun, L. Nguyen, T. Nguyen, D. Kim, S. Eldin, A. Huyen, E. Chow, “Small target detection for search and rescue operations using distributed deep learning and synthetic data generation,” in Pattern Recognition and Tracking XXX (Vol. 10995, p. 1099507), International Society for Optics and Photonics, 2019.

[25] M. Buric, G. Paulin, M. Ivasic-Kos, “Object Detection Using Synthesized Data,” ICT Innovations 2019 Web proceedings, (14) pp. 110-124.

[26] M. Ivasic-Kos, Mate Kristo and Miran Pobar, “Human Detection in Thermal Imaging Using YOLO,” in Proceedings of the 5th ACM International Conference on Computer and Technology Applications, ICCTA 2019, NY, USA, pp.20-24.

[27] M. Kristo, M. Ivasic-Kos, M. Pobar, “Thermal Object Detection in Difficult Weather Conditions Using YOLO,” IEEE Access 8, 2020, 125459-125476

[28] E. Bondi, F. Fang, M. Hamilton, D. Kar, D. Dmello, J. Choi, R. Nevatia “Spot poachers in action: Augmenting conservation drones with automatic detection in near real-time,” in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[29] A. Al-Kaff, M. J. Gómez-Silva, F. M. Moreno, A. de la Escalera, J. M. Armingol, “An appearance-based tracking algorithm for aerial search and rescue purposes,” Sensors, 2019, 19(3), 652.

[30] S. E. Wei, V. Ramakrishna, T. Kanade, Y. Sheikh, “Convolutional pose machines,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4724-4732.

[31] M. Buric, M. Ivasic-Kos and M. Pobar, "Player Tracking in Sports Videos," 2019 IEEE International Conference on Cloud Computing Technology and Science, Sydney, Australia, 2019, pp. 334-340.

[32] S. Ren, K. He, R. Girshick, J. Sun, “Faster r-CNN: Towards real-time object detection with region proposal networks,” in Advances in neural information processing systems, 2015, 91-99.

[33] S. Sambolek, M. Ivašić-Kos, “Detection of toy soldiers taken from a bird’s perspective using convolutional neural networks,” in International Conference on ICT Innovations, Springer, Cham, 2019, pp. 13-26.

[34] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, “Feature pyramid networks for object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117-2125.

[35] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, “Microsoft COCO: Common objects in context,” in Proceedings of the European Conference on Computer Vision - ECCV, 2014.

[36] https://github.com/facebookresearch/detectron2/blob/master/MODEL_ZOO.md Accessed: 18. Mar

RAD 4. AUTOMATIC PERSON DETECTION IN SEARCH AND RESCUE OPERATIONS USING DEEP CNN DETECTORS

Ovaj rad je objavljen kao: Sambolek, Saša, and Marina Ivašić-Kos. "Automatic Person Detection in Search and Rescue Operations Using Deep CNN Detectors," in IEEE Access, vol. 9, pp. 37905-37922, 2021,

doi: 10.1109/ACCESS.2021.3063681.

This work was supported in part by the Croatian Science Foundation through projects “Automatic recognition of actions and activities in multimedia content from the sports domain" (RAASS) under Project IP-2016-06-8345, in part by the “A Knowledge-based Approach to Crowd Analysis in Video Surveillance" (KACAVIS) under Project IP-2018-01-7619, and in part by the University of Rijeka under Project 18-222.

https://ieeexplore.ieee.org/abstract/document/9369386

1. Introduction

Many people are included in sport tourism to actively spend leisure time such as skiing, hiking, or nautical, which motivate them to stay in nature. Adrenaline or adventure tourism such as hiking, free climbing, mountain biking, paragliding, and rafting is gaining popularity, therefore the need to protect human life in hard-to-reach areas such as mountains, forests, canyons, caves, bodies of water and, karst phenomena is growing.

Due to a growing number of people living and carrying out various activities in the mountains and other inaccessible places, and because of the very nature of these activities and the physical and mental lack of preparedness for such activities, there is an increasing number of injuries, fractures and various accidents such as slipping, burying, etc. Risks that increase the insecurity of hikers, climbers, and other adrenaline athletes are, in addition to the occurrence of injury or illness, their skills and experience in coping with possible emergencies. Emergencies can arise, for example, due to incorrect assessment of the distance of the destination, incorrect assessment of the difficulty of the road, due to changes in weather conditions, inadequate clothing or equipment, non-compliance with information and warnings, or insufficient preparation and overestimation of one's capabilities or knowledge. Reports of missing persons due to disorientation, illness, or suicidal intentions are also common.

To aid and health care to the injured in these circumstances, it is necessary to organize a search and rescue operation. The search action refers to a situation when the position and condition of the missing person are unknown, so the goal of the action is to locate the position of the missing person in nature. The rescue operation refers to a situation in which it is known that it is necessary to intervene and organize a person's rescue. If the accident's location is unknown in advance, this action includes search elements, too [1], [2].

The organization of assistance and health care in inaccessible areas is very complex, whatever the reason for the intervention. It is necessary to conduct demanding searches of large and complex terrains, especially when searching for a missing person. Besides, time is also an important factor in the search. As

time goes on, the probability of a missing person's survival decreases and the searching area grows exponentially [3].

Search and rescue operations (SAR) require great human potential and material resources because they usually involve a large number of members of the mountain rescue service, search dogs, police and air forces, and more recently, crewless aerial vehicles (drones). Drones are now used for various purposes [4]-[10] and have become a standard in all SAR services globally. Except for searches in urban and non-urban areas, drones are used for searches on water (sea, rivers, floods [11]) or from avalanches. Their compactness, mobility, relatively low cost, and high-resolution real-time video recording are important when making quick decisions during actions and performing tasks that are potentially dangerous to humans, e.g., cliff search. The use of drones has increased the probability of finding a person, and due to "scanning" a larger area in one flight, the search time is shortened.

During search and rescue operations, the operator must analyze real-time images on a small screen while operating the aircraft. As the searching person is relatively small compared to the environment, they often take up only a few pixels on the screen. It is challenging to maintain long-term concentration and attention, even for people trained for it, to search for people in a large mountainous area or an area covered with vegetation. Persons searched for are often sheltered by vegetation, hidden behind a stone, or fused to the ground, further complicating the search even during favorable weather conditions. During rain, fog, and snow, the challenge of searching for a person is even more significant. Also, the searching person is very often in unusual places, most often due to loss of orientation, fall or dementia, in atypical postures and body positions due to injury, such as lying with unnaturally placed limbs or kneeling and sitting on the ground due to exhaustion or sudden disease or covered with stones due to slipping or landslides and the like and are very difficult to spot even in these selected parts of the image (Figure 1).

In SAR operations, operators could be greatly assisted by automatic person detection methods that would mark the persons in the images in real-time, i.e., their position and movement direction.

In recent years, deep convolutional neural networks such as Faster-RCNN [12], Cascade R-CNN [13], RetinaNet [14], SSD [15], YOLOv3 [16] have become successful in detecting people in images of mainly urban scenes and achieve even greater accuracy than humans. To achieve such good performances, deep network models had to be trained on large data sets such as MS COCO [17], Pascal VOC [18], ImageNet [19]. Then, to achieve good detection results or significant improvements in specific domains such as thermal images of the monitored area, some sports scenes, etc., not included in large data sets, it is necessary to additionally train deep networks on the image set from the selected domain [20]-[23].

In SAR operations, the key object is the person, however, recorded from a bird's eye view, and such recordings are not contained in the large data sets on which these state-of-the-art detectors are trained. To achieve the highest possible accuracy of the detection model, the data set on which the model is trained must have similar conditions to those that appear when testing the model, so it is necessary to train the model with a bird's eye view data. Recently, datasets that include images taken by a drone such as Visdrone [24], Okutama-action [25], UAVDT [26] have emerged. Those images are collected for various purposes [24] - [30], such as detecting objects in images and videos, tracking one or more persons, detecting an action, predicting a person's movement, or recognizing events in images. On the other hand, each dataset is tailored to a specific purpose and generally does not include scenes and rescue operations cases. The most similar scenarios shot by a drone to those in search and rescue are those involving people in a park while walking or running, standing in a square, walking down a street, or lying on a beach. Nevertheless, in these cases, persons' poses differ significantly from those who are injured, exhausted, or lost. For this reason, our dataset called SARD was created.

In this work, the SARD dataset was used for transfer learning of the selected state of the art person detectors: Faster R-CNN, YOLOv4, RetinaNet, and Cascade R-CNN and for fine-tuning for person detection in search and rescue scenes. We compared the model results on the SARD dataset. TheYOLOv4 model was selected for further research because of achieving the highest accuracy and detection speed. To improve the detection results of the YOLOv4 model, we have analyzed the influence of different network resolutions, detection accuracy, and transfer learning settings on detection performance. The robustness of the YOLOv4 model to weather conditions and motion blur was also tested. Finally, after comprehensive testing and analysis of the results, we propose a model for person detection in search and rescue scenarios that can be of great help in SAR operations.

The main contributions of the paper are:

a) a novel dataset (SARD) of drone imagery in search and rescue operation, with statistics of the occurrence of small, medium and large object, annotated and prepared for supervised machine learning,

b) comparison of the performance of selected CNN detectors (Cascade R-CNN, Faster R-CNN, RetinaNet, Yolov4) for use in SAR operations,

c) analyses of the influence of different network resolutions, detection accuracies and confidence values on YOLOv4 performance,

d) analysis of different transfer learning strategies considering the impact on model results, e) proposal of ROpti metrics for evaluating detector performances for SAR operations taking into account that there are as many positive detections as possible and as few false detections as possible,

f) proposal of YOLOv4 model to be used for person detection in SAR actions taking care to achieve the highest possible accuracy, with a few false detections as possible, with a network configuration that allows a person's online location and a configuration for off-line analysis, robust to different weather conditions.

Figure 1. Some of the unusual places and atypical positions of the people being searched for, cut from images taken by a drone.

The rest of the paper is organized as follows: Section 2 provides an overview of the research related to the commonly used methods for person detection in search and rescue operations assisted by drones and drone datasets. In Section 3, the SARD dataset was described, which was built and prepared for training models for person detection in SAR operations as well as CNN architectures used for person detection. Section 4 describes in detail the experiments and analyzes the obtained results. The paper ends with the conclusion and direction for future research.

2. Related work

Today most object detectors consist of two parts, the backbone of the detector as a CNN network trained to extract features and a head that predicts the class and boundary box of the detected objects. Networks such as VGG [31], ResNet [32], ResNeXt [33] or MobileNet [34, 35, 36] pre-trained on the ImageNet [19] or OpenImages [37] dataset, are most commonly used as backbones. The head of a detector can be divided into two types: one-stage and two-stage detectors. YOLO [38, 39, 16, 40], SSD [15] and RetinaNet [14] are examples of the one-stage detector. The most representative two-stage detectors are R-CNN detectors [41] including Fast R-CNN [42], Faster R-CNN [12] and, R-FCN [43]. Two-stage detectors are usually more accurate in terms of localization and classification accuracy. On the other hand, they are slower in processing than one-stage detectors. Many detectors add extra layers between the backbone and head (neck), like e.g. Feature Pyramid Network (FPN) [44] whose layers are typically used to collect multiple feature maps each with a different resolution, which is useful for recognizing objects at different scales.

A. Deep CNN detectors in search and rescue operations and drone imagery

According to [45], search and rescue operations can be divided into four areas: search in military operations, search on water, in urban and non-urban areas. The use of drones in search and rescue operations has been discussed in [46 – 49]. The domain of our interest is the non-urban area and water.

In [49], image segmentation and contrast enhancement were applied, followed by an SSD detector to detect persons in drone images. They also used a 3D game editor to generate synthetic datasets depicting search and rescue actions.

The Inception model with the Support Vector Machine (SVM) classifier was used in [50] to detect people trapped in an avalanche by searching with drones. In [51] the focus is on detecting people at sea recorded by unmanned aerial vehicles equipped with a multi-spectral camera and a modified MobileNet architecture is used for detection.

The authors in [52] developed a system for detecting people and recognizing actions on the Okutama-action dataset with GPS location calculation. A model upgraded to MobileNetv2 and named POINet was used to detect objects. Another example of the use of GPS signals in search and rescue operations is given in [53]. It is assumed that the injured person has a mobile device switched on, so the position of the injured person is determined by combining the strength of the GSM signal and the GPS position of the drone.

A platform for detecting a person in the water with the Tiny YOLO V3 architecture was presented in [54]. The model is trained on the MS COCO dataset and dataset recorded by a drone equipped with a GoPro camera in HD resolution. A real-time algorithm for detecting and tracking ocean surface objects has been proposed in [55].

A strategy for using semi-supervised and supervised machine learning approaches to classify aerial imagery and object detection, along with a proposed hardware and software architecture for the UAV platform, is given in [56].

An algorithm for planning a search path for unmanned aerial vehicles (UAVs) and using unmanned ground vehicles (UVGs) to verify the identity of the object detected by the UAV is given in [57]. In [58], the authors compare several CNN architectures for the binary classification task to classify drone images as with or without persons. According to [59], it was the first work to apply multiple visual tracking of objects on aerial photographs for search and rescue purposes. Person detection is based on color and depth data and the use of a human shape filter that uses human joint locations derived from the Convolutional Pose Machine [60]. The purpose of the filter is to investigate the shape of the human body on the proposed detections to avoid false detections.

B. Drone image datasets for CNN training

Recently, an increasing number of datasets have been made using a drone as well as prepared for the training of deep neural networks. These datasets include footage containing scenes of urban areas such as squares, streets, playgrounds, parking lots, etc. (Figure 2).

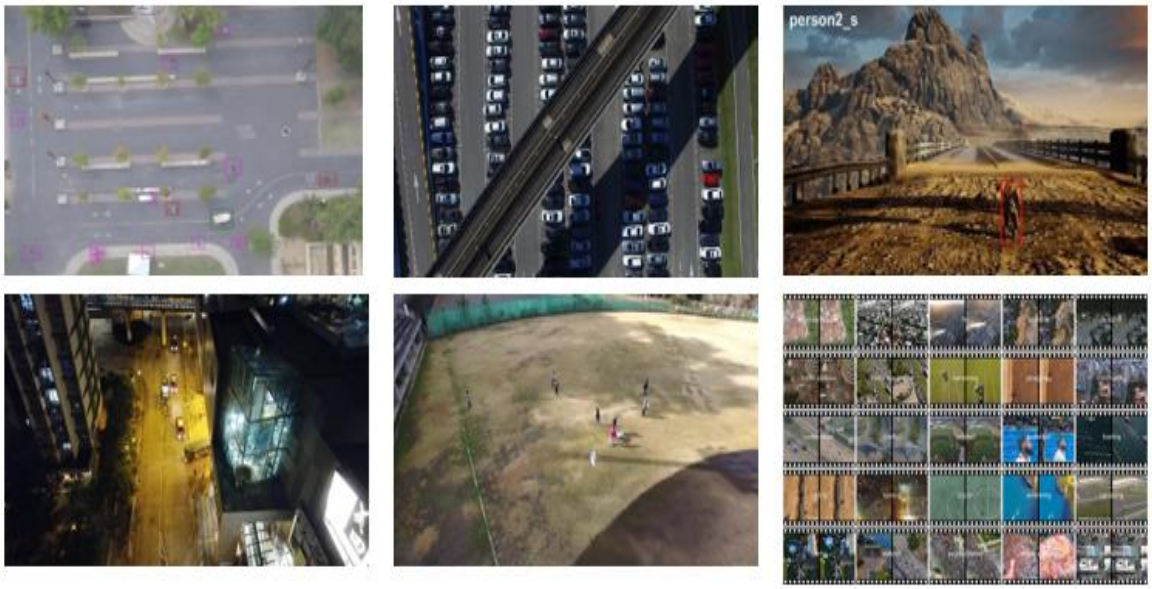


Figure 2. Examples of images from existing datasets. Top-left Campus [28], top-middle CARPK [17], top-right UAV123 [29], bottom-left VisDrone [24], bottom-middle Okutama-action [25], bottom-right ERA [30].

VisDrone [24] data set contains 263 video clips and additional 10,209 images related to detection tasks and tracking of one or more objects. Videos/images were taken on different drone platforms (DJI Mavic, DJI Phantom Series 3, 3A, 3SE, 3P, 4, 4A, 4P) in 14 different cities in China. The set covers different weather and light conditions of maximum video (3840 x 2160 px) and image (2000 x 1500 px) resolution.

Okutama-action [25] contains videos that tag people and the actions of those people such as walking, running, sitting, or lying down. Also, interaction with other objects is annotated such as reading, drinking, carrying, pushing, and interactions between people such as hugging and handling. For counting the objects, there is a CARPK dataset [27] which contains 89,777 marked cars recorded by a drone. Campus [28] is the largest set of data recorded from a bird's eye view, which includes pedestrians, cyclists, cars, buses, etc.

The UAV123 [29] dataset, in addition to drone images, also contains synthetic video recordings made by a drone simulator on the Unreal 4 Game Engine. UAVDT [26] contains drone-recorded videos of an urban area such as streets, squares, intersections, taken in different weather conditions (day, night, fog).

ERA [30] dataset has 24 event classes that can occur on aerial video footage such as fire, flood, traffic jam, concert, etc.

None of the above datasets contain recordings specific to search and rescue operations, so although there are object detectors who achieve excellent results in detecting people on urban scenes, the question is how successful they would be in SAR operations in rural/ mountainous areas? How to test the performance of detectors in the SAR domain if there is no appropriate test set? What performance can be achieved after training the model on examples of SAR scenes and with which model and learning parameters?

3. Experiment workflow

A. Problem formulation

The experiment automatically detects persons using object detectors in images taken by a drone in non-urban areas during search and rescue operations.

Guided by the experience from previous work [61], [62], we have analyzed state-of-the-art object detectors such as Faster-RCNN [12], YOLOv4 [40], RetinaNet [14], and Cascade R-CNN [13]. The aim was to select the one that achieves the best results in terms of accuracy and inference speed and best fits our task.

All considered detectors were pretrained on the MS COCO dataset, and the feature maps learned on that dataset are expected to be useful for detecting persons for our task, too. However, to improve the detection results in SAR applications, the models should be re-trained on an appropriate dataset that contains scenes typical for search and rescue operations.

We searched the available databases of drone images and found out that appropriate publicly available datasets for this purpose did not exist. The existing [24]-[30] do not fully coincide with the intended goal of detecting (injured/exhausted) persons in the non-urban area. However, we decided to use the VisDrone dataset for transfer learning since it contains images of people in the urban scenes that are the closest scenario to our task. Also, we decided to build a dataset of images with scenes that simulate the poses of

injured/exhausted people in the non-urban area taken by drones. Also, to simulate different weather conditions and increase the generality of the model, we will use the available algorithms and generate new images to increase the data set.

We re-train the models on the built dataset, and the model that achieves the best results was selected for further testing and adjustments to improve the detection result further.

B. Dataset creation

SARD database was built to detect casualties and persons in search and rescue scenarios in drone images and videos. The actors in the footage have simulate exhausted and injured persons and "classic" types of movement of people in nature, such as running, walking, standing, sitting, or lying down. Since diverse terrain and backgrounds determine possible events and scenarios in captured images and videos, the shots include persons on macadam roads, quarries, low and high grass, forest shade, and similar.

1) Collection and preprocessing of SARD dataset

During the daylight, the shooting was carried out in the fall, with a high-performance camera of the DJI Phantom 4A drone with a 3-axis solo gimbal stand. Videos were recorded at an FHD resolution of 1920 x 1080 pixels at a frequency of 50 frames per second. The drone flew at different altitudes from 5 m to 50 m and different camera angles (ranging from 45° to 90°). All videos were shot in the area of Moslavacka gora, in Croatia, outside the urban area. Positions of persons in the images range from standard (standing position, sitting, lying, walking, running) to positions typical of exhausted or injured persons reconstructed by actors at their discretion, Figure 3. The actors were nine people of different ages and genders, aged 7 to 55 years, to include differences in movement and postures associated with age and different body constitutions. Also, actors are in various locations, from clearly visible (to the eye) to locations in the woods, tall grass, shade, and similar, which further complicates detection.

Figure 3. Actors, different ages and genders who participated in the recording of the SARD dataset.

From the recordings with a total length of about 35 minutes, 1,981 single frames with people on them were singled out. In the selected images, the persons were manually tagged by a horizontal bounding box typically used for object annotation in remote sensing images and natural scene images [63] so that annotated images could be used to train a supervised model. Tags are stored as XML files in PASCAL VOC format and the YOLO format.

2) Generation of Corr dataset

An extension of the SARD set called Corr was created to increase the robustness of the SARD data. Corr dataset includes images that further simulate various weather conditions that may occur in actual search and rescue situations such as fog, snow, and ice. Also, blur images are included in the Corr set to simulate camera movement and aerial shooting in motion.

The Corr train set was generated from images of the SARD train set, and likewise, the Corr test set was generated from the images of the SARD test set using the

same methods [64]. To achieve an even distribution of data with different weather conditions in the set, we generated the images sequentially by adding the effect of snow, fog, frost, and blur in turn. Each of the effects was added at four levels of concentration to simulate the range of possible weather conditions and motion effects that may occur in actual SAR missions, e.g., light snow and heavy snow, snowstorms, rain, and showers, and the like. For the maximum level of concentration of an effect, we chose the level at which objects, which are relatively small in most images, could still be visually recognized. To test the detection results for specific weather conditions, we created four subsets for testing Corr-snow, Corr-fog, Corr-frost, Corr-fogging, each containing 714 images. The image tags remained the same as in the SARD dataset, so no additional tagging was required. An example of generated images of the Corr dataset is given in Figure 7.

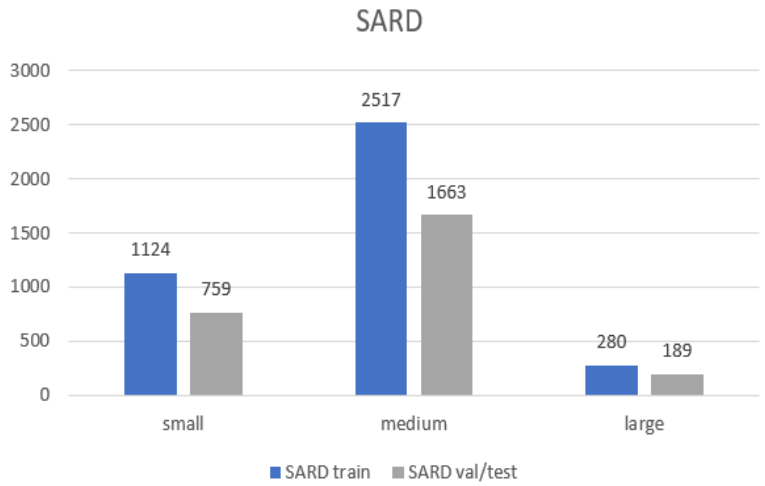


Figure 4. Marked persons according to the size of the bounding box area for the SARD dataset.

3) Statistics of datasets used for transfer learning

The SARD set images were divided 60:40 into a train set and a test set so that they were evenly distributed according to the scenes (background, lighting, person pose, camera angle). The training set contains 1189 images, on which 3921 persons are marked, while the test set contains 792 images, on which 2611 persons are marked. The bounding boxes' dimensions in the SARD set range from 7px for the smallest width and 8px for the smallest height, while the maximum width is 353px and the maximum height is 337px. The area of the

smallest object bounding box is 7 x 12px while the largest is 322 x 231px, and the average bounding box size is 47px x 58 px. The SARD set contains 1883 small person objects (objects whose boundary box area is less than 32^2), 4180 medium person objects (32^2 < boundary box area < 96^2), and 469 large objects (boundary box area > 96^2). The frequency of occurrence of persons in the SARD dataset concerning the size of the object bounding box is graphically shown in Figure 4.

The Corr train set's size corresponds to the SARD train set in terms of the number of images and the number and size of objects. There are 1,903 images in the set, which show 6,265 persons, of which 1,775 are small objects, 4,026 are medium-sized objects, and 464 are large objects. The Corr test set is slightly smaller than the SARD test set because the images on which the persons were not visible after adding blur, rain, snow were deleted. These are mostly images in which people were in the shadows, took up very few pixels, or were occluded. The number of persons in the Corr dataset is shown in Figure 5.

Another 2,129 images from the VisDrone image set, which includes a person or pedestrian tag, were selected for model training to generalize the learning data set. For selected images, person or pedestrian tags are merged into one class: person. This set is referred to as VisDrone2000. The VisDrone2000 drone image dataset was divided into a training set consisting of 1,598 images with 29,797 tagged persons and a test set containing 531 images with 13 969 persons.

The set contains 36,951 small person objects, 6,719 medium-sized objects, and only 96 large person objects. A VisDrone2000 data statistic shows that the VisDrone dataset recordings were made at higher altitudes than the SARD dataset (Figure 6.).

Combinations of used sets and learning methods are described in Section 4.

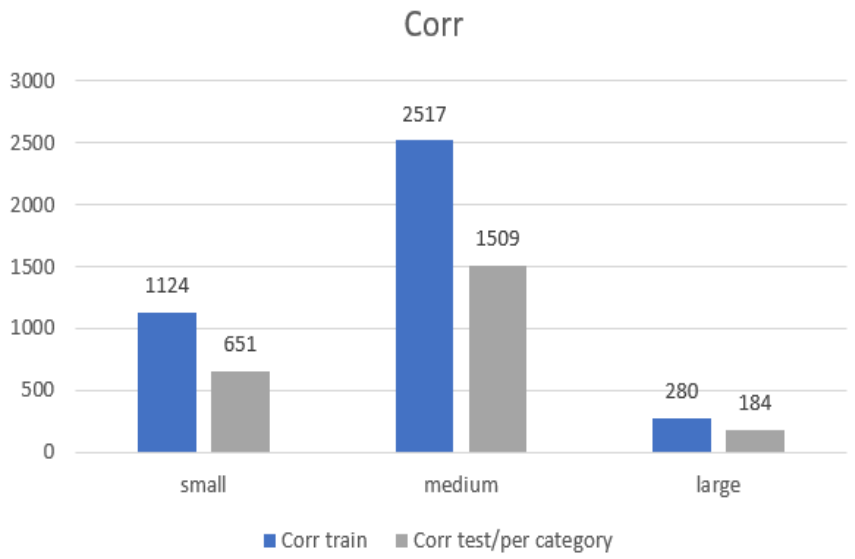


Figure 5. Marked persons according to the size of the bounding box area for the Corr dataset.

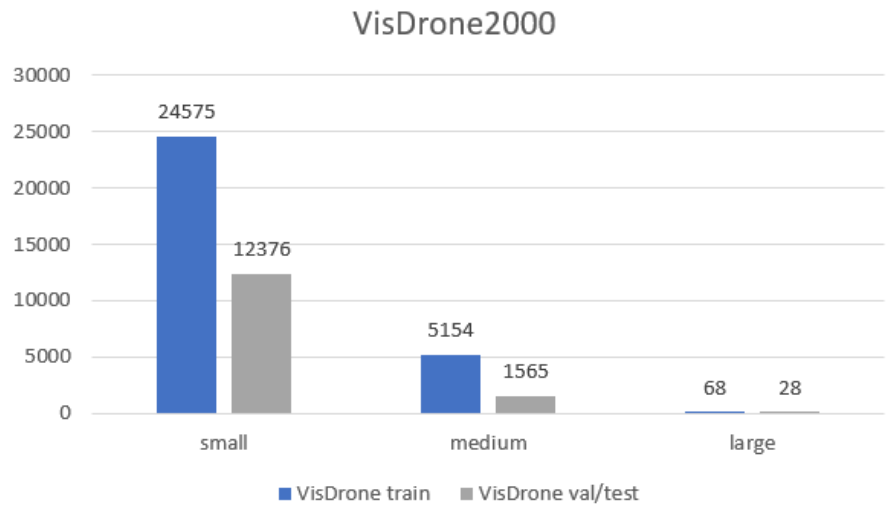


Figure 6. Marked persons according to the size of the bounding box area for the VisDrone2000 dataset.

C. Selected object detectors

We have tested the state-of-the-art object detectors on a custom-made SARD dataset and selected drone images from the VisDrone benchmark dataset to select the best-suited detector for our task of detecting persons in search and rescue scenes.

In the experiment, we have compared the performance of the CNN-based detectors: Faster R-CNN, YOLOv4, RetinaNet, and Cascade R-CNN. All selected detectors were previously trained on the MSCOCO [17] dataset. All detector

models are further trained on bird's eye view images from a part of the VisDrone and a SARD custom dataset to improve their performances.

Below is a brief description of the architecture of examined detectors.

1) Faster R-CNN

The Faster R-CNN detector from the R-CNN series [12], [41], [42], detectors is a two-phase region-based detector. These detectors' basic idea is to select the regions of interest from the image in the first phase. In the second phase, the classification and correction of the coordinates of the object will be performed.

In our case, ResNet50 [32], a pre-trained deep neural network, is used as a backbone, which receives an image at the input and provides feature maps at the output that predicts regions of interest using the Region Proposal Network (RPN). RPN for feature maps of any dimensions, as an output gives a list of RoI's with a certain probability that the object is in the default RoI. The tested Faster R-CNN detector uses FPN to collect multiple feature maps of different resolutions. In this experiment, the implementation of a faster_rcnn_r50_fpn_1x detector from a MMDetection codebase [65] was used.

2) YOLOV4

The YOLO architecture seeks to merge localization and classification problems into one deep convolutional neural network. It divides the image into a grid of dimensions S x S in which each cell provides frames for the object. The probability, which is calculated for each frame, tells us how sure the model is when there is an object inside the frame and how sure it is of the boundaries' accuracy.

For the latest version of the YOLO detector, the authors explored typical algorithms used in deep learning models and further designed and improved some modules.

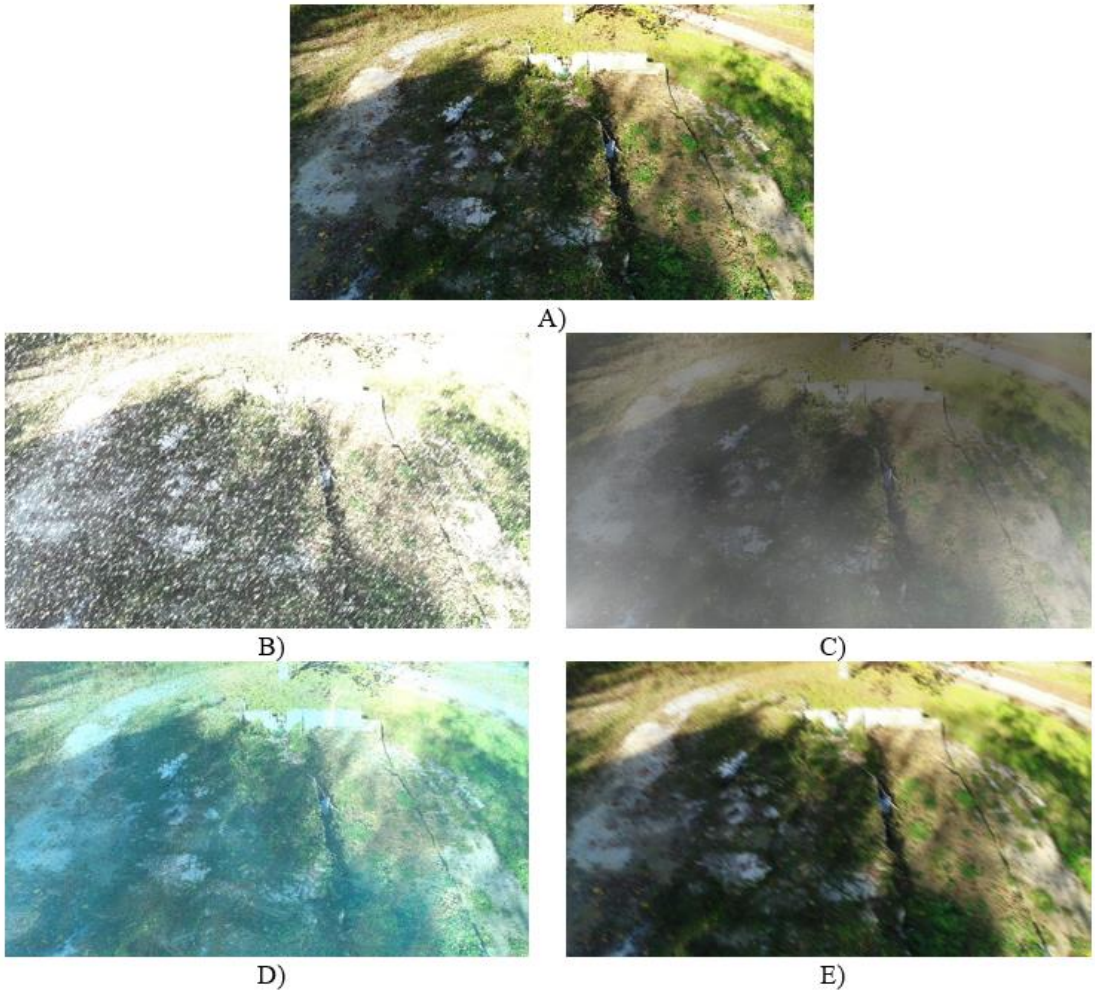


Figure 7. SARD Corr set with the added effect of bad weather and camera shift on the image, examples of generated images: A) original image, B) snow, C) fog, D) ice, E) motion blur.

This model uses CSPDarkNet53 as the backbone [66]. DarkNet53 is a deep residual network with 53 layers, while in the case of YOLOv4, CSPNet (Cross Stage Partial Network) is added to the basic DarkNet53. The authors added Spatial Pyramid Pooling (SPP) [67] as a neck to increase the receiving (receptive) field without causing a decrease in velocity. Instead of the Feature Pyramid Network (FPN) used in the YOLOv3 version, the authors chose the Path Aggregation Network (PAN) [68] while using the original YOLOv3 [16] network for the head.

In addition to the new architecture, the authors also use training optimization to achieve greater accuracy without additional hardware costs, which the authors call "Bag of Freebies." Bag of Freebies includes CutMix, Mosaic, CIoUloss,

DropBlock regularization, etc. On the other hand, the authors propose a "Bag of Specials," a set of modules such as Mish activation, SAM-block, Cross-stage partial connections (CSP), etc., that only slightly increase the hardware cost with a significant increase in detection accuracy.

We used the Darknet framework to train and evaluate the YOLOv4 model.

3) RetinaNet

RetinaNet is a single-phase detector composed of a backbone and two sub-networks specific to the task. The ResNet-FPN network, as the RetinaNet detector's backbone, is responsible for calculating the input image's feature map. The first sub-network performs the classification while the second regresses the boundary frames.

A sub-classification network predicts the probability of an object's presence in each spatial position for each class. This subnetwork is a small FCN associated with each FPN level; this sub-grid parameter is shared at all pyramids levels. Unlike RPN [12], the RetinaNet sub-network for object classification is deeper, uses only 3 x 3 convolutions, and does not share parameters with the frame regression network.

In parallel with the sub-network of object classification, they attach another small FCN to each level of the pyramid for regression of the boundary frame. In experiments, the implementation of the retinanet_r50_fpn_1x detector in the MMDetection codebase [69] was used.

4) Cascade R-CNN

Cascade R-CNN is a multi-phase extension of the Faster R-CNN architecture that aims to increase detection quality by constantly increasing IoU values. The focus is on the detection subnet, adopting an RPN to detect suggestions. However, Cascade R-CNN is not limited to this proposed mechanism since other choices are possible. The goal is to simultaneously increase the quality of hypotheses and improve detection results by combining cascade boundary frame regression and cascade detection. The implementation of a cascade_rcnn_r50_fpn_1x detector in a MMDetection codebase [70] was used.

D. Evaluation metrics

Detector performance (bounding box of detected objects, the class assignment, and a reliability value) was assessed on unseen images using standard evaluation measures such as precision, recall, and mean average precision (mAP). In our case, only the class person is considered, so the mAP is equal to the average precision (AP).

In the case of SAR operations finding a person as soon as possible is key to a successful SAR operation, so it is essential to detect missing people if they exist on the scene. Equally important is to have a few false detections as possible so that human resources are not wasted. Precision measures how accurate the detection results are, i.e., the percentage of true positive detections to the total number of detections. In contrast, recall measures how many true positive detections there are concerning the number of all possible detections [62].

$$Precision = \frac{TP}{(TP + FP)} \quad (1)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (2)$$

TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative.

The detection is considered positive if the intersectional ratio of the detected bounding box and the corresponding ground truth bounding box and their union is 50% or higher. This measure is referred to as intersection-over-union (IoU). An example of positive and negative person detection considering IoU >= 0.5 is shown in Figure 8.

To precisely evaluate and characterize the performance of the detector, taking into account not only the accuracy of detection but also the size of objects in the image, six average precision measures in MS COCO format were considered

using the original script[1]:

- AP overall 10 IoU thresholds (0.5: 0.05: 0.95),
- AP50 at IoU = 0.50,
- AP75 at IoU = 0.75.

Average precision across different object scales is evaluated as:

- AP_S for small objects with an area of less than 32^2 px,
- AP_M for medium objects with an area between 32^2 and 96^2px,
- AP_L for large objects with an area of more than 96^2px.

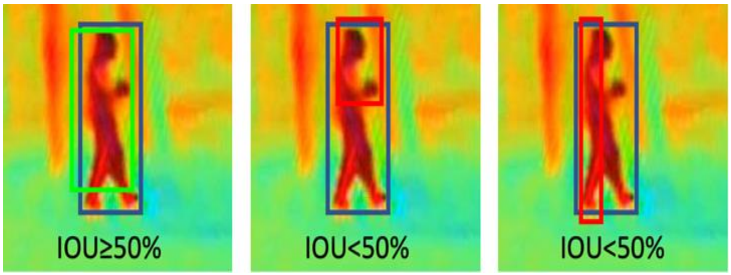


Figure 8. Visual representation of positive (left) and negative (center and right) representation of intersection over union (IoU) criteria equal to or greater than 50% [71].

4. Experiments

A. Preliminary detection results

Preliminary detection results of models of selected state-of-the-art CNN-based object detectors pre-trained on MS COCO dataset on our custom-made SARD test set andVisDrone2000 are given in Table 1. The best results are marked in bold. YOLOv4 achieved significantly better overall results on both test sets considering precision accuracy and object scales.

Table 1. Comparative preliminary detection results on SARD and VisDrone datasets (%).

Test set	Model	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
VisDrone	Cascade R-CNN	8	18	**6**	4	25	48
	Faster R-CNN	8	19	**6**	5	26	43

[1] https://github.com/cocodataset/cocoapi

	RetinaNet	7	15	4	3	22	43
	YOLOv4	**13**	**30**	1	**11**	**36**	**80**
SARD	*Cascade R-CNN*	19	35	18	9	21	43
	Faster R-CNN	20	39	18	10	22	42
	RetinaNet	17	34	14	5	19	**47**
	YOLOv4	**23**	**40**	**25**	**13**	**26**	41

B. Detection performance after training on domain images

To achieve better person detection in the search and rescue scenes, we have also trained the original detectors on the Visdrone data set and on the SARD data set and compared the models' performances.

The MMDetection codebase was used to train the Cascade R-CNN, Faster R-CNN, and RetinaNet models, and the darknet framework model was used to train the YOLOv4 model. The learning rate (lr) was set to 0.005 as the training was performed on a single GPU computer. All other settings are the same as the original model settings. YOLOv4 models are trained on Google Colab with batch = 64 and subdivision = 32, with the network resolution set to 512 x 512. All models are tested on a laptop with one 1660Ti GPU.

After training the model on the selected dataset, each model is referred to in the text as a ***model(dataset)*** to make it easier to compare the models' performances. For example, *Cascade R-CNN (VisDrone2000)* means a Cascade R-CNN detector trained on the VisDrone2000 dataset.

Table 2. Comparative results of models trained and tested on VisDrone2000 dataset (%).

Model	AP	IMP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Cascade R-CNN (VisDrone2000)	17	8,80	38	13	12	39	38

Faster R-CNN (VisDrone2000)	13	4,70	34	8	9	33	28
RetinaNet (VisDrone2000)	8	1,80	26	3	6	20	18
YOLOv4 (VisDrone2000)	**23**	**10,2**	**55**	**15**	**21**	**40**	**34**

1) Transfer learning with the VisDrone dataset

The Cascade R-CNN(VisDrone2000), Faster R-CNN (VisDrone2000) and RetinaNet(VisDrone2000) models were trained in 6 epochs with batch_size set to 1, while the YOLOv4(VisDrone2000) model was trained with max_batches = 6000 and batch = 64.

The detection results on the VisDrone2000 test set for AP are shown in Table 2. The Imp column shows the progress of the model relative to the pre-trained model tested on the same data set.

YOLOv4 (VisDrone2000) achieves an average score of 23% AP which is the best result compared to other tested detectors. Yolo proved to be equally the best in all AP measures related to object size and detection accuracy. By far, the best results of 55.1% AP YOLOv4 achieved on IoU = 0.50.

Cascade R-CNN(VisDrone2000) achieves the second-best results but still significantly worse results than YOLOv4(VisDrone2000). Similar conclusions were reached in [72], [73].

2) Transfer learning with the SARD dataset

When training the models on the SARD set, the same model learning parameters were used as at the Visdrone2000 set. The detection results on the SARD test set are given in Table 3. The best results were obtained with YOLOv4 (SARD), while the results of Cascade R-CNN (SARD) and Faster R-CNN (SARD) detectors are very similar but significantly worse than YOLOv4. All detectors achieve the best results for the case of AP_{50}, with the best results of over 96% achieved by YOLOv4 (SARD). If higher detector precision is required, AP_{75}, all detectors perform significantly worse, with the highest mean accuracy of 71% being achieved again by YOLOv4 (SARD). All detectors' results are significantly

higher on the SARD set than on the VisDrone set and significantly better than the original model when no additional training on domain images was performed.

Table 3. Comparative results of models trained and tested on the SARD dataset (%).

Model	AP	IMP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Cascade R-CNN (SARD)	49	30,0	88	51	31	54	63
Faster R-CNN (SARD)	50	29,9	91	51	30	56	65
RetinaNet (SARD)	34	17,2	73	25	13	41	53
YOLOv4 (SARD)	**61**	**37,9**	**96**	**71**	**45**	**66**	**73**

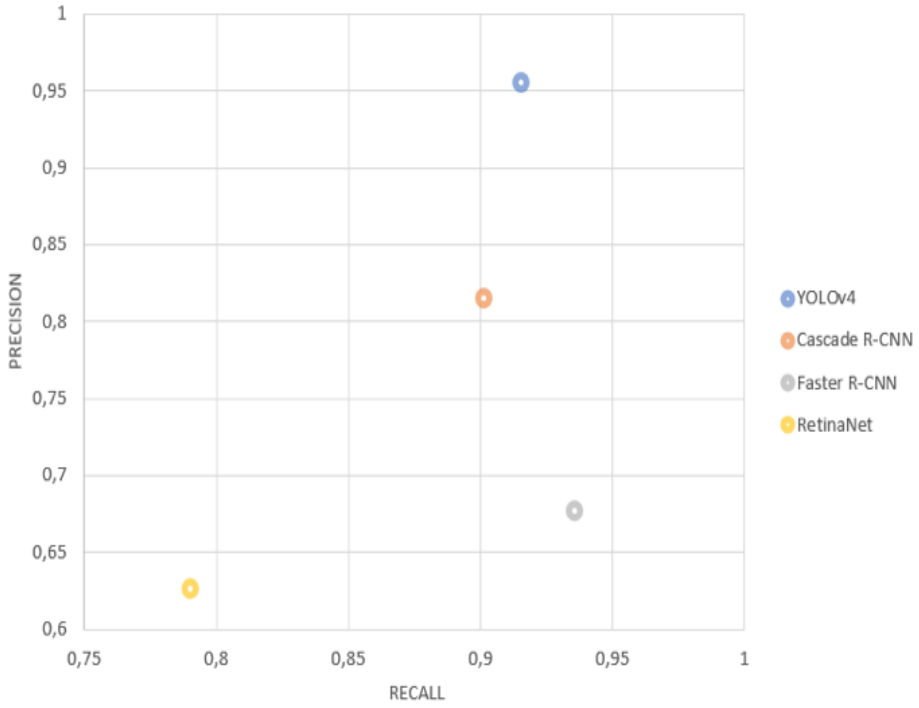


Figure 9. The precision vs. recall ratio for models YOLOv4 (SARD), Cascade R-CNN (SARD), Faster R-CNN (SARD), and RetinaNet (SARD).

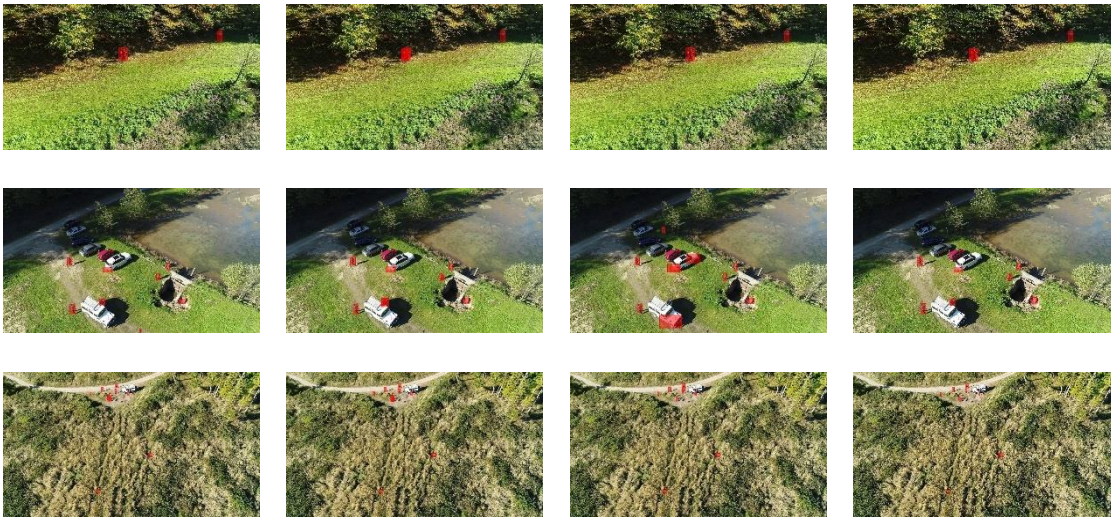
When comparing the detection results concerning the objects' size, it is clear from Table 3 that all detectors achieve significantly better results for large objects than for medium and small objects. The best average accuracy of 73% is achieved by YOLOv4 (SARD) large objects, followed by 66% for medium objects. Faster R-CNN (SARD) and Cascade R-CNN (SARD) perform similarly but score 10%

lower in the case of large and medium objects. For small object detection (APS), YOLOv4 (SARD) proved to be the best with an accuracy of 45%, while Faster R-CNN (SARD) and YOLOv4 (SARD) achieved comparative results, for about 15% lower.

Figure 9. shows the precision and recall ratio for all tested models. The best ratio of precision and recall, with 96% of precision for recall greater than 91% was achieved by YOLOv4 (SARD), which means that it was the most precise in the detection and has detected the most significant number of objects that exist in the image (ground truth). The best recall was achieved by Faster R-CNN (SARD) but with a precision of 67% and much more false positive detections than YOLOv4 (SARD). RetinaNet (SARD) had the lowest precision and the lowest recall.

In search and rescue operations, the goal is to detect all persons present on the scene. Still, on the other hand, the detector's precision is also important so that resources are not wasted on false detections. For this reason, based on the achieved results of average precision and the ratio of precision and recall, the YOLOv4 detector was selected for further research.

Examples of person detection results with models trained on the SARD dataset are shown in Figure 10. The columns in Figure 10 represent the detection results, respectively, in column A) Cascade R-CNN (SARD) model, in column B) Faster R-CNN (SARD) model, C) RetinaNet (SARD), D) YOLOv4 (SARD), and in E) ground truth. All possible detection outcomes appeared in Figure 10.: a positive detection where a person is detected, and IoU of bounding box and person's ground truth is more or equal than 50%, then a negative detection where a person is not detected, or IoU of the bounding box and person's ground truth is less than 50% and a false-positive detection where a part of the image that does not contain a person was marked as a person.



A) B) C) D)

Figure 10. Examples of person detection results of different models retrained on the SARD dataset: A column: Cascade R-CNN (SARD), B column: Faster R-CNN (SARD), C column: RetinaNet (SARD), D column: YOLOv4 (SARD), E column: ground truth.

The first row in Figure 10 shows a quarry case with one person on a pile of rocks while two people are on a dusty road. All detectors successfully detect a person on the road, while only Cascade R-CNN (SARD) and YOLOv4 (SARD) also detect a person sitting on rocks. Faster R-CNN (SARD) has one false detection and multiple detections of a person on the road.

The second row shows an example of three people with an overlap (occlusion) on low grass. All detectors successfully detected the standing person on the top right. Faster R-CNN (SARD) gives multiple detections of overlapping persons. At the same time, Cascade R-CNN (SARD) and RetinaNet (SARD) have occlusion problems and did not detect a person kneeling behind a moving person. YOLOv4 (SARD) successfully detects all persons.

The third scene with eight people was shot from a greater height than the first two examples. Cascade R-CNN (SARD) detects seven individuals with one false detection, Faster R-CNN (SARD) has five accurate detections as well as RetinaNet (SARD), which also has three false detections. YOLOv4 (SARD) precisely detects all persons in the image.

In the last case, taken from an even greater height and distance from the object, nine people are in the tall grass and macadam road. The Cascade R-CNN (SARD and the Faster R-CNN (SARD) accurately detect seven persons while the RetinaNet (SARD) detects only five of them. YOLOv4 (SARD) successfully detects all subjects in the image.

From the qualitative analysis of the selected examples, it is clearly shown that YOLOv4 (SARD) was the most successful in detecting persons in SAR scenarios. However, there are also examples where the YOLOv4 (SARD) model was not successful, Figure 12. The most common examples of false detection are the cases when two people are standing very close to each other or overlap (Figure 12, first row) and when the detector detects darker parts of vegetation (Figure 12, second row) or shadows (Figure 12, third row) as a person. It is almost typical for a person to merge with the background in search and rescue operations practically. In that case, it is challenging to detect a person even for a trained person, so it is not unexpected that the detectors have the most missed detections in that case (Figure 12, third row).

Figure 11. Comparison of different images resolution.

We try to adjust the model parameters and learning conditions to achieve even better detection results with the YOLOv4 detector in the experiment's continuation.

Figure 12. Miss detections of the YOLOv4 (SARD) model (cropped images to make it easier to notice the persons in the image): two people are standing very close to each other or overlapping (first row); darker parts of vegetation detected as a person (second row); shadows detected as persons (third row).

C. Detection results regarding the network resolution

The YOLO architecture resizes the input image, preserving the aspect ratio to the resolution defined in the .cfg weights file, defined by the width and height parameters. These parameters are called network resolution. Transformation of input image resolution in Yolo architecture is given by:

$$Img_{train\_width} = Net_{width},$$
$$Img_{train\_height} = \frac{Net_{width}}{Img_{width}} Img_{heigth} \quad (3)$$

For example, if the input resolution of an image is 1920 x 1080 and the network resolution is defined as width, Net_{width} = 512 height, Net_{height} = 512, YOLO will

change the resolution of the input image to the set width, Net_{width}, preserving the original ratio between image width, Img_{width} and height, Img_{height} e.g., 1920 x 1080 will be transformed to 512 x 288. Comparison of different images resolution is shown in Figure 11.

When done in both train and test sets of the model, this subsampling of image resolution does not violate the general rule of model training since the model was trained on similar object sizes as those that appear in the test set.

To improve the detection performance, especially the detection of small objects, one alternative was to use the higher resolution of input images and train the network at higher resolutions, e.g.:

$$Net_{\overline{width}} = Net_{width} + k, k = 32n, n \in \mathbb{N} \quad (4)$$

Values Net_{width} and Net_{height} that are multiples of 32 can be used, such as 608 x 608 or 832 x 832, because the YOLO network down-samples the input image by 32.

In our case, the input images size is (Img_{train_width}) 1920 x 1080, and the YOLOv4 (SARD) model was trained on (Net_{train_width}) 512 x 512 network resolution. Our computer was too weak to train the network at higher resolutions than that, so the alternative was to increase the network resolution during testing (Net_{test_width}) [74]. The idea was always to use input images of the same resolution of 1920 x 1080 when training and to test the model on higher resolution images without compromising the sizes and ratio among the objects learned during training:

$$\frac{Net_{test_{width}}}{Img_{test_{width}}} = \frac{Net_{train_{width}}}{Img_{train_{width}}};$$
$$\frac{Img_{test_{width}}}{Img_{train_{width}}} = \frac{Net_{test_{width}}}{Net_{train_{width}}} \quad (5)$$

To preserve the ratio (5) for higher image resolution during testing, it was necessary to increase the network resolution. To examine the effect of changing the network resolution during testing (Net_{test_width}) on object detection performance, we have tested different network resolutions below and above the resolution at which the model was trained: 320 x 320, 416 x 416, 512 x 512, 608

x 608, 832 x 832, 1024 x 1024. The network resolutions 320 x 320 and 416 x 416 are below the resolution at which the YOLOv4 (SARD) model was trained, while the resolutions 608 x 608, 832 x 832, 1024 x 1024 are above. The detection results are given in Table 4.

Table 4. YOLOv4 (SARD) detection performance depending on the network resolution (%).

Network resolution,Net_{test}	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	fps
320x320	37	77	31	11	44	68	**10.3**
416x416	51	88	54	26	59	75	9.73
512x512	57	93	63	34	63	**77**	7.37
608x608	60	95	67	39	63	**77**	6.61
832x832	**61**	**96**	**71**	45	**66**	73	3.73
1024x1024	60	95	64	**46**	65	66	2,50

The best accuracy results are achieved for a network resolution of 832 x 832, Table 4, except in the case of large objects (AP_L). A comparison of the results shows that better detection results can be obtained by increasing the network resolution when testing. Better results are achieved at resolutions 608 x 608 and 1024 x 1024 than at a resolution of 512 x 512 at which the model was trained. However, results also show that there is a limit after which the results no longer improve, such as in the case of network resolution of 1024 x 1024, when the results started to decrease.

In the case of testing at the lower resolutions than the network resolution on which the model was trained, in general, worse results are obtained except in the case of the large object where just slightly worse results are achieved. It can be noted that the inference speed is about 10 fps for the lowest network resolution, which is 2.5x faster than at a resolution of 832 x 832, at which the most accurate results are obtained.

The best average precision of 77% is obtained with 512 x 512 pixels and 608 x 608 pixels for large objects. For medium objects, the best average precision is

66% with a network resolution of 832 x 832, and for small objects, the best average precision of 46% is got with a network resolution of 1024 x 1024 pixels.

When approving the resolution of the most suitable model for SAR operations, we were guided by the fact that detection can be performed on-site during flight operations and off-line on the recorded materials since the drone's flight time is limited to the battery life.

In real-time detection on a video received while the drone was flying over the area being searched, the detection speed is important, as well as the model's accuracy. There is also a need to transfer as little data as possible from the drone to the tablet control console. For this mode of use, the most suitable would be a network resolution of 416 x 416 at which the model has 10 fps with an accuracy of only 2% less than the same model at a network resolution of 832 x 832 for larger objects that are likely to be directly detectable in the field, and about 10% less for other cases.

Off-line detection is performed on the recorded materials using a computer with a higher power CPU + GPU. The required detection speed is not crucial in that case, especially if we compare it with about 25 seconds needed for a human video analyst to detect a victim on drone images [50]. In that case, the best model is the one that achieves greater accuracy, and that would be with a resolution of 832 x 832 or 608 x 608 since the differences in performance are negligible.

D. Detection results as a function of TP-FP

We mentioned earlier that in search and rescue operations, the crucial is the accuracy of detection and the speed of finding the missing person. Therefore, it is important to build a model with a few false detections (FP) as possible because they consume human resources and take valuable time.

For this reason, we introduced additional metrics that we called ROpti, computed as the ratio of the difference between true (TP) and false positive (FP) detections and possible detections (TPCFN) in the dataset:

$$ROpti = \frac{(TP + FP)}{(TP + FN)} \tag{6}$$

For perfect precision (no false positive), ROpti is equal to recall, and with perfect recall (no false negative), ROpti is equal to 1, and this is a perfect score. As the number of FPs grows, ROpti decreases. In case TP is equal to FP, then ROpti is equal to zero, ROpti becomes negative, while TP is less than FP.

The detection results considering ROpti measure, e.g., true and false-positive detections out of a total of 2611 objects for different network resolutions with default thresh of 0.25, are given in Table 5.

Considering the ROpti measure, the resolution 832 x 832 surpass all other tested network resolutions as it has only 88 FP and the highest ROpti value of 0.928.

Therefore, we propose a model for detection of persons in SAR actions shown in Figure 13, with 416 x 416 network resolution for on-board detections on videos received from the drone to the control console-tablet (or using RTMP server to live stream from a drone to laptop) and 832 x 832 for further off-line analyses.

Table 5. YOLOv4 (SARD) detection results in terms of a true positive, false negative, and ROpti for different network resolutions.

Model	TP	FP	ROpti
320x320	2088	295	0,687
416x416	2346	184	0,828
512x512	2438	147	0,877
608x608	2485	133	0,901
832x832	2512	**88**	**0,928**
1024x1024	2491	102	0,915

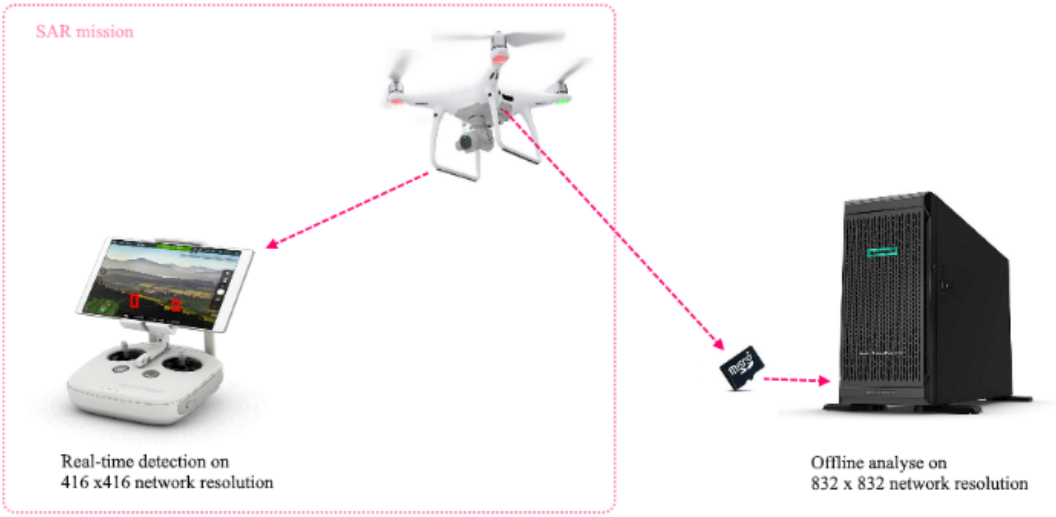


Figure 13. Proposed model for person detection in SAR mission.

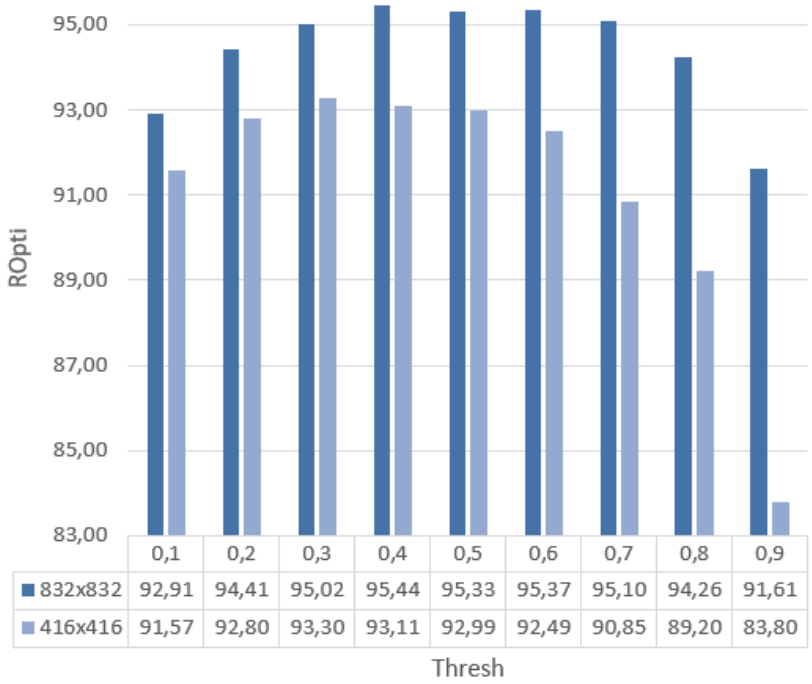
E. Detection results as a function of confidence (thresh) value

In the case of searching for a particular object, any detection that recognizes the object and its location can be taken as a positive detection, regardless of the percentage of the IoU between the ground truth bounding box and the detected bounding box, i.e., precision in terms of the bounding box which is the smallest closure of the object is not so important, so in our case, an IoU of 10% is also acceptable. Decreasing the IOU value and confidence value of the model affects the accuracy of the detection, and this, in turn, affects the model usability for automatic detection of persons in SAR missions. On the other hand, the goal is to achieve as few false-positive detections as possible, i.e., achieve the highest possible ROpti value, so the limit to which it is still effective to decrease the confidence or threshold value needs to be determined.

By default, YOLO detects objects with a confidence (threshold) of 0.25 or more. This value directly affects the number of marked objects in the set, so we examined how the thresh value changes affect the ROpti value.

Figure 14. shows detection results for thresh in the range from 0.10 to 0.90 with a step of 0.10 in two network resolutions 832 x 832 and 416 x 416. The best results were achieved when the network resolution was 832 x 832, and the thresh

was 40%, so this is the configuration we would recommend for the model for person detection in SAR scenes.



	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
832x832	92,91	94,41	95,02	95,44	95,33	95,37	95,10	94,26	91,61
416x416	91,57	92,80	93,30	93,11	92,99	92,49	90,85	89,20	83,80

Figure 14. The ROpti value for the YOLOv4(SARD) model with network resolutions 832 x 832 and 416 x 416 considering different confidence values (thresh).

With a network resolution of 416 x 416, results are 1 to 8% worse than with 832 x 832, but with 2.5 times shorten detection times, so this network setting with thresh = 0.10 can be recommended as a reasonable solution in on-board online detection when speed and a small amount of data are important. To improve the ROpti results and reduce the number of FP detections in real-time, the drone pilot can "remove" false-positive detections by lowering the drone to a lower altitude when necessary to capture larger objects.

F. Detection dependence of recording height

The altitude at which the drone is located plays a major role in detecting people in aerial photographs. The higher the altitudes at which the drone flies, the smaller the captured material and fewer pixels are used to represent them. However, at higher altitudes, the drone can capture a larger terrain area. In the case of SAR operations, it makes no sense to increase the flight altitude above the level at which persons can be detected. Obviously, it is easier to detect a person represented in the image with a larger number of pixels, so it will be more suitable

for detecting people when the drone is flying at a lower altitude. But this extends the time required to cover the target search area. Therefore, the goal is to determine the highest possible altitude at which the drone should fly so that people on the scene can still be detected automatically by a detector.

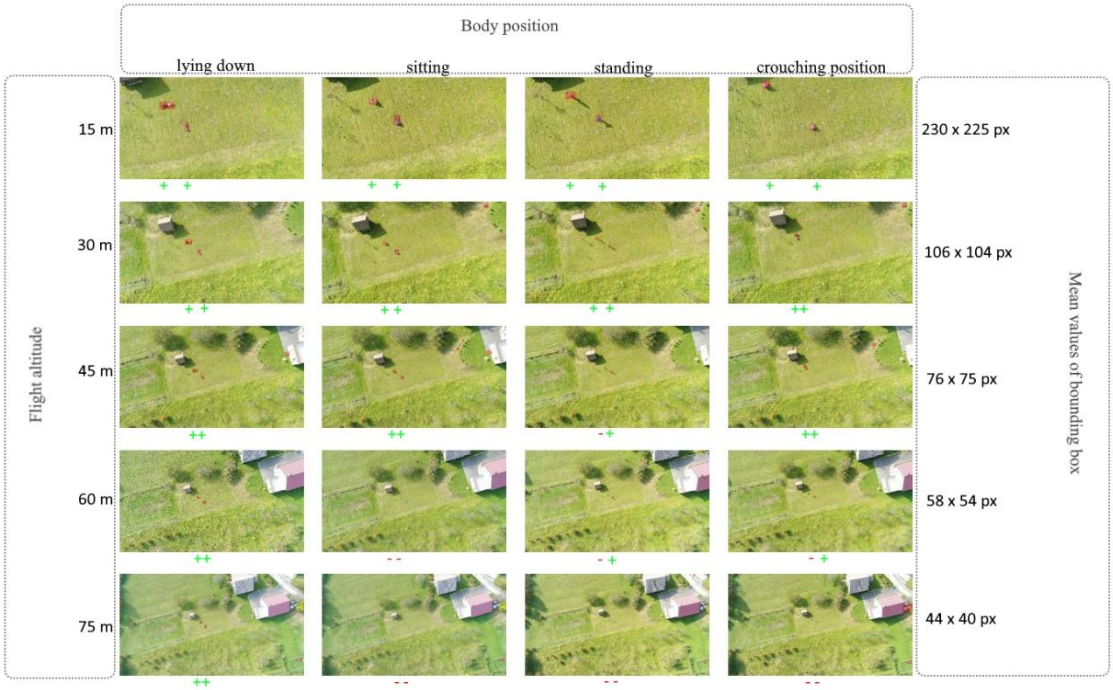


Figure 15. Detection results on different drone heights (15 m, 30 m, 45 m, 60 m, and 75 m) show that below or equal to 30 m of height all detections are accurate.

Flight altitude recommendations depend on the number of pixels in the camera and the lenses used, and the area being monitored. With DJI Phantom 4 Advance, we record images at a resolution of 5472 x 3078 px with a camera angle of 90°, Field of View (FOV) by specification is 84°.

In the experiment, we took images of two persons (women and a boy) at different heights (15 m, 30 m, 45 m, 60 m, and 75 m). Figure 15. shows detection results, and it can be seen that all detections are accurate at the height of 30 m. Therefore, considering that there are different specifications of drone cameras, we suggest that the drone flies at a height from which it can capture images in which people occupy an area of 100 x 100 px.

G. Robustness to weather conditions and motion blur

To test the YOLOv4 (SARD) model's performance with a network resolution of 832 x 832 in conditions that can occur in search and rescue operations, we have tested the model's performance on the Corr test set. The Corr test set includes images with various weather conditions such as snow, fog, frost (Corr-Snow, Corr-Fog, Corr-Frost), and motion blur that may occur during recording, e.g., due to moving and camera shake (Corr-M. Blur).

Table 6. Comparative results of YOLOv4(sard) and YOLOv4(sardCcorr) on corr dataset and its parts concerning different weather conditions (%).

Model	TEST	AP	IMP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv4(SARD)	SARD	**61**		**96**	**71**	**45**	**66**	**73**
	Corr	35,5		65,7	34,7	20,8	39,4	53,3
	Corr-Snow	32,5		59,0	32,9	18,0	35,8	56,9
	Corr-Fog	30,2		55,0	30,4	22,5	32,2	40,4
	Corr-Frost	35,9		62,9	37,1	22,5	39,3	50,8
	Corr-M. blur	31,6		67,8	24,7	14,7	35,1	58,1
YOLOv4(SARD+Corr)	SARD	**59,4**		**94,7**	**67,4**	**42,2**	**64,7**	**72,8**
	Corr	51,9		89,5	53,0	32,7	57,1	69,0
	Corr-Snow	50,3	17,8	88,5	51,5	33,4	54,7	65,1
	Corr-Fog	54,7	24,5	91,6	60,2	38,2	59,5	65,3
	Corr-Frost	53,1	18,2	90,5	57,8	36,7	57,5	66,5
	Corr-M. blur	43,9	12,3	84,9	41,0	24,4	49,4	61,6

The examination results are given in Table 6 in terms of average precision (AP), respecting IoU precision and the object size. The results show several important facts.

A significant decrease in detection performance occurred in the case of testing on images with bad weather conditions and blur images that did not exist in the training set. e.g., the decrease in AP_{50} was from 96% on SARD set to 66% on the Corr dataset that contains the same images but with bad weather conditions. The

drop in performance is not the same for all bad weather conditions, e.g., AP_{50} is 59% for snow, 55% for fog, 63% for frost, and 68% for motion blur.

To improve the YOLOv4 (SARD) model results in bad weather, we additionally trained the model on the Corr train set, referred to as the YOLOv4 (SARD+Corr) model. The YOLOv4 (SARD+Corr) model achieves similar or slightly worse results than the YOLOv4 (SARD) model on the SARD test set and significantly better results on the Corr test set. Detection results are presented in Table 6.

Examples of detection results of the YOLOv4 (SARD+Corr) model for all different weather categories and motion blur are shown in Figure 16.

Figure 16. Example of detection of YOLOv4 (SARDCCorr) model. Up-left snow, up-right fog, down left ice, down right motion blur.

H. Transfer learning strategy

To improve model training, we wanted to investigate further how different transfer learning strategies regarding different combinations of datasets affect the detection result. We examined the possibility of learning the models successively, on one training set and then on the other, taking into account the order of sets used for training or in one step but using the images taken from both training sets.

Table 7. Detection results for YOLOv4 model trained on different sets and tested on SARD test set and mixture of SARD and VisDrone2000 test sets (%).

TRAIN	TEST	AP	IMP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_S	AR_M	AR_L	ROpti
SARD	SARD	61,3	37,90	95,7	71,1	45,0	66,4	72,6	52,3	72,1	**77,8**	92,8
VisDrone2000	SARD	18,9	-4,5	33,2	20,5	13,2	21,6	17,4	14,8	23,6	19,5	30,0
Corr	SARD	54,9	31,5	90,5	61,9	35,1	61,3	66,8	42,8	67,2	73,1	85,9
S+V	SARD	22,8	-0,6	41,7	23,7	16,4	25,5	23,0	19,1	28,3	25,8	35,4
V+S	SARD	61,3	37,9	95,8	70,6	46,2	66,3	71,5	53,1	71,8	76,6	**93,5**
V+C+S	SARD	**62,0**	**38,6**	**95,9**	**71,9**	**46,9**	**66,9**	**72,1**	**53,9**	**73,2**	77,1	92,6
SC	SARD	59,4	36,0	94,7	67,4	42,2	64,7	72,8	49,5	69,9	76,7	90,9
SV	SARD	55,4	32,0	92,5	60,8	38,4	60,6	67,1	46,0	66,0	71,7	86,2
SVC	SARD	56,4	33,0	93,6	63,1	39,9	61,5	67,3	47,5	66,8	71,7	88,2
S+V	SV	23,7	8,3	52,9	17,4	21,0	32,9	24,8	28,6	38,3	28,1	33,1
V+S	SV	18,7	3,3	35,7	17,5	9,9	48,2	**96,9**	12,9	53,6	74,6	26,3
SV	SV	**29,7**	**14,3**	**61,7**	**24,6**	**22,3**	**52,4**	65,8	**30,0**	**58,3**	70,3	**40,3**

The goal was to get the best possible results of the YOLOv4 model at the SARD test set. Firstly, to train the model, SARD, VisDrone, and Corr sets were used separately, and then combinations of them. The results achieved by training the models at different training sets using one by another in a different order, or mixed, are shown in Table 7. In addition to the accuracy values, the improvement (Imp) of the model concerning the initial weights (original model) and ROpti value are also shown.

The S + V means that the model is first trained on the SARD train set and then on the VisDrone train set, V + S that it is first trained on VisDrone, then on the SARD train set, and for V + C + S, the model is trained on VisDrone2000, and then on Corr and finally on SARD train set.

The SV refers to a mixture of SARD and VisDrone2000 train sets when images are used randomly from both of them for training, while for testing purposes, the SV test refers to a combination of SARD and VisDrone test images. Similarly, the

SC is a mixture of SARD and Corr set, and SVC is a mixture of SARD, VisDrone2000, and Corr test set.

It can be observed that the improvement of the results is achieved in the case when the last trained set is the closest to the tested set (S + V vs. V + S). Also, training models with more data from multiple sets ultimately contribute to a better result, especially if the sets are compatible, i.e., contain similar images. In this experiment, the best results (AP 62%, AP_S 46.9%) were achieved when learning the model on sets in the order V + C + S, but this is an improvement of only 1% than in the case when the model was trained only on the SARD set of images, which is certainly not a significant improvement.

The V + S model achieves the same results on the SARD test set as the model trained only on the SARD set for all cases except for smaller objects. The V + S model gets better results since a larger number of smaller objects from VisDrone2000 train set were included in the V + S training set.

Training the model on data from a mixture of sets (SV, SC, SVC) had given worse results than when the model was trained only on the SARD set or on a series of sets ending with SARD so that the weights of the model are last adjusted to the set being tested.

The same conclusion applies when the model is tested on images from multiple sets, e.g., the SV set. The best results are achieved when the model is trained on a particular combination of these sets.

5. Conclusion

The ability to detect people on drone images using computer vision methods automatically is a significant help in SAR operations. In this paper, we explored the state-of-the-art person detectors in drone images and proposed a model for detecting persons in SAR actions.

We have re-trained and tested CNN-based object detectors, Cascade R-CNN, Faster R-CNN, RetinaNet, and YOLOv4 on selected drone images in the VisDrone set and our custom-made set of SAR-s scenes.

YOLOv4 has achieved the best detection performances on the SARD dataset in terms of average precision (AP) considering IoU precision and the object size as well as the least false detection (FP), so it was further used in the experiment, referred to as YOLOv4 (SARD). When the model was trained on 512 x 512 image resolution, the best AP of 60% was achieved for a network resolution of 832 x 832.

In SAR operations, the model must have a few false detections (FP) as possible that resources are not wasted unnecessarily, so we introduced an additional metric called ROpti, calculated as the ratio of the difference between true and false positive detections and possible detections in a dataset.

In searching for a missing person, the most important thing is that the detector locates that person, and it is less important how accurate the detection is. We experimentally selected parameters as a trade-off between accuracy and recall so that the model can be helpful in SAR actions. The results showed that the YOLOv4 (SARD) model in a network resolution of 832 x 832, IoU = 0.1, achieved the best results for thresh of 0.4, namely AP of 97.15% (TP: 2538, FP: 46).

The model's robustness was tested on images with artificially generated bad weather conditions and image blur, and the results show a severe decrease in AP in more than 30%. After the model was also trained on the part of the images with bad weather effects, the model achieves significantly better results (AP 50.3% for snow, 54.7% fog, 53.1% ice, 43.8% motion blur).

In future work, the plan is to use a thermal camera to increase detection performance and develop a model for recognizing human activity (running, walking, standing, sitting, lying down) and tracking people in SAR scenes.

References

[1] M. Šuperina and K. Pogačić, "Učestalost Hrvatske gorske službe spašavanja u traganju za nestalim osobama," Policija i sigurnost, 16(3-4), 2007, 235-256.

[2] G. Milani, M. Volpi, D. Tonolla, M. Doering, C. Robinson, M. Kneub¨uhler, and M. Schaepman, “Robust quantification of riverine land cover dynamics

by high-resolution remote sensing," Remote Sensing of Environment, vol. 217, pp. 491–505, 2018.

[3] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in Advances in neural information processing systems, 2015, pp. 91-99.

[4] Z. Cai and N. Vasconcelos, "Cascade r-CNN: Delving into high-quality object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6154-6162.

[5] T. Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, "Focal loss for dense object detection," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980-2988.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, A. C. Berg, "SSD: Single shot multi-box detector," in European conference on computer vision, Springer, Cham, 2016, pp. 21-37.

[7] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement." arXiv preprint arXiv:1804.02767, 2018.

[8] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In ECCV, 2014.

[9] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, "The pascal visual object classes challenge: A retrospective," International journal of computer vision, 2015, 111(1), 98-136.

[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, `ImageNet large scale visual recognition challenge," Int. J. Comput. Vis., vol. 115, no. 3, pp. 211252, Dec. 2015.

[11] M. Pobar, M, Ivasic-Kos, Mask R-CNN, and Optical flow-based method for detection and marking of handball actions, 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2018.

[12] M. Ivasic-Kos and M. Pobar, “Building a labeled dataset for recognition of handball actions using mask R-CNN and STIPS,” in 2018 7th European Workshop on Visual Information Processing (EUVIP), 2018, pp. 1–6.

[13] M. Ivasic-Kos, M. Kristo, and M. Pobar, "Human Detection in Thermal Imaging Using YOLO," in Proceedings of the 5th ACM International Conference on Computer and Technology Applications, ICCTA 2019, NY, USA, 2019, pp.20-24.

[14] M. Ivasic-Kos, M. Kristo and M. Pobar, "Person Detection in thermal videos using YOLO," Proceedings of SAI Intelligent Systems Conference. Springer, Cham, 2019.

[15] P. Zhu, L. Wen, D. Du, X. Bian, Q. Hu, and H. Ling, "Vision Meets Drones: Past, Present, and Future," arXiv preprint arXiv:2001.06303, 2020.

[16] M. Barekatain, M. Martí, H. F. Shih, S. Murray, K. Nakayama, Y. Matsuo, H. Prendinger, “Okutama-action: An aerial view video dataset for concurrent human action detection,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 28-35.

[17] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, Q. Tian, “The unmanned aerial vehicle benchmark: Object detection and tracking,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 370-386.

[18] M. R. Hsieh, Y. L. Lin, W. H. Hsu, “Drone-based object counting by spatially regularized regional proposal network,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4145-4153.

[19] A. Robicquet, A. Sadeghian, A. Alahi, S. Savarese, “Learning social etiquette: Human trajectory understanding in crowded scenes,” in European conference on computer vision, Springer, Cham, 2016, pp. 549-565.

[20] M. Mueller, N. Smith, B. Ghanem, “A benchmark and simulator for UAV tracking,” in European conference on computer vision, Springer, Cham, 2016, pp. 445-461.

[21] K. Butorac, M. Šuperina and L. Mikšaj-Todorović, "Developing Police Search Strategies for Elderly Missing Persons in Croatia," Varstvoslovje, 17(1), 2015.

[22] R. J. Koester, Lost Person Behavior: A Search and Rescue. DBS Productions LLC, 2008.

[23] A.S.Laliberte and A.Rango, “Textureandscaleinobject-based analysis of subdecimeter resolution unmanned aerial vehicle (UAV) imagery,” IEEE Transactions on Geoscience and Remote Sensing, vol. 47, no. 3, pp. 761–770, 2009.

[24] G. Pajares, “Overview and current status of remote sensing applications based on unmanned aerial vehicles (UAVs),” Photogrammetric Engineering & Remote Sensing, vol. 81, no. 4, pp. 281–330, 2015.

[25] A. Bhardwaj, L. Sam, F. Mart´ın-Torres, and R. Kumar, “UAVs as remote sensing platform in glaciology: Present applications and future prospects,” RemoteSensingofEnvironment, vol.175, pp.196–204, 2016.

[26] S. Harwin and A. Lucieer, “Assessing the accuracy of georeferenced point clouds produced via multi-view stereopsis from unmanned aerial vehicle (UAV) imagery,” Remote Sensing, vol. 4, no. 6, pp. 1573–1599, 2012.

[27] S. Yahyanejad and B. Rinner, “A fast and mobile system for registration of low-altitude visual and thermal aerial images using multiple smallscale UAVs,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 104, pp. 189–202, 2015.

[28] N. Tijtgat, W. Van Ranst, T. Goedeme, B. Volckaert, and F. De Turck, “Embedded real-time object detection for a UAV warning system,” in IEEE International Conference on Computer Vision Workshop (ICCVW), 2017.

[29] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, “Deep learning in remote sensing: A comprehensive review and list of resources,” IEEE Geoscience and Remote Sensing Magazine, vol. 5, no. 4, pp. 8–36, 2017.

[30] L. Mou, Y. Hua, P. Jin and X. X. Zhu, "ERA: A Dataset and Deep Learning Benchmark for Event Recognition in Aerial Videos," arXiv preprint arXiv:2001.11394, 2020.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXivpreprint arXiv:1409.1556, 2014.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[33] S. Xie, R. Girshick, P. Dollar, Z. Tu and K. He, "Aggregated residual transformations for deep neural networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1492–1500.

[34] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.

[35] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proceedings of the IEEE Conference on Computer Vision and pattern recognition (CVPR), 2018, pp. 4510–4520.

[36] A. Howard et al, "Searching for MobileNetV3," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019.

[37] A. Kuznetsova et al, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," arXiv preprint arXiv:1811.00982, 2018.

[38] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779-788.

[39] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," Proceedings of the IEEE conference on computer vision and pattern recognition, 2017.

[40] A. Bochkovskiy, C. Y. Wang and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.

[41] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580–587.

[42] R. Girshick, "Fast R-CNN," Proceedings of the IEEE international conference on computer vision, 2015.

[43] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object detection via region-based fully convolutional networks,” in Advances in neural information processing systems, 2016, pp. 379–387.

[44] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117-2125.

[45] S. N. A. M. Ghazali, H. A. Anuar, S. N. A. S. Zakaria, Z. Yusoff, “Determining position of target subjects in maritime search and rescue (MSAR) operations using rotary-wing unmanned aerial vehicles (UAVs),” in 2016 International Conference on Information and Communication Technology (ICICTM), IEEE, 2016, pp. 1-4.

[46] P. Doherty, P. Rudol, “A UAV search and rescue scenario with human body detection and geolocalization,” in Australasian Joint Conference on Artificial Intelligence, Springer, Berlin, Heidelberg, 2007, pp. 1-13.

[47] M. A. Goodrich, B. S. Morse, C. Engh, J. L. Cooper, J. A. Adams, “Towards using unmanned aerial vehicles (UAVs) in wilderness search and rescue: Lessons from field trials,” Interaction Studies, 2009, 10(3), 453-478.

[48] S. Waharte, N. Trigoni, “Supporting search and rescue operations with UAVs,” in 2010 International Conference on Emerging Security Technologies, IEEE, 2010, pp. 142-147.

[49] C. A. Baker, S. Ramchurn, W. T. Teacy, N. R. Jennings, “Planning search and rescue missions for UAV teams,” in Proceedings of the Twenty-second European Conference on Artificial Intelligence, IOS Press, 2016, pp. 1777-1778.

[50] K. Yun, L. Nguyen, T. Nguyen, D. Kim, S. Eldin, A. Huyen, E. Chow, “Small target detection for search and rescue operations using distributed deep learning and synthetic data generation,” in Pattern Recognition and Tracking XXX (Vol. 10995, p. 1099507), International Society for Optics and Photonics, 2019.

[51] A. J. Gallego, A. Pertusa, P. Gil, R. B. Fisher, “Detection of bodies in maritime rescue operations using unmanned aerial vehicles with multispectral cameras,” Journal of Field Robotics, 2019, 36(4), 782-796.

[52] R. Geraldes, A. Gonçalves, T. Lai, M. Villerabel, W. Deng, A. Salta, H. Prendinger, “UAV-based situational awareness system using deep learning,” IEEE Access, 2019, 7, 122583-122594

[53] S. O. Murphy, C. Sreenan, K. N. Brown, “Autonomous unmanned aerial vehicle for search and rescue using software-defined radio,” in 2019 IEEE 89th Vehicular Technology Conference VTC2019-Spring, 2019, pp. 1-6. IEEE.

[54] E. Lygouras, N. Santavas, A. Taitzoglou, K. Tarchanidis, A. Mitropoulos, A. Gasteratos, “Unsupervised human detection with an embedded vision system on a fully autonomous UAV for search and rescue operations,” Sensors, 2019, 19(16), 3542.

[55] F. S. Leira, T. A. Johansen, T. I. Fossen, “Automatic detection, classification and tracking of objects in the ocean surface from UAVs using a thermal camera,” in 2015 IEEE aerospace conference, IEEE, 2015, pp. 1-10.

[56] J. Sun, B. Li, Y. Jiang, C. Y. Wen, “A camera-based target detection and positioning UAV system for search and rescue (SAR) purposes,” Sensors, 2016, 16(11), 1778.

[57] Z. Kashino, G. Nejat, B. Benhabib, “Aerial wilderness search and rescue with ground support,” Journal of Intelligent & Robotic Systems, 2019, 1-17.

[58] T. Marasović, V. Papić, “Person classification from aerial imagery using local convolutional neural network features,” International Journal of Remote Sensing, 2019, 1-19.

[59] A. Al-Kaff, M. J. Gómez-Silva, F. M. Moreno, A. de la Escalera, J. M. Armingol, “An appearance-based tracking algorithm for aerial search and rescue purposes,” Sensors, 2019, 19(3), 652.

[60] S. E. Wei, V. Ramakrishna, T. Kanade, Y. Sheikh, “Convolutional pose machines,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4724-4732.

[61] S. Sambolek, M. Ivasic-Kos, "Detection of Toy Soldiers Taken from a Bird’s Perspective Using Convolutional Neural Networks," ICT Innovations 2019, Ohrid, Springer Communications in Computer and Information Science, 2019.

[62] M. Ivasic-Kos, I. Ipsic, and S. Ribaric, "A knowledge-based multi-layered image annotation system," Expert systems with applications 42 (24), pp. 9539-9553.

[63] LabelImg [Online] Available: https://github.com/tzutalin/labelImg

[64] C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker and W. Brendel, "Benchmarking robustness in object detection: Autonomous driving when winter is coming," arXiv preprint arXiv:1907.07484, 2019.

[65] Faster R-CNN mmdetection models. [Online]. Available: https://github.com/open-mmlab/mmdetection/tree/master/configs/faster_rcnn

[66] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN, "in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 390-391.

[67] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence TPAMI, 37(9), 2015, pp. 1904–1916.

[68] S. Liu, L. Qi, H. Qin, J. Shi and J. Jia, "Path aggregation network for instance segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR, 2018, pp. 8759–8768.

[69] RetinaNet mmdetection models. [Online]. Available: https://github.com/open-mmlab/mmdetection/tree/master/configs/retinanet

[70] Cascade R-CNN mmdetection models. [Online]. Available: https://github.com/open-mmlab/mmdetection/tree/master/configs/cascade_rcnn

[71] M. Kristo, M. Ivasic-Kos, M. Pobar. Thermal Object Detection in Difficult Weather Conditions Using YOLO, 2020, IEEE Access 8, 125459-125476

[72] D. R. Pailla, “VisDrone-DET2019: the vision meets drone object detection in image challenge results, 2019.

[73] S. Sambolek, M. Ivašić-Kos, „Detecting objects in drone imagery: a brief overview of recent progress“, Mipro 2020, Opatija.

[74] Yolo-v4 and Yolo-v3/v2 for Windows and Linux. [Online]. Available: https://github.com/AlexeyAB/darknet

[75] M. Pobar, M. Ivasic-Kos, "Active Player Detection in Handball Scenes Based on Activity Measures," Sensors 20 (5), 1475.

RAD 5. TRANSFER LEARNING METHODS FOR TRAINING PERSON DETECTOR IN DRONE IMAGERY

Ovaj rad je objavljen kao: Sambolek, Saša, and Marina Ivašić-Kos. Transfer Learning Methods for Training Person Detector in Drone Imagery. In: Arai, K. (eds) Intelligent Systems and Applications. IntelliSys 2021. Lecture Notes in Networks and Systems, vol 295. Springer, Cham. https://doi.org/10.1007/978-3-030-82196-8_51

https://link.springer.com/chapter/10.1007/978-3-030-82196-8_51

1. Introduction

Deep learning methods have been successfully applied in many computer vision applications in recent years. Unlike traditional machine learning methods, deep learning methods allow automatic learning of features from data and reduce manual extraction and presentation features. However, it should be emphasized that the deep learning model is highly data-dependent. Large amounts of data are needed in the learning set to detect patterns among the data, generate features of the deep learning model, and identify the information needed to make a final decision.

Insufficient data to learn deep learning models are a significant problem in specific application domains such as search and rescue (SAR) operations in non-urban areas. The process of collecting relevant image data, in this case, is demanding and expensive because it requires the use of drones or helicopters to monitor and record non-urban areas such as mountains, forests, fields, or water surfaces. The additional problem is that scenes with detected casualties rarely appear on the recorded material, which is the most useful for learning the model for detecting an injured person. Besides, the data collected should be processed, each frame inspected, and each occurrence of a person marked with a bounding box and labeled, which is a tedious and time-consuming process.

One way to overcome the problem of data scarcity is to use transfer learning. Transfer learning allows a domain model not to be learned from scratch, assuming that the learning set data is not necessarily independent and identically distributed as the data in the test set. This assumption makes it possible to significantly reduce the amount of data required in the learning set and the time required to learn the target domain model.

This paper aims to detect persons on the scenes of search and rescue (SAR) operations. Today, it has become commonplace to use drones in SAR missions that fly over the search area and film it from a bird’s eye view. They can capture a larger area at higher altitudes, but then the people in the image are tiny and take up only a few pixels. People can be detected more efficiently at lower altitudes, but in that case, the field of view is smaller. People who are searched

for are very often barely noticeable because of the branches and trees, occluded by some vegetation, in the shadow, fused with the ground, which further complicates the search even for favorable weather conditions. During SAR operations, the drone operator has a demanding task to analyze the recorded material in real-time to detect a relatively small person on a large, inaccessible surface that requires great concentration, so automatic detection can be valuable.

We used the YOLOv4 model for the person detector trained on the MS COCO dataset, which proved to be the most successful in previous research after additional learning on domain images [1–3].

To train theYOLOv4 model, we used the custom-made set of SARD scenes that were shot in a non-urban area with actors simulating injured people and prepared for machine learning. To increase the set, we have generated the Corr-SARD set from SARD scenes by adding atmospheric conditions. Since tailor-made SARD and Corr-SARD datasets were relatively small for learning deep learning models, we have additionally used the VisDrone dataset to include more images of people taken by drone, although not in non-urban areas.

This paper examined three different transfer learning methods for building YOLOv4 models for detecting persons in search and rescue operations. In the next section, three different methods of transfer learning will be presented. In the third section, the experimental setup is given along with the description of image data sets SARD, Corr-SARD, VisDrone, and basic information about the YOLO4 detector. In the fourth section, the experimental results of applying different transfer learning methods will be presented and compared. In conclusion, we list important characteristics regarding the impact of different transfer learning approaches on person detection in search and rescue scenes and a plan for future research.

2. Transfer Learning

Transfer learning involves taking a pre-trained neural network and adapting that neural network to a new distinct set of data by transferring or repurposing the learned features. Transfer learning is beneficial when learning models with limited

computing resources and when a modest set of data is available for model learning.

Many state-of-the-art models took days, or even weeks, on powerful GPU machines to train them. So, to not repeat the same procedure over a long time, learning transfers allow us to use pre-trained weights as a starting point.

Different levels and methods of applying deep transfer learning can be classified into four categories according to [4]: network-based transfer learning, instance-based transfer learning, mapping-based transfer learning, and adversarial-based transfer learning, which we will not examine here.

2.1 Network-Based Deep Transfer Learning

Network-based deep transfer learning refers to the reuse of a part of the network (without fully connected layers) previously trained in the source domain and is used as part of the target network used in the target domain [4].

The CNN architecture contains many parameters, so it is difficult to learn so many parameters with a relatively small number of images. Therefore, for example, in [5], the network is first trained on a large set of data for classification (ImageNet, source domain), and such pre-trained parameters of the inner layers of the network are transferred to the target tasks (classification, detection, domain target). An additional network layer was added and trained on the labeled target set data to minimize the differences between the source and the target data regarding various image statistics (object type, camera position, lighting) and fit the model to the target data task.

Suppose the source domain and the target domain differ in scenes. In that case, the objects' appearance, lightings, background, position, distance from the camera, and similar lower detection results can be expected on target sets than achieved on the source. For example, the original model of the YOLO object detector trained on the COCO data set was used for detecting players in video frames of handball sports [6] and for person detection on thermal images [7]. In the case of player detection in handball scenes, the original YOLO model achieved an AP of 43.4%, which is often better than person detection in thermal

images, where an AP of 19.63% was achieved. Lower results on thermal images are due to significant differences between thermal and RGB images. Lower detection results on handball scenes were achieved since the detector did not accurately identify the player and often drop to mark a high-raised hand or leg in the jump, as handball-specific poses did not exist in the original set.

2.2 Instances-Based Deep Transfer Learning

Instance-based deep transfer learning refers to a method in which a union of selected instances from the source domain and instances of the target domain is used for training. It is assumed that regardless of differences in domains, the source domain's instances will improve detections in the target domain.

In deep learning, the approach of fine-tuning models on the target domain, which are pre-trained on large benchmark datasets of source domains, is standard to improve results in other similar target domains. The authors in [8] use an instance-based deep transfer approach to measure each training sample's impact in the target domain. The primary purpose was to improve the model's performance in the target domain by optimizing its training data. In particular, they use a selected pre-trained model to assess each training sample's impact in the target domain. According to the impact value, remove negative samples and thus optimize the target domain's training set.

In the previously mentioned research in the sports domain [6] and thermal images [7], itwas shown that additional learning at the appropriate set and fine-tuning the parameters of the pre-trained model to tasks of interest could significantly improve the detection results at the target set. Thus, the basic model's AP on the set of thermal images with AP 19.63% with additional adjustment on the customized set of thermal images achieved AP of 97.93%. In additional learning in the handball scenes, AP increased from an initial 43% to 67%. Similar results after fine-tuning with state-of-the-art backbone deep neural networks such as Inception v2, ResNet 50, ResNet 101 were also reported in [9].

2.3 Mapping-Based Deep Transfer Learning

Mapping-based deep transfer learning refers to mapping instances from the source domain and the target domain to a new data space [4]. Mapping-based deep transfer learning finds a common latent space in which feature representations for the source and target domains are invariant [10]. In [11], a CNN architecture was proposed for domain adaptation by introducing an adaptation layer for learning feature representations. The maximum mean discrepancy (MMD) metric is used to calculate the overall structure's distribution distance concerning a particular representation, which helps select the architecture's depth and width and regulate the loss function during fine-tuning. Later, in [12] and [13], a multiple kernel variance of MMD was proposed (MKMMD) and joint MMD (JMMD) to improve domain adaptation performances. However, the main limitation of the MMD methods is that the computational cost of MMD increases quadratically with the number of samples when calculating Integral Probability Metrics (IPM) [14]. Therefore, Wasserstein distance has recently been proposed in [15] as an alternative for finding better distribution mapping.

2.4 Adversarial-Based Deep Transfer Learning

Adversarial-based deep transfer learning mainly refers to introducing adversarial technology inspired by generative adversarial networks (GAN) [16] to find transferable representations that apply to both the source and target domain but can also refer to the use of synthetic data used to enlarge the original dataset artificially.

In adversarial networks, the extracted features from two domains (source and target) are sent to the adversarial layer that tries to discriminate the features' origin. If there is a slight difference between the two types of features, the adversarial network achieves worse performance, and it is a signal for better transferability, and vice versa. In this way, general features with greater portability are revealed in the training process.

In the case of using synthesized data in order to increase the learning set of the deep learning model, it is necessary to analyze the content of the reference video

scene and select elements to be generated on the virtual scene taking into account the background, objects on the scene and accessories, such as [17].

3. Experimental Setup

3.1 Dataset

In this paper, three datasets were used: the publicly available VisDrone dataset, custom made SARD dataset and synthetically enlarged SARD dataset, Corr-SARD datasets.

From the VisDrone dataset [18] containing images of urban scenes taken by the drone, we selected 2,129 images that include a person or pedestrian tag. We combined both labels into one class: person. The obtained dataset was divided into a training set (1,598 images) and a test set (531 images). The selected dataset from the VisDrone set includes shots of people taken under different weather and lighting conditions in different urban scenarios such as roads, squares, parks, parking lots, and the like.

The SARD dataset [19] was recorded in a non-urban area to show persons in scenes specific to search and rescue operations. The set contains footage simulating poses of injured people found in inaccessible terrains in the hills, forests, and similar places by searching and rescuing actions and standard poses of people such as walking, running, sitting. The set contains 1,981 images divided into two subsets, a training set containing 1,189 images and a test set with 792 images.

The Corr-SARD dataset is derived from the SARD set so that the effects of snow, fog, frost, and motion blur are added to the SARD images. The training set has the same number of images as the SARD training set, while the test set has slightly fewer images (714) because images in which no persons are seen after adding the effect have been removed.

For the experiment, we created an additional three datasets containing images of the sets mentioned above. The SV refers to a mixture of SARD and VisDrone

sets. Similarly, the SC is a mixture of SARD and Corr set, and SVC is a mixture of SARD, VisDrone, and Corr test set.

Figure. 1. Example of images from SARD dataset.

3.2 YOLOv4 Person Detection Model

Detection of persons in high-resolution images taken by a drone is a challenging and demanding task. People who are searched for due to loss of orientation, fall, or dementia are very often in unusual places, away from the road, in atypical body positions due to injury or fall, lying on the ground due to exhaustion, covered with stones due to slipping or landslides (Fig. 1). On top of all that, the target object is relatively small and often camouflaged in the environment, so it is often challenging to observe.

In this experiment, for person detection, we used the YOLOv4 model [20]. YOLOv4 uses CSPDarkNet53 as a backbone [21] that includes the DarkNet53, a deep residual network with 53 layers, and the CSPNet (Cross Stage Partial Network). To increase the receptive field without causing a decrease in velocity, the authors added Spatial Pyramid Pooling SSP [22] as the neck, and PAN, Path Aggregation Network [23] for path aggregation, instead of the Pyramid Feature Network (FPN) used in YOLOv3. The original YOLOv3 network is used for the head [24].

In addition to the new architecture, the authors also used training optimization called “Bag of Freebies” to achieve greater accuracy without additional hardware costs, such as CutMix, Mosaic, CIoU-loss, DropBlock regularization. There is also a “Bag of Specials” set of modules that only slightly increase the hardware costs with a significant increase in detection accuracy.

To train and evaluate the YOLOv4 model, we used the Darknet framework [25], an open-source neural network framework written in C and CUDA that supports CPU and GPU computing. For the experimentation, we used Google Colab [26], a free tool for machine learning and local computer Dell G3 i7-9750H CPU, 16 GB RAM, GeForce GTX 1660 Ti 6 GB, with Ubuntu 16.04. 64-bit operating system.

3.3 Evaluation Metrics

We use average accuracy (AP) to evaluate the detection results. AP is a metric that considers the number of correctly and incorrectly classified samples of a particular class and is used to determine the detection model’s overall detection power, not just accuracy [27]. In this experiment, we have used three precision measures in the MS COCO format that takes into account detection accuracy (IoU):

- AP thresholds of 10 IoU (0.5: 0.05: 0.95),
- AP50 at IoU = 0.50,
- AP75 in IoU = 0.75.

The original COCO script was used to calculate the results.

4. Results of Transfer Learning Methods and Discussion

This section presents the overall performance results from the conducted experiments. It is worth mentioning that the pre-trained YOLOv4 with weights (yolov4.conv.137 [25]) learned on the MS COCO [28] dataset was trained on three training datasets with different transfer learning methods to identify the transfer learning variant that provides the best solution for person detection in SAR scenes.

In all cases, the YOLOv4 model was trained with a batch size of 64, a subdivision of 32, and iterations of 6000. The learning rate, momentum, and decay for the training process were set to values of 0.001, 0.949, 0.0005, and width and height to value 512.

Before training, the parameters of the original model should be changed and adapted to our domain. The first step is to change the number of classes from 80, which corresponds to the number of MS COCO classes, to 1 class, a person in this experiment. After defining the class size, each Conv filter must be set to 18 as defined in (1), where the class corresponds to the number of classes (class = 1 in our case).

$$x\ filters = \ (classes + 5)\ x\ 3 \tag{1}$$

The impact of applying each of the transfer learning methods in training the detection model on the detectors' results in search and rescue operations is given below.

4.1 Fine-Tuning the YOLOv4 Model to the Target Domain

In the network-based deep transfer learning, the pre-trained YOLOv4 model trained on the COCO source domain was fine-tuned to the target domain: SARD, VisDrone, or Corr-SARD dataset. The sketch of network-based deep transfer learning is shown in Fig. 2.

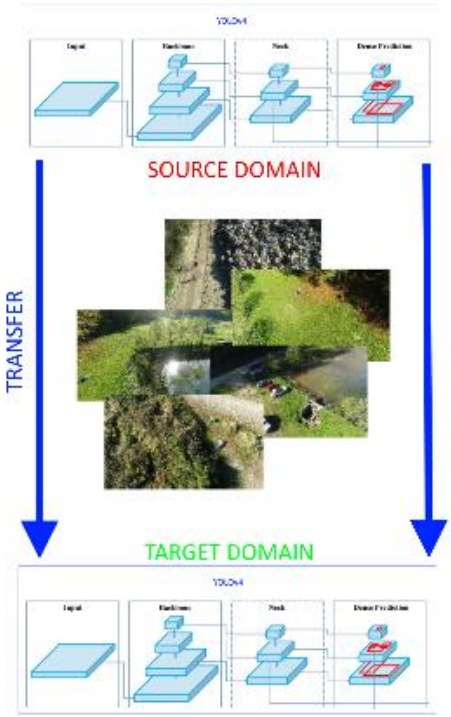


Figure. 2. A network-based deep transfer learning: the first network was trained in the source domain (in our case MS COCO), and then the pre-trained network was fine-tuned on the target domain (SARD dataset).

For a more straightforward presentation of the results, the model trained on the SARD training dataset was designated as the SARD model. The model labeled COCO refers to the pre-trained model on the MS COCO dataset.

Table 1 shows the results of person detection on SARD images concerning the AP metric with the original YOLOv4 model and theYOLOv4model that was further trained on SARD images. The results show a significant improvement in AP (Imp 37,9) and AP50 and AP75 metrics of the detection results after fine-tuning the model to the SARD dataset.

Table 1. Results of YOLOv4 models on SARD test dataset in case of network-based deep transfer learning

Model	AP	AP_{50}	AP_{75}	Imp
COCO	23.4	40.2	25.3	
SARD	61.3	95.7	71.7	37.9

4.2 Instances-Based Deep Transfer Learning with SARD, Corr-SARD, and VisDrone Datasets

After we applied the network-based transfer learning, we applied several instance-based deep transfer learning to train further the YOLOv4 model, including a series of sets (VisDrone and Corr-SARD and SARD).

Using the VisDrone set, we selected only those instances from that set relevant to our target domain, i.e., those that contained a person. In the VisDrone training set that we used, there is approximately the same number of images as in the SARD training set, but in the VisDrone set, there are 25,876 objects more than in the SARD dataset that is 29,797 marked persons in VisDrone and 3,921 marked persons in SARD dataset.

In the first case of instance-based transfer learning, the original model was trained first on a selected part of the VisDrone dataset and then fine-tuned on the SARD training dataset (V + S model). The sketch of instances-based deep transfer learning with VisDrone and SARD dataset is shown in Fig. 3.

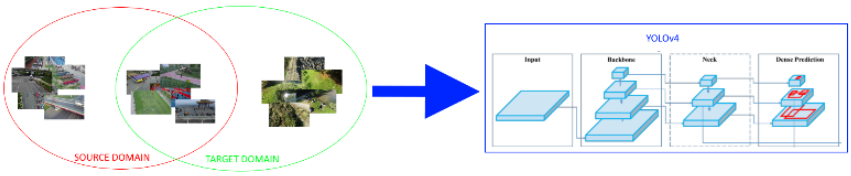


Figure. 3. Instance-based deep transfer learning. We selected only images relevant to our target domain and trained the model with it from the source domain. In the second step, the model was trained on the SARD dataset.

According to the results presented in Table 2, additional model training on the VisDrone set (model V + S) did not affect the detection results obtained on the SARD model. However, it improved the results compared to the original model (Imp 37,9).

Training on the Corr-SARD training dataset contributed to a slight improvement in detection results concerning the SARD model and significant AP improvement to the original model (Fig. 4).

Also, the results show that transfer learning is not commutative and that the order of the sets used to train the model affects the detection results. The best results

are achieved when the model is fine-tuned on the dataset on whose examples it will be tested, so the V + S model achieves significantly better results than the S + V model.

We also tested instance-based deep transfer learning using three datasets so that the original model was fine-tuned on the SARD training set after training on VisDrone, and the Corr-SARD datasets (V + C + S model).

Table 2. Results of YOLOv4 models on SARD test set to build with instance-based transfer learning

Model	AP	AP_{50}	AP_{75}	Imp
S + V	22.8	41.7	23.7	-0.6
V + S	61.3	95.8	70.6	37.9
V + C + S	**62.0**	**95.9**	**71.9**	**38.6**

Table 3 shows the individual detection results on the SARD test set obtained when the original model was additionally trained on the VisDrone and Corr-SARD sets. For an easier results notation, a model trained on the VisDrone dataset is designated as VisDrone, and the model trained on the Corr-SARD as Corr-SARD.

The results are interesting and show that fine-tuning the original model to the VisDrone set even lowered the detection results even though the original COCO dataset does not include shots of people taken by the drone. The VisDrone set includes them just like the target SARD test set, but in urban areas. The use of the synthetic Corr-SARD set contributed to improved person detection outcomes in the SARD test set.

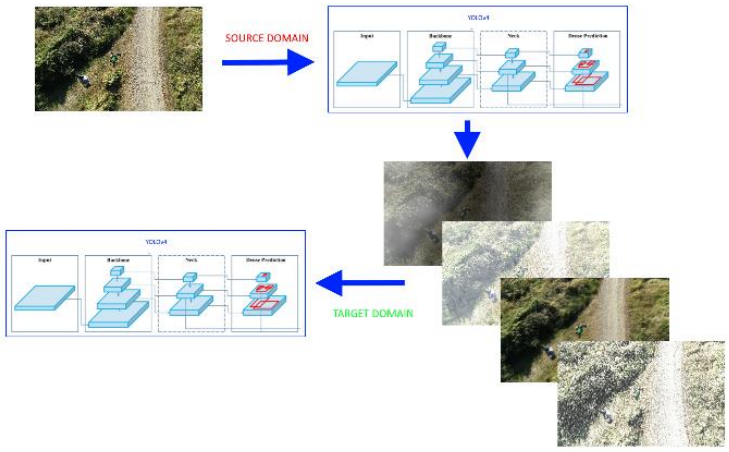


Figure. 4. Using Corr-SARD dataset for transfer learning. After training on the SARD dataset, the model was re-train with the same images with added effect.

Table 3. Results of YOLOv4 models on SARD test dataset after learning on the VisDrone set and Corr-SARD set

Model	AP	AP_{50}	AP_{75}	Imp
VisDrone	18.9	33.2	20.5	-4.5
Corr-SARD	54.9	90.5	61.9	31.5

4.3 Mapping-Based Deep Transfer Learning with Images from SARD, Corr-SARD, and VisDrone Datasets

In mapping-based deep transfer learning, several new sets were made for training the model as a union of images from the VisDrone, SARD, and Corr-SARD training sets. These are the SV sets created as a union of images from the SARD training set and VisDrone set, the SC model created by merging images from the SARD training set and Corr-SARD, and the SVC set created as a union of images from all three sets. A sketch of mapping-based deep transfer learning is shown in Fig. 5.

The results in Table 4 show that transfer learning on newly created sets (SV, SC, SCV) significantly contributed to the improvement of the detection result concerning the original model with a relatively high AP score achieved: for SC model 59.4%, SV 55.4%, and SVC 56.4%. The AP increase after transfer learning the model on new sets is 32 to 36 percent higher than with the original model (Imp column in Table 4). However, it can be noticed that the results of the model

trained on the newly created sets SV, SC, SCV are comparable but still slightly lower than the case when the model was fine-tuned only on the training data from the target set (model SARD).

Table 4. Results of YOLOv4 model on SARD test set to build with mapping-based transfer learning methods

Model	AP	AP_{50}	AP_{75}	Imp
SV	55.4	92.5	60.8	32.0
SC	59.4	94.7	67.4	36.0
SVC	56.4	93.6	63.1	33.0

From the obtained results, it can be concluded that in the case of deep transfer learning based on mapping, relatively good AP results were achieved, but that results are still worse compared to deep transfer learning based on instances and network transfers. Overall, the best AP score of 62.0% was achieved with the V + C + S model, and immediately afterward, with the AP 61.3%, a SARD model was fine-tuned only on the SARD training set.

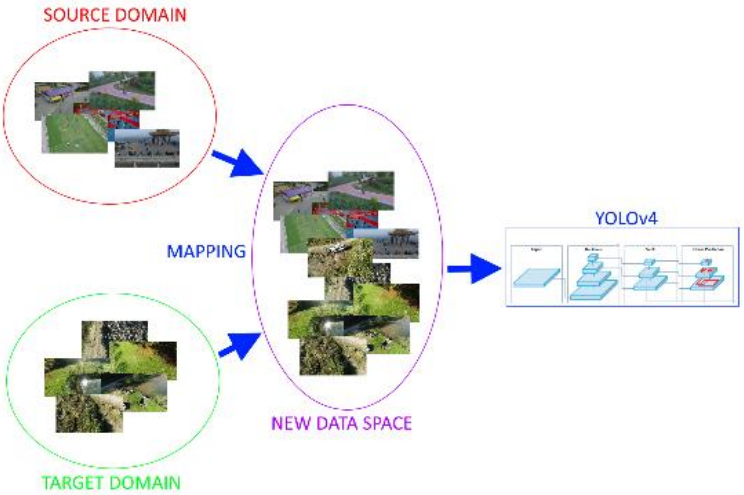


Figure. 5. Mapping-Based Deep Transfer Learning. Images from the Target SARD Dataset are Mapped with Images from the VisDrone and Corr-SARD Datasets.

Additionally, to evaluate the performance of the SV, SC, SCV models built with mapping-based transfer learning on the appropriate test sets, additional testing of the models was done on the test sets generated in the same way as SV, SC, SCV training sets but from the corresponding test sets.

Table 5. Results of YOLOv4 models build with mapping-based transfer learning on appropriate test sets

Model	Test set	AP	AP_{50}	AP_{75}
SV	SV test	29.7	61.7	24.6
SC	SC test	55.8	91.6	61.7
SVC	SVC test	31.7	64.4	27.9

The obtained results of the models obtained with the mapping-based transfer learning tested on the testing part of SV, SC, SCV sets are shown in Table 5 and have worse results than when tested only at the set SARD test set.

The SC model achieved a minor difference in performance on the SC test set, comparing the SARD test set's detection results. This was expected because the Corr-SARD set images included in the SC test set are those from the SARD set only with the added effects of bad weather.

5. Conclusions

In this paper, transfer learning approaches to improve person detection on drone images for the SAR mission were examined. We have fine-tuned the YOLOv4 model using different transfer learning methods on three datasets: a tailor-made SARD set for SARD missions, a VisDrone drone-recorded dataset in urban places, and a Corr-SARD dataset with synthetically added weather effects on SARD images.

We compared and discussed the impact of the transfer learning methods used in YOLOv4 model training on detection results. Testing was performed on the target dataset SARD and the newly created datasets SV, SC, and SVC, created by merging the initial sets.

The results show that the best detection results are achieved on the target SARD domain using network-based transfer learning when the set on which the model is finetuned is equally distributed as the set on which the model is tested. The best results were achieved by applying the network transfer learning method,

which transmits features obtained on large data sets, and the instance-based transfer learning method, in which the model is trained on images of the domain corresponding to the images on which the model will be tested. The use of synthetic image instances further improved the performance of the model.

From the results, we also see that the worst results were obtained when the datasets were merged because, in that case, the model could not fully adapt to the data of interest. However, this way, by increasing the learning data, a more general model can be achieved. It has been shown that when training models with multiple datasets, it is not insignificant whether we train with all images simultaneously or individually on each set and the sets' order during training.

For future work, we plan to explore the impact of different transfer learning methods on various application domains and determine the key characteristics of learning datasets that positively impact model performance. Also, we are interested in further exploring different network strategies for selecting, merging, and changing network layers to improve detection results.

References

1. Sambolek, S., & Ivasic-Kos, M.: Detection of toy soldiers taken from a bird's perspective using convolutional neural networks. In International Conference on ICT Innovations (pp. 13-26). Springer, Cham. (2019, October).
2. Sambolek, S., & Ivasic-Kos, M.: Person Detection in Drone Imagery. In 2020 5th International Conference on Smart and Sustainable Technologies (SpliTech) pp. 1-6. IEEE. (2020, September).
3. Kristo, M., Ivasic-Kos, M.. & Pobar, M.: Thermal Object Detection in Difficult Weather Conditions Using YOLO, IEEE Access, 2020, pp. 125459-125476
4. Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C.: A survey on deep transfer learning. In International conference on artificial neural networks (pp. 270-279). Springer, Cham (2018, October).
5. Oquab, M., Bottou, L., Laptev, I., & Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In

Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1717-1724. (2014).

6. Buric, M., Pobar, M., Ivasic-Kos, M.: Adapting YOLO network for ball and player detection, 8th International Conference on on Pattern Recognition Applications and Methods, 2019, pp. 845-851
7. Ivasic-Kos, M., Kristo, M., & Pobar., M. :Human detection in thermal imaging using YOLO, 5th International Conference on Computer and Technology Applications 2019, p. 20-24
8. Wang, T., Huan, J., & Zhu, M.: Instance-based deep transfer learning. In 2019 IEEE Winter Conference on Applications of Computer Vision (WACV) (pp. 367-375). IEEE (2019, January).
9. Pobar, M., & Ivasic-Kos, M.: Active Player Detection in Handball Scenes Based on Activity Measures, Sensors, 20 (5), 2020, pp. 1475.
10. Cheng, C., Zhou, B., Ma, G., Wu, D., & Yuan, Y.: Wasserstein distance based deep adversarial transfer learning for intelligent fault diagnosis. arXiv preprint arXiv:1903.06753 (2019).
11. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., & Darrell, T.: Deep domain confusion: Maximizing for domain invariance. arXiv preprint arXiv:1412.3474 (2014).
12. Long, M., Cao, Y., Wang, J., & Jordan, M.: Learning transferable features with deep adaptation networks. In International conference on machine learning (pp. 97-105). PMLR (2015, June).
13. Long, M., Zhu, H., Wang, J., & Jordan, M. I.: Deep transfer learning with joint adaptation networks. In International conference on machine learning (pp. 2208-2217). PMLR (2017, July).
14. Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A.: A kernel two-sample test. The Journal of Machine Learning Research, 13(1), 723-773 (2012).
15. Arjovsky, M., & Chintala, S.: Bottou. Wasserstein gan. arXiv preprint arXiv:1701.07875, 7 (2017).

16. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y.: Generative adversarial networks. arXiv preprint arXiv:1406.2661. (2014).
17. Buric, M., Paulin, G., Ivasic-Kos, M.: Object Detection Using Synthesized Data, ICT Innovations 2019, Web Proceedings, 2019.
18. Zhu, P., Wen, L., Bian, X., Ling, H., & Hu, Q. Vision meets drones: A challenge. arXiv preprint arXiv:1804.07437. (2018).
19. Sambolek, S., & Ivasic-Kos, M.: Detecting objects in drone imagery: a brief overview of recent progress. In 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO). pp. 1052-1057. IEEE. (2020).
20. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934. (2020).
21. Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H.: CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 390-391. (2020).
22. He, K., Zhang, X., Ren, S., & Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9), 1904-1916. (2015).
23. Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J.: Path aggregation network for instance segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8759-8768. (2018).
24. Redmon, J., & Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767. (2018).
25. Darknet, https://github.com/AlexeyAB/darknet, last accessed 2021/02/21.
26. Google Colab, https://colab.research.google.com/, last accessed 2021/02/21.
27. Padilla, R., Netto, S. L., & da Silva, E. A.: A survey on performance metrics for object-detection algorithms. In 2020 International Conference on Systems, Signals and Image Processing (IWSSIP). pp. 237-242. IEEE. (2020, July).
28. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L.: Microsoft coco: Common objects in context. In European

conference on computer vision pp. 740-755. Springer, Cham. (2014, September).

RAD 6. PERSON DETECTION AND GEOLOCATION ESTIMATION IN UAV AERIAL IMAGES: AN EXPERIMENTAL APPROACH

Ovaj rad je objavljen kao: Sambolek, Saša, and Marina Ivašić-Kos. "Person Detection and Geolocation Estimation in UAV Aerial Images: An Experimental Approach." Proceedings of the 13th International Conference on Pattern Recognition Applications and Methods - ICPRAM; ISBN 978-989-758-684-2; ISSN 2184-4313, SciTePress, pages 785-792. DOI: 10.5220/0012411600003654

This research was partially supported by HORIZON EUROPE Widening INNO2MARE project (grant agreement ID: 101087348).

https://www.scitepress.org/PublicationsDetail.aspx?ID=ptBwsiKhlSk=&t=1

1. Introduction

Object detection is a key research area within computer vision, focusing on the precise positioning and recognition of various objects in the image (Zou et al., 2023). Despite achieving promising results in ground-level object detection, the task of object detection in aerial images is still a challenge, especially in its application in search and rescue (SAR) operations (Sambolek & Ivasic-Kos, 2021) whose primary objective is to assist as soon as possible to the casualty and save human lives.

SAR is carried out on different terrains such as mountains, rivers, lakes, canyons. The speed of finding a missing person directly affects their chances of survival, so unmanned aerial vehicles (drones) equipped with RGB cameras and sensors are nowadays commonly included in the search missions. The search area is inspected during the flight and offline with the subsequent analysis of the recorded material if the missing person is not found during the online search. In both cases, artificial intelligence can help track down the missing person, however, the automatic detection of victims is still a challenge (Andriluka et al., 2010; Bejiga et al., 2017; Doherty & Rudol, 2007; Geraldes et al., 2019; Shakhatreh et al., 2019; Sun et al., 2016). When analyzing the recorded material, it is crucial not only to detect the person in the images, but also to estimate the distance of the person from the drone and to geolocate it so that a SAR mission can be organized accordingly.

The primary goal of this paper is to evaluate the effectiveness of the latest version of the widely used YOLO object detector, YOLOv8 (Ultralytics, n.d.-c), in detecting people in drone images. Using the publicly available SARD dataset (Sambolek & Ivasic-Kos, 2021) adapted for object detection in SAR, we fine-tuned different models of the Yolov8 family and conducted an in-depth analysis and comparison of drone-captured person detection performance. In addition, we have built custom SARDAG_overflight dataset for developing and testing the algorithm for determining the geolocation of a detected person.

The structure of this paper is as follows: Section 2 provides an overview of previous research related to YOLO object detectors and person geolocation

algorithms. The YOLOv8 family of models and the performance achieved after fine-tuning on the customized SARD dataset are described in Section 3, followed by a description of the geolocation algorithms proposed for use in SAR missions. The experimental part of the work and the metrics used are presented in Section 4 along with the results and explanation. The concluding section highlights the main contributions of this paper.

2. Related works

For our proposed method of detection and geolocation of persons in SAR missions, the object detector and the geolocation algorithm are key. In the following, we will focus on the review of the state-ofthe-art CNN detectors from the YOLO family (Redmon et al., 2016), which are an example of single-stage detectors that constantly achieve top performance in real time, and algorithms for deterministic geolocation.

2.1 YOLO Object Detectors

The most popular and stable version of YOLO, showcasing improved performance with multi-scale prediction frameworks and a deep backbone network, was introduced by Redmon and Farhadi (Redmon & Farhadi, 2018). Bochkovskiy et al. (Bochkovskiy et al., 2020) developed YOLOv4, which featured significant new features, outperforming YOLOv3 in terms of accuracy and speed. (Ultralytics, n.d.-a) introduced YOLOv5, along with a PyTorch-based variant, bringing remarkable improvements. In 2022, the Meituan Vision AI Department unveiled YOLOv6 (Li Chuyi et al., 2022). YOLOv6 features an efficient backbone, RepVGG or CSPStackRep blocks, PAN topology gates, and efficient separate heads with a hybrid channel strategy. The model also employs advanced quantization techniques, including post-training quantization and channel distillation, resulting in faster and more accurate detectors. In July of the same year, YOLOv7 (Chien-Yao Wang, Alexey Bochkovskiy, 2023) outperformed all existing object detectors in terms of speed and accuracy. It follows the same COCO dataset training approach as YOLOv4 but introduces architectural changes and improvements that enhance accuracy without compromising

inference speed. The most recent version of the YOLO family released in January 2023 is YOLOv8 (Ultralytics, n.d.-c) designed for speed and precision for various computer vision applications (Ultralytics, n.d.-c). The architecture of YOLOv8 can be divided into two main components: the backbone and the head. The backbone is like the YOLOv5 model and contains the CSPDarknet53 architecture with 53 convolutional layers, but with the change in the building blocks of the C3 module. The module is now called C2f and all outputs from the gate (bottleneck – 3x3 convolutions with residual connections) were chained, while in C3 only the output from the last gate was used. In the neck, the features are connected directly without forcing the same channel dimensions, which reduces the number of parameters and the total size of the tensor. The head of YOLOv8 consists of several convolutional layers, followed by fully connected layers responsible for predicting bounding boxes, objectivity (probability that the bounding box contains an object), and class probabilities for recognized objects. For class probabilities, the softmax function is used, while the output layer uses the sigmoid function as the activation function.

The loss functions used by YOLOv8 for improving detection, especially when working with smaller objects are: CIoU (Complete Intersection over Union) and DFL (Distribution Focal Loss) for bbox-related losses, and binary cross-entropy for classification loss.

YOLOv8 uses an anchor-free model with a decoupled head for independent object detection, classification, and regression processing. This design allows each branch to focus on its task and contributes to improving the overall accuracy of the model

2.2 Target Geolocation Algorithms

To calculate the geolocation of objects in the image, an algorithm based on the Earth ellipsoid model is usually used, (Leira et al., 2015; Sun et al., 2016; Wang et al., 2017; Zhao et al., 2019) which uses information about the average height, the field of view of the camera, the width and height of the image, the tilt of the camera and the position of the detected point within the image. This algorithm is easy to calculate, but it is not precise because it considers the average elevation

information as the reference height for the target, which leads to significant positional inaccuracies, especially in regions with significant topographic relief. Figure 1 shows the positioning of the target on the Earth's surface according to the model of the Earth's ellipsoid and the errors that arise due to the difference in the geodetic heights of the point from which the drone took off and the point where the detected person is located. In the given scenario, the SAR operation would be carried out at position P' instead of at position P where the person is actually located.

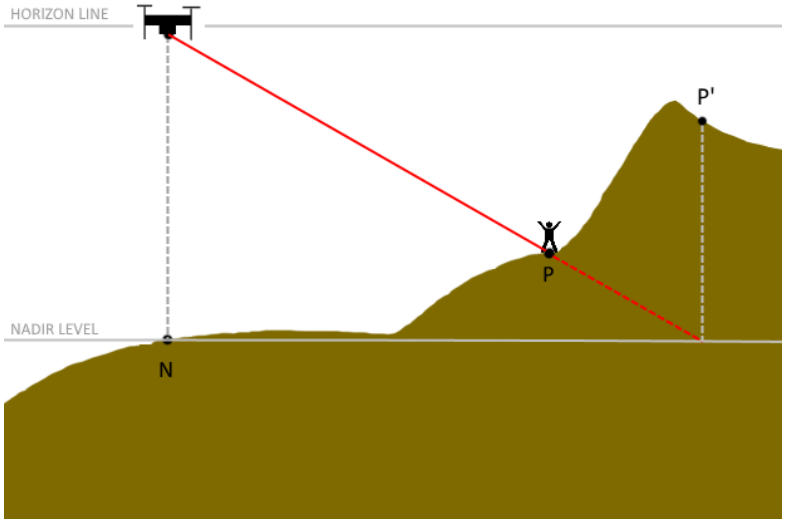


Figure 1: Schematic diagram of target geolocating error using the Earth ellipsoid model in areas with uneven terrain.

In the case of geographically complex terrains, data that rely on the Digital Elevation Model (DEM) (El Habchi et al., 2020), (Huang et al., 2020) can be used. DEM includes a database of the height of any location on Earth, expressed in relation to sea level. In (Paulin et al., 2024) a methodology for precise geolocation using DEM and the RayCast method was introduced and it was shown that the use of DEM significantly increases the accuracy of person positioning on complex terrain.

Another approach focused in reducing the elevation error includes two-point shooting on known GPS positions (I1 and I2 on Fig. 2) at a single target and a direction vector that usually depends on angle sensor of drone camera (Qu et al., 2013), (Xu et al., 2020). This algorithm can only be used for geolocation of

stationary targets because its accuracy is significantly degraded when the target moves. The solution is the approach in (Bai et al., 2017), which uses two drones at positions I1 and I2, for simultaneous recording of the same target and determination of the cross-section and the position of the target. However, this algorithm is not applicable for the case of SAR due to the additional cost of the drone that should record the same search area and due to the safety issue where the simultaneous use of the same airspace by multiple drones is avoided to reduce the risk of collision.

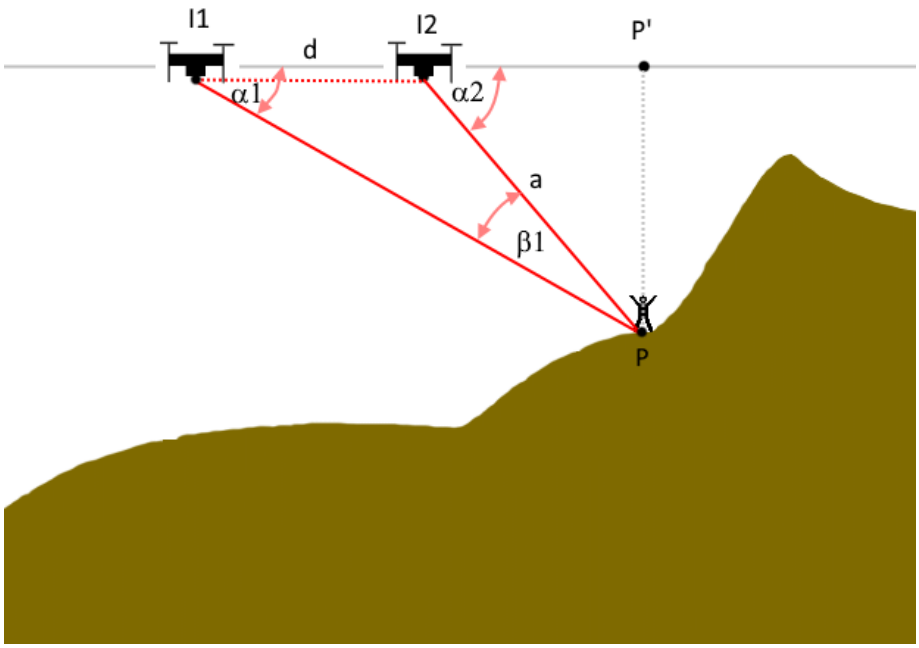


Figure 2: Two-point intersection positioning model

3. Person detection and geolocation in SAR missions

3.1 YOLOv8 for person detection

The YOLOv8 is engineered with a focus on improving performance of real-time detection of objects of various sizes while reducing inference time and computing requirements (Ultralytics, n.d.-c) which makes it potentially interesting for use in SAR missions that generally have small objects of interest and limited resources.

The YOLOv8 is presented in five distinct scaled versions with different number of free parameters: YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. The

YOLOv8n has the simplest architecture with 3 million parameters, while YOLOv8x, has 68 million parameters and shows the best performance within the shortest time (Table 1.).

Table 1. Comparison of five YOLOv8 models, trained and evaluated on the COCO test-dev 2017 dataset with 640 px input, according to the report from (Ultralytics, n.d.-b).

Version of YOLO	mAP 50-95	Speed CPU ONNIX (ms)	Speed A100 TensorRT (ms)	params (M)
YOLOv8n	37.3	80.4	0.99	3.2
YOLOv8s	44.9	128.4	1.20	11.2
YOLOv8m	50.2	234.7	1.83	25.9
YOLOv8l	52.9	375.2	2.39	43.7
YOLOv8x	53.9	479.1	3.53	68.2

We have fine-tuned all five versions of the YOLOv8 model on the SARD dataset adapted for object detection in SAR with two changes to the original architecture: the input to the network was changed to dimensions of 640 for images of 640x360 pixels, and the output, to one class (a person).

3.2 Geolocation estimation

In SAR missions, it is very often the case that missing persons are motionless because they are injured and/or exhausted. Therefore, we propose a geolocation intersection measurement algorithm for locating missing person, that relies on the analysis of multiple shots taken by a single drone and uses terrain configuration data to reduce geolocation error. The algorithm starts to be used after a person is detected in an image, and then an intersection is determined with each subsequent image in which there is also a detected person. In Figure 2, label d is the distance between two drone positions from which the images were captured. Angles α1 and α2 are determined in the same manner as in (Sambolek & Ivašić-Kos, n.d.). By applying the same rule, we calculate the length of side I1P, which is the distance from the drone to the person (point P) when the first image was taken, and the length of side I2P (length a in Figure 2, equation 1), represents the distance from the location where the second image was taken. Then, from the triangle I2PP', we determine the length of side I2P' (Eq. 2), based on which we calculate the GPS coordinates of point P, considering known GPS coordinates of the drone's position and the azimuth toward point P.

$$\frac{a}{\sin \alpha 1} = \frac{d}{\sin \beta 1} \tag{1}$$

$$\overline{I2P'} = a \cdot \cos \alpha 2 \tag{2}$$

Geolocation results is the distance in meters between two points at Earth according to the current standard WGS 84 that is reference system used by the GPS and identifies an Earth-centered, Earth-fixed coordinate system with absolute accuracy of 1-2 meters. The mean error (Eq. 4) indicates the average value of all distances ΔPi (Eq. 3) calculated between predicted geolocation of detected points and the GT point, P_{GT_i} for each image in the dataset.

$$\Delta P_i = Geodesic.WGS84.Inverse(P_i, P_{GT_i}) \tag{3}$$

$$Mean\ Error = \frac{\sum_{i=1}^{n} \Delta P_i}{n} \tag{4}$$

4. Experiments

4.1 Datasets

In our study, we used two datasets, SARD and SAR-DAG_overflight. The SARD dataset was used for training the YOLOv8 model for person detection, while the SAR-DAG_overflight dataset was prepared for the validation of the geolocation algorithm of detected persons.

4.1.1 SARD - dataset for training detector

The SARD dataset was designed with a particular focus on detecting missing or injured persons captured by drones in non-urban terrains. The data was recorded by a DJI Phantom 4 Advanced drone in continental Croatia and includes 1,981 images with a total of 6,532 people. Examples of images from the SARD set are shown in Figure 3.

Figure 3: Examples of detections on images from the SARD dataset with an enlarged image to better emphasize the person in the image that needs to be detected.

The images from the SARD set are of 640 x 360 resolution and are evenly distributed in a ratio of 60:40 into a training set and a validation set based on various factors such as background, lighting, person pose, and camera angle. The training set contains 1,189 images with 3,921 tagged persons, while the validation set contains 792 images with 2,611 tagged persons (Sambolek & Ivasic-Kos, 2021).

In this experiment, we removed from the training set all images that contained a frame with a person with an area of less than 102 pixels, which significantly saved the amount of computer time during training without negatively affecting the performance of the model. After this intervention, the training set contains 817 images with 2017 people, of which 1779 are small objects (area < 32^2 pixels) and 238 medium objects (area between 32^2 and 96^2 pixels), while there are no large objects (area > 96^2 pixels).

4.1.2 SAR-DAG_overflight - datasets for evaluating geolocation method

To test the geolocation algorithm, we created a set of images taken at two locations, a meadow, and a vineyard. The images were captured by a Phantom 4 Advance drone, equipped with a camera with a field of view of 84° that flew at a height of 30 meters and captured images at regular time intervals as is usual in

SAR missions. The images have a resolution of 5472 x 3648 pixels, and an example is given in Figure 4. The set contains 40 marked persons. From the metadata of the images taken at the position where the drone took off and at the position when the drone is vertically above the person, GPS position data is taken to obtain the starting point and the actual position of the person on the ground.

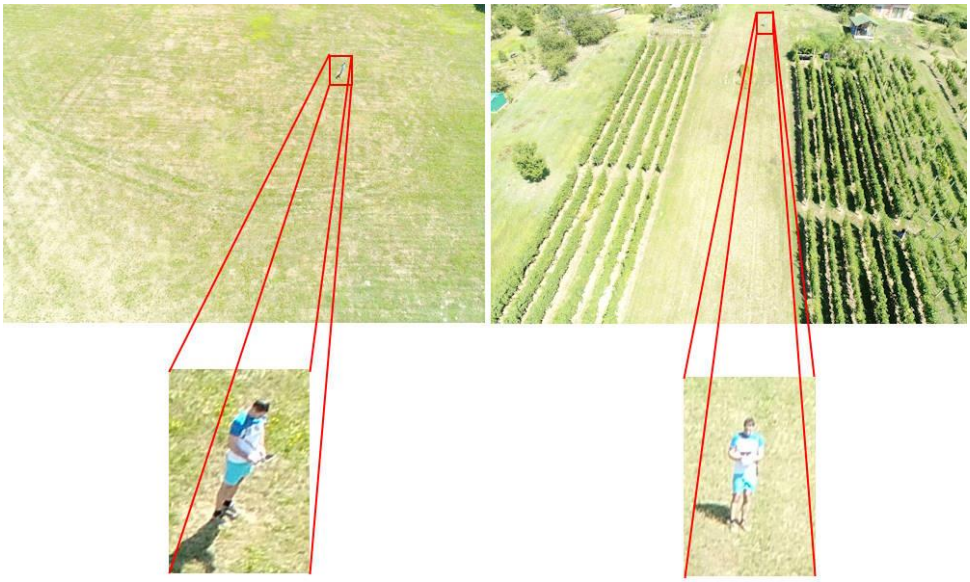


Figure 4: Examples of SAR-DAG_overflight images with zooming in on a part of the image where the person is.

4.2 Evaluation Metric

In the experiment, we use several standard metrics to evaluate detector performance and metrics that we have purpose-developed for detection and geolocation in SAR missions as explained below.

Intersection over Union (IoU) is a traditional metric for evaluating the performance of an object detector calculated as the ratio of the intersection and union of the detected bounding box and the ground true bounding box. The equation is as follows:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{5}$$

Higher IoU values indicate better overlap between detection and the real data.

Recall (R) and Precision (P) are calculated as:

$$P = \frac{TP}{TP + FP}, \qquad R = \frac{TP}{TP + FN} \tag{6}$$

where TP is positive detection that are true, FP is false positives, and FN is false negative detection.

Mean average precision (mAP) is a common evaluation metric in object detection. In the experiment, we use mAP 50, the average precision at IoU greater than or equal to 0.5 and mAP 50-95 the average precision in the range of IoU from 0.5 to 0.95, with intervals of 0.05.

For SAR operations, it is important that the detector is optimized to have as few false positive (FP) detections as possible, because they consume human resources and time. Therefore, the performance of the detector is also evaluated using the ROpti (Recall Optimal) metric, which penalizes false positive detections (Sambolek & Ivasic-Kos, 2021). ROpti is calculated as the ratio of the difference between true positive (TP) and false positive (FP) detections and the total number of detections (TP+FN):

$$ROpti = \frac{TP - FP}{TP + FN} \tag{7}$$

The experiments also evaluate the accuracy of geolocating a person using the proposed algorithm (Section 3.2).

4.3 Experimental Results

4.3.1 YOLOv8 person detection

We conducted the experiments using all five versions of the YOLOv8 models modified to detect a person class and implemented in PyTorch using Python version 3.9.16.

First, on the SARD validation set we tested original YOLOv8 models trained on the COCO dataset, and the obtained results are shown in Table 2. The confidence threshold was set to 0.25 and the IoU threshold to 0.5.

The YOLOv8x model achieved the best result of all YOLOv8 versions on the SARD validation set, namely mAP@0,5 of 74.6%, recall of 49.2%, and mAP@ 0.5:0.95 of 35%, which is significantly worse than when tested on the COCO set.

Although it is a simplified detection task with only one class (person), all YOLOv8 models show the same performance degradation with many false detections (low ROpti). Considering that the SARD set was recorded from a completely different perspective (bird's eye view) and with many small objects for which the models were not trained, it was necessary to fine-tune them to SARD datasets so that they can be used in SAR missions.

We trained all version of YOLOv8 models for 500 epochs using Tesla T4 GPUs on the Google Collaboratory platform while the hyperparameters remained unchanged. We used SGD optimizer, and the weight decay set to 5×10^{-4}, while the initial learning rate was set to 10^{-3}. Input image size was 640 and batch size set to 16.

Detection performances on SARD validation dataset were evaluated using standard metrics of Precision, Recall, mAP@0.5, and mAP@0.5:0.95, and customized ROpti measure (Sambolek & Ivasic-Kos, 2021). After fine-tuning on the SARD data set all models show a significant improvement in detection (Table 2.). The best results were achieved by YOLOv8x with mAP@0.5 91.3% and mAP@0.5:0.95 68.8%, which makes it the most suitable for offline analysis of materials recorded during drone flight because the accuracy is in that case the most important.

The YOLOv8n model has the significantly fastest detection of only 4.6 ms per image and achieves mAP@0.5 only 4.5% lower than the best results. The same is true for the YOLOv8s model, which achieves the second-best inference time with almost the same mAP@0.5 performance as YOLOv8x. This makes it most suitable for use during a SAR operation when, in addition to detection accuracy, it is important for the model to inference quickly, in real time, and to be used on a drone without the need for large computing resources.

4.3.2 Person Geolocation

We have conducted a comparison of existing geolocation methods using a simplified ellipsoidal model of the Earth, an algorithm using DEM (Digital Elevation Model) and an intersection measurement algorithm. The results of the

first two measurements were taken from the paper (Sambolek & Ivašić-Kos, n.d.). Table 3 shows the results of the distance estimation between the calculated GPS location of a person using the mentioned three algorithms and the exact GPS location where the person was located. The algorithms were tested on five different data sets, two of which were recorded in a meadow (flat terrain), while three were recorded in a vineyard (sloping terrain). In data sets recorded in the meadow, no major deviation was observed for intersection algorithms that consider changes in the terrain configuration (e.g., a mean error of 4.5 m for PhantomLP1), however, on terrains with different slopes, the intersection measurement algorithm shows significantly better results than other algorithms.

The best result was achieved in the first set recorded in the vineyard (PhantomVP1), with an average error of 4.8 meters. In the case of the Earth ellipsoid model and the DEM model, accuracy was checked for each image in the dataset.

If a person is detected in one image or is in motion during the search, it is recommended to use the DEM model to determine the geolocation. When detecting a stationary person in multiple images, it is suggested to use the intersection measurement algorithm, which achieves the best results.

Table 2. Performance of five versions of the YOLOv8 model on the SARD test dataset. The first five rows correspond to models trained on the COCO dataset and the last five to models that are fine-tuned on the SARD dataset, with the best results highlighted in bold.

Version of YOLO and training dataset	**Precision (%)**	**Recall (%)**	**mAP @0,5 (%)**	**mAP @ 0.5:0.95 (%)**	**ROpti**	**Speed per image [ms]**
YOLOv8n @COCO	61	26	35.9	16.5	0.09	**4,8**
YOLOv8s@COCO	66	37	47.5	23.8	0.18	8,5
YOLOv8m@COCO	74	46	59.6	32	0.29	17.5
YOLOv8l@COCO	**75**	47	60.7	34.5	0.31	34.5
YOLOv8x@COCO	**75**	**49**	**62.0**	**35.3**	**0.32**	46.6
YOLOv8n@SARD	93	78	86.8	54.9	0.71	**4.6**
YOLOv8s @SARD	94	81	90.3	60.6	0.76	8.0
YOLOv8m@SARD	93	83	90.6	62.1	0.77	17.3
YOLOv8l@SARD	94	83	90.8	60.8	0.78	34.4

YOLOv8x@SARD	**95**	**83**	**91.3**	**63.8**	**0.79**	46.5

Table 3. Coordinates calculation of person standing on a known location.

Dataset	No. of Images	Earth ellipsoid model (Sambolek & Ivašić-Kos, n.d.)			DEM (Sambolek & Ivašić-Kos, n.d.)			Intersection measurement algorithm		
		MeanError	MaxError	MinError	MeanError	MaxError	MinError	MeanError	MaxError	MinError
PhantomLP 1	10	8.963	10.539	7.87				13.446	14.377	12.713
PhantomLP 2	10	8.704	11.595	6.212				8.439	8.832	7.592
PhantomVP 1	4	18.374	29.262	8.412	10.935	15.833	5.630	**4.794**	5.451	4.004
PhantomVP 2	7	50.488	73.028	14.427	23.604	34.681	7.327	10.534	11.139	10.351
PhantomVP 3	9	51.312	98.203	22.815	29.911	66.887	14.762	12.388	14.465	9.725

5. Conclusions

In this paper, we have demonstrated that the YOLOv8 models can be successfully fine-tuned on UAV images for person detection in real-world environments. Our experiment was conducted on the publicly available SARD dataset.

Furthermore, we built a set of SAR-DAG_overflight for testing the geolocation of a person and tested three geolocation algorithms on it: the Earth's ellipsoid model, the DEM model, and the modified cross-section measurement algorithm that we proposed in the paper.

We believe that the fine-tuned YOLOv8@SARD models that we fine-tuned at the SARD dataset and the proposed person geolocation algorithms along with the given recommendations can be greatly utilized in SAR operations as they can help in the detection of persons in drone images, and thus contribute to providing more precise information for coordinating the operation and reducing search time.

In future work, we plan to further investigate the model's robustness to weather conditions, night shooting, and camera motion blur, as well as conduct experiments with multiple datasets to increase the robustness and generalizability of our model.

References

Andriluka, M., Schnitzspan, P., Meyer, J., Kohlbrecher, S., Petersen, K., Von Stryk, O., Roth, S., & Schiele, B. (2010). Vision based victim detection from unmanned aerial vehicles. *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*. https://doi.org/10.1109/IROS.2010.5649223

Bai, G., Liu, J., Song, Y., & Zuo, Y. (2017). Two-UAV intersection localization system based on the airborne optoelectronic platform. *Sensors (Switzerland)*, *17*(1). https://doi.org/10.3390/s17010098

Bejiga, M. B., Zeggada, A., Nouffidj, A., & Melgani, F. (2017). A convolutional neural network approach for assisting avalanche search and rescue operations with UAV imagery. *Remote Sensing*, *9*(2). https://doi.org/10.3390/rs9020100

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *Yolov4: Optimal speed and accuracy of object detection*.

Chien-Yao Wang, Alexey Bochkovskiy, H.-Y. M. L. (2023). YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7464–7475.

Doherty, P., & Rudol, P. (2007). A UAV search and rescue scenario with human body detection and geolocalization. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4830 LNAI*. https://doi.org/10.1007/978-3-540-76928-6_1

El Habchi, A., Moumen, Y., Zerrouk, I., Khiati, W., Berrich, J., & Bouchentouf, T. (2020). CGA: A New Approach to Estimate the Geolocation of a Ground Target from Drone Aerial Imagery. In *4th International Conference on Intelligent Computing in Data Sciences, ICDS 2020*. https://doi.org/10.1109/ICDS50568.2020.9268749

Geraldes, R., Goncalves, A., Lai, T., Villerabel, M., Deng, W., Salta, A., Nakayama, K., Matsuo, Y., & Prendinger, H. (2019). UAV-based situational awareness system using deep learning. *IEEE Access*, *7*. https://doi.org/10.1109/ACCESS.2019.2938249

Huang, C., Zhang, H., & Zhao, J. (2020). High-efficiency determination of coastline by combination of tidal level and coastal zone DEM from UAV tilt photogrammetry. *Remote Sensing*, *12*(14). https://doi.org/10.3390/rs12142189

Leira, F. S., Trnka, K., Fossen, T. I., & Johansen, T. A. (2015). A ligth-weight thermal camera payload with georeferencing capabilities for small fixed-wing UAVs. *2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015*. https://doi.org/10.1109/ICUAS.2015.7152327

Li Chuyi, Li Lulu, Jiang Hongliang, Weng Kaiheng, Geng Yifei, Li Liang, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, Yiduo Li, Bo Zhang, Yufei Liang, Linyuan Zhou, Xiaoming Xu, Xiangxiang Chu, Xiaoming Wei, X. W. (2022). *YOLOv6: A single-stage object detection framework for industrial applications.*

Paulin, G., Sambolek, S., & Ivasic-Kos, M. (2024). Application of raycast method for person geolocalization and distance determination using UAV images in Real-World land search and rescue scenarios. *Expert Systems with Applications*, *237*. https://doi.org/https://doi.org/10.1016/j.eswa.2023.121495

Qu, Y., Wu, J., & Zhang, Y. (2013). Cooperative localization based on the azimuth angles among multiple UAVs. *2013 International Conference on Unmanned Aircraft Systems, ICUAS 2013 - Conference Proceedings*. https://doi.org/10.1109/ICUAS.2013.6564765

RangeKing. (n.d.). *YOLO v8 architecture*. https://github.com/ultralytics/ultralytics/issues/189

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer*

Society Conference on Computer Vision and Pattern Recognition, *2016-December*. https://doi.org/10.1109/CVPR.2016.91

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *Tech Report*.

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-January*. https://doi.org/10.1109/CVPR.2017.690

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6). https://doi.org/10.1109/TPAMI.2016.2577031

Sambolek, S., & Ivasic-Kos, M. (2021). Automatic person detection in search and rescue operations using deep CNN detectors. *IEEE Access*, *9*, 37905–37922. https://doi.org/10.1109/ACCESS.2021.3063681

Sambolek, S., & Ivašić-Kos, M. (n.d.). *Determining the Geolocation of a Person Detected in an Image Taken with a Drone*.

Shakhatreh, H., Sawalmeh, A. H., Al-Fuqaha, A., Dou, Z., Almaita, E., Khalil, I., Othman, N. S., Khreishah, A., & Guizani, M. (2019). Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. In *IEEE Access* (Vol. 7). https://doi.org/10.1109/ACCESS.2019.2909530

Sun, J., Li, B., Jiang, Y., & Wen, C. Y. (2016). A camera-based target detection and positioning UAV system for search and rescue (SAR) purposes. *Sensors (Switzerland)*, *16*(11). https://doi.org/10.3390/s16111778

Ultralytics. (n.d.-a). *Yolov5 GitHub*. Retrieved September 15, 2023, from https://github.com/ultralytics/yolov5

Ultralytics. (n.d.-b). *YOLOv8 Doc*. https://docs.ultralytics.com/tasks/detect/

Ultralytics. (n.d.-c). *YOLOv8 GitHub*. Retrieved September 15, 2023, from https://github.com/ultralytics/ultralytics

Wang, X., Liu, J., & Zhou, Q. (2017). Real-time multi-target localization from unmanned aerial vehicles. *Sensors (Switzerland)*, *17*(1). https://doi.org/10.3390/s17010033

Xu, C., Yin, C., Han, W., & Wang, D. (2020). Two-UAV trajectory planning for cooperative target locating based on airborne visual tracking platform. *Electronics Letters*, *56*(6). https://doi.org/10.1049/el.2019.3577

Zhao, X., Pu, F., Wang, Z., Chen, H., & Xu, Z. (2019). Detection, tracking, and geolocation of moving vehicle from UAV using monocular camera. *IEEE Access*, *7*. https://doi.org/10.1109/ACCESS.2019.2929760

Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2023). Object Detection in 20 Years: A Survey. *Proceedings of the IEEE*, *111*(3). https://doi.org/10.1109/JPROC.2023.3238524

RAD 7. DETERMINING THE GEOLOCATION OF A PERSON DETECTED IN AN IMAGE TAKEN WITH A DRONE

Sambolek, Sasa and Ivasic-Kos, Marina, Determining the Geolocation of a Person Detected in an Image Taken with a Drone. Available at SSRN: https://ssrn.com/abstract=4373987 or http://dx.doi.org/10.2139/ssrn.4373987

Radi jasnoće, rad je preoblikovan, inače je sadržaj isti kao i verzija poslana na recenziju 1. ožujka 2023. godine.

https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4373987

1. Introduction

A few years ago, there was a rapid increase in the use of uncrewed aerial vehicles (UAVs, drones) in various applications. This includes search and rescue (SAR) operations and searching for missing people in non-urban areas and hard-to-reach terrain, natural disaster management, detecting abnormal human behavior, crowd management during the evacuation, and many other areas where people's location information is important.

A drone has many components, including electronic speed and flight controllers, a battery, a navigation system including a GPS module, accelerometer, gyroscope, altimeter and various onboard sensors, including ultrasonic, laser or lidar distance sensors, collision avoidance sensors, time of flight sensors, stabilization sensors and orientation. Drones are usually equipped with cameras with standard or infrared visual sensors that allow capturing images or videos from a bird's eye view. During the flight, drones record a large number of metadata related to the trajectory of the drone, camera parameters at the time of taking the photo, and the like with each image taken.

Controlling the drone, i.e., its launch, navigation, and landing, is often done manually using a remote pilot. Still, trends are moving toward increasing the automation of some flight operations and using drones supported by artificial intelligence. AI-based drones rely heavily on computer vision methods and models such as deep convolutional neural networks (CNNs) or recurrent neural networks (RNNs) that allow the analysis of image/video data captured by the drone during flight and concluding detection, identification, and tracking of objects.

Today, there are already many models of CNN architectures that are more and more precise in detecting people in images and faster in performing and reaching conclusions. However, to be effective and provide usable results on drone footage, they should be additionally trained on the data collected by the drone due to changing recording conditions, image distortion due to UAV movement, identification of a small target, computationally demanding implementation of algorithms, etc. [1]. Furthermore, data preparation and model learning has

become extremely demanding due to the specificity of the recording conditions and the need to collect specific rare data in everyday scenes, such as an injured person on a mountain. Therefore, detection models should be tuned for a specific task and application. A self-sustaining real-time person detection system can be particularly useful for these applications. Time is an essential resource, and early identification of people and knowledge of their distribution is important.

This paper will focus on using drones in search and rescue operations. Search and rescue rely heavily on situational awareness. If a person is missing, certain steps must be taken according to the context and probabilities associated with the person's status and behavior. Field [2] lists the three main tasks: investigation, containment, and hasty search. All these tasks must be done as quickly as possible to prevent increasing the risk to the missing person's safety. During investigative actions, useful information about the subject is collected, including his activity plans before he went missing. This will help investigators understand where the person is most likely to be. Containment aims to prevent the search area from expanding. The more time passed before the missing person was found, the larger the search area. Using drones can help scan the search area faster and find the missing person faster. It also reduces the risk to the safety of the field search team.

A larger search area is scanned by flying at higher altitudes, but the number of pixels occupied by a person on the image/screen is reduced. For this reason, it is possible that the requested person will not be detected during the flight of the drone by the remote pilot and software support for real-time person detection. That is why it is advisable to repeat the search/detection of persons on the recorded material afterward, offline, in the command center, with the help of a person detection algorithm that can use a higher-powered computer, because its goal is the precision of detection and not the speed of execution.

This work focuses on systems used when a person is not detected in real-time during a drone flight. First, the paper describes the process of developing a system for detecting persons in search and rescue cases, which, along with the detection of a person, determines the person's GPS location and the direction

and speed of movement if the person is detected in several images. Finally, based on the obtained data, the system proposes correcting the search area. We also present a more practical and physical approach rather than a mathematical approach to reduce estimation errors.

The main contributions of this paper are:

- algorithm for geolocating the detected person using Digital Elevation Model (DEM);
- algorithm for determining the speed and direction of movement of the detected person and correction of the search area;
- improved YOLO model (using SARD-832-1024 dataset) for person detection with better ROpti;
- prototype AI-SAR application for searching for missing persons in non-urban terrain.

The remainder of the paper is organized as follows: Section 2. provides an overview of the area of CNN-based object detection and geolocation in aerial imagery. In Section 3, the system for detection and geolocation of a person in a image taken by a drone is presented. Section 4. describes the method for geolocalization of a person and determining the speed of a person and the new search area, and gives a description of the experiment. Section 5. analyzes the obtained experimental results. The paper ends with a conclusion and guidelines for future research.

2. Related works

2.1. Object detection

Object detection methods based on deep learning have greatly progressed in recent years. Two-stage methods and one-stage methods are two of their branches. As for the two-stage R-CNN [3] and Fast R-CNN [4], algorithms divide detection into region proposal generation and classification. They focus on improving detection accuracy while sacrificing detection speed. For single-stage

algorithms, generating region proposals is eliminated, and the probability and position coordinates are obtained directly through a single network. The one-stage detector creates bounding box candidates at a given position and scale and then calculates their actual bounding box and score for each class like the single-shot multibox detector (SSD) [5]. These algorithms perform well in terms of speed. However, most perform worse than two-stage algorithms in detection accuracy and small object detection. YOLO [6] and its improved versions, denoted as YOLOv2 [7], YOLOv3 [8], YOLOv4 [9], are typical methods in single-phase algorithms. YOLOv4 performs better speed and accuracy and works especially well in detecting small objects. In the field of our research, searching for missing persons, the speed of execution, as well as the detection of small objects, is of great importance. In earlier experimental work, we compared the detector's performance, chose YOLOv4, and additionally trained it (fine tuning) on our data set to improve its performance of person detection in aerial search and rescue scenes [1].

2.2. Object geolocation

Most computer vision techniques rely on a camera model and calibration. A camera model is a geometric approximation of how light travels through a camera lens and forms images. Camera calibration is required to correct major deviations due to the model used. In addition, camera calibration can relate camera pixel measurements to the real three-dimensional world [10].

Pinhole model

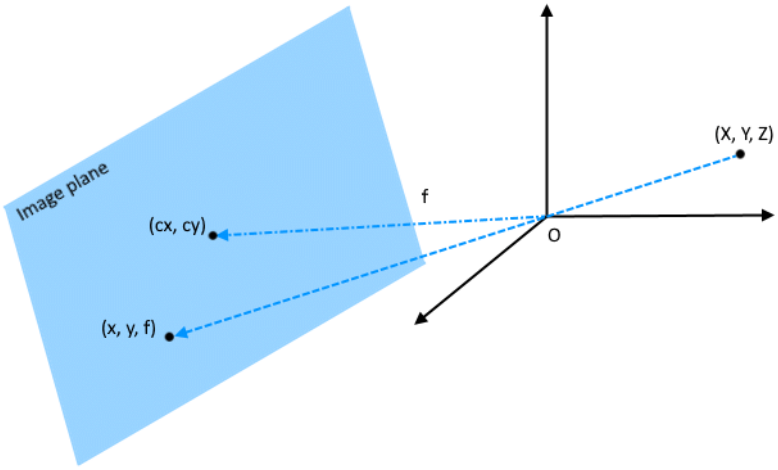


Figure 1: The three dimensional point (X, Y, Z) is projected through the pinhole to the two dimensional plane (reproduced from reference [10]).

In Fig. 1 shows the principle on which the pinhole model is based. As can be seen, each point P with coordinates (X, Y, Z) in 3D space is projected through a pinhole (which is taken as the origin of the coordinate system) to a point P' with coordinates (x, y, f) in the camera plane. It follows from the similarity of the triangles:

$$\frac{x'}{X} = \frac{y'}{Y} = \frac{f}{Z} = \lambda \qquad (1)$$

Where λ is the ratio factor. If the focal length and are known, it is possible to calculate the 3D coordinates of the point from the 2D coordinates projected onto the image plane and the focal length. Usually, the focal length along with the intrinsic and extrinsic parameters can be obtained from the camera calibration procedure.

Camera Calibration

The basic pinhole model does not include distortion, usually in real cameras. Camera calibration provides a model of camera geometry and lens-induced distortions. This information can be used to define internal and external camera parameters.

Let P' be the projected point of the 3D point in the camera plane (as shown in Fig. 1). Then, using homogeneous coordinates, we define P = [X Y Z 1]^T and P' = [x, y, 1]^T. We can then express the mapping from P to P' in matrix multiplication.

$$\lambda P' = A[R\ t]P \tag{2}$$

here P is the 3D object point in homogeneous coordinates; P' is the same point of the object in homogeneous 2D coordinates; [R t] is the matrix of extrinsic parameters (rotation and translation); λ is an arbitrary scale factor and A is the matrix of intrinsic parameters. The matrix of intrinsic parameters is:

$$A = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Here, fx, and fy provide information (depending on the pixel size) on the focal distance in the x and y direction, respectively; cx and cy are the coordinates of the main point of the image; s is known as skew and represents the angle of inclination of the pixel.

The object's position, relative to the camera coordinate system, could be described in terms of the rotation matrix R and the translation vector t. The rotation matrices Rx, Ry, and Rz, respected can represent rotation around the x, y, and z axes:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix},$$

$$R_y = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos(\beta) \end{bmatrix}, \tag{4}$$

$$R_z = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Here, α, β and θ are the angles of rotation around the x, y, and z axes, respectively. Specifically, α, β and θ are the pitch, roll and yaw angles of the

camera. Finally, the rotation matrix R can be constructed by multiplying the three rotation matrices.

In our work, we do not rely on camera calibration because several different drones participate in search and rescue operations, where the camera calibration of each drone would slow down the search.

Triangulation

Triangulation is a method used to estimate the distance to an object, which is most often achieved with the help of a stereo camera. Stereo cameras are two cameras at a constant distance that record the same scene; the working principle is shown in Fig. 2. A point in the scene is mapped in different places in the images of the two cameras, depending on the distance of that point from the stereo pair. If we want to calculate the depth of the scene mapped into the pixel p_L on the left image, we only need to find the pixel p_R on the right image into which the same part of the scene has been mapped.

Some authors using a monocular camera use this method by taking two images at two locations. Stereovision does not apply to our problem since the focus is on widely available drones. Therefore, in this paper, we consider only the case of localization based on a monocular vision for one image using mathematical methods from the data recorded in the image's metadata.

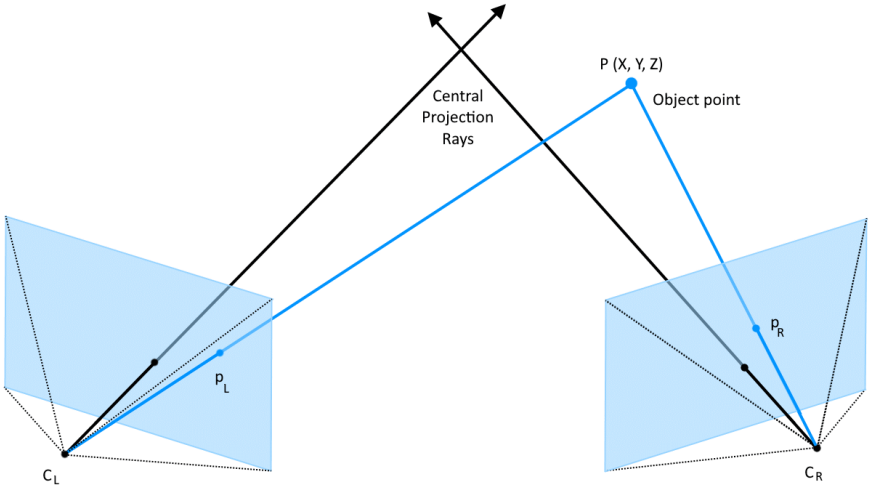


Figure 2: Working principles of the stereo camera.

Geolocation based on drones is divided into active and passive methods according to the mechanism of operation [11]. Active UAV methods for object localization are based on a laser range finder. For example, the DJI M30 drone [12] has a laser rangefinder that can provide precise coordinates of objects up to 1200 meters away. However, these devices are not available on small aerial platforms and are not useful when object detection is performed offline on captured images [13]. For small drones, GPS (or GLONASS) and IMU (Inertial Measurement Unit) can provide the location and position of the drone, so passive methods are widely chosen. An Inertial Measurement Unit (IMU) is an electronic device that uses accelerometers and gyroscopes to measure acceleration and rotation, which can be used to provide position data. The raw measurements output by an IMU (angular rates, linear accelerations, and magnetic field strengths) or AHRS (roll, pitch, and yaw) can be fed into devices such as Inertial Navigation Systems (INS), which calculate relative position, and orientation and velocity to aid navigation and control of drones. Onboard processors continuously calculate the drone's current position. First, it integrates the sensed acceleration with an estimate of gravity to calculate the current velocity. Then it integrates the velocity to calculate the current position. To fly in any direction, the flight controller gathers the IMU data on current positioning, then sends new data to the motor electronic speed controllers (ESC). These electronic speed controllers signal to the motors the level of thrust and speed required for the drone to fly or hover.

The GPS position of the object is calculated using image analysis methods. Mathematical methods of transforming a point on the image into Earth geographic coordinates are often used [14][15]. The basic information required for such a calculation is the camera angles, the height at which the drone is located, the geolocation of the drone, and the pixel coordinates of the point that we want to transform into the Earth's coordinate system. The transformation of the camera frame to the ENU (East-North-Up) frame is presented as:

$$P_t^{(E)} = R_{(C)}^{(E)} P_t^{(C)} + P_{uav}^{(E)} \quad (5)$$

Where $R_{(C)}^{(E)}$ is the rotation matrix from the ENU frame to the camera frame, which contains the rotation matrix around the x, y, and z axes taking into account the camera angles (pitch, roll, yaw) Equation 5. $P_{uav}^{(E)}$ is the position of the UAV in the ENU frame and represents the position of the target in the camera frame. These transformations assume that the terrain on which the vehicles drive is relatively flat.

Geolocating that in the calculation also uses the diagonal field of view (FOV) of the camera is given in [16] only for cases of vertical aerial photography. Also, for vertical aerial photos, the authors in [17] propose a method that combines landmarks on the video that match the detections of the deep learning network (YOLOv3) on the reference image, thereby geolocating targets in cases where GPS signals are not available. In another paper, [18] uses a CNN detector for the case when the GPS signal is unavailable. It changes the pixel perspective from the drone's perspective to an orthogonal view of the detected object, and four reference points determine the geolocation of the detected object. Known data, such as coordinates of reference points or information about the size of reference objects (e.g., cars, buildings, plots) are used for detection.

In [19], the authors propose a new approach based on position encoding-decoding and use the SRTM DEM (The Shuttle Radar Topography Mission Digital Elevation Model) to model the ground terrain. The model encodes the world positions of each terrain point with a unique color and later decodes it from the terrain mesh to recover the world position. After detecting an object on the

image with OpenGL readPixel functionality, it reads the current vertex buffer and retrieves the color at a given 2D position on the camera window. So, the color can be encoded using only each point's x and z coordinates. The y coordinate (elevation) can be later retrieved knowing the x and z coordinates (latitude and longitude). Fig. 3 shows color encoding-decoding and the top view of the whole SRTM tile.

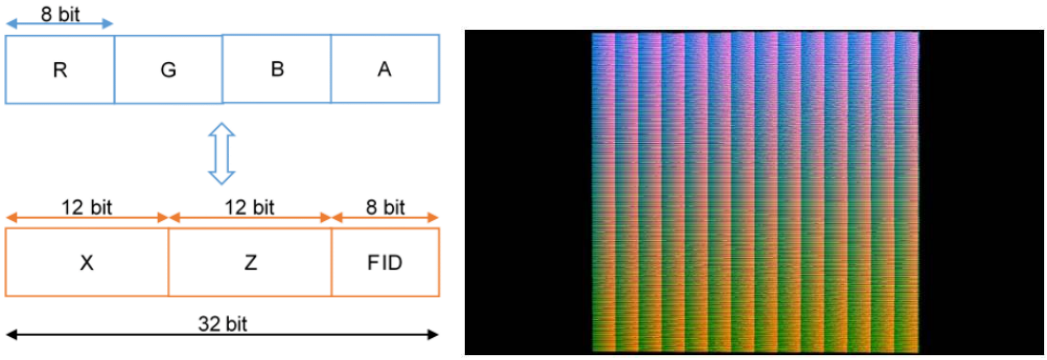


Figure 3: a) Color encoding-decoding, b) top view of the whole SRTM tile

A Digital Elevation Model (DEM) consists of a database that contains the elevation above sea level of a specific location. Despite the existence of different formats, it most often divides the earth's surface into squares of regular size. The highest altitude is saved for each square. The header of the DEM file lists the GPS coordinates of the origin as well as the size of each square (also called resolution). Since it is easy to retrieve the row and column number when reading a DEM file, the GPS location of each cell can be easily calculated (i.e., by adding the offset from the origin). Digital Elevation Models can be produced by various techniques, such as digitization of contours from existing topographic maps, topographic leveling, EDM (Electronic Distance Measurement), differential GPS measurements, (digital) photogrammetry, radar remote sensing (InSAR), and Light Detection and Ranging (LiDAR). Today, a wide range of data sources can be chosen for generating DEMs, and it is especially important for research that DEM files are freely available on the Internet. For our research, we used the Digital Elevation Model over Europe [20] from the GMES RDA project (EU-DEM), which is a Digital Surface Model (DSM) representing the first surface as illuminated by the sensors. The EU-DEM dataset is a realization of the

Copernicus program, managed by the European Commission, DG Enterprise, and Industry. The difference between DSM and DTM is shown in Fig. 4 and refers to the DSM including buildings, trees, and other fixed objects on the ground's surface, while the DTM model shows the terrain itself. DSMs more accurately represent the real world (especially in cities), but they also need to be updated much faster. Since the file formats of DTM and DSM files are very similar, our approach works for both DTM and DSM files, as it uses them as an additional parameter when determining the direction of movement of a detected person in non-urban terrain.

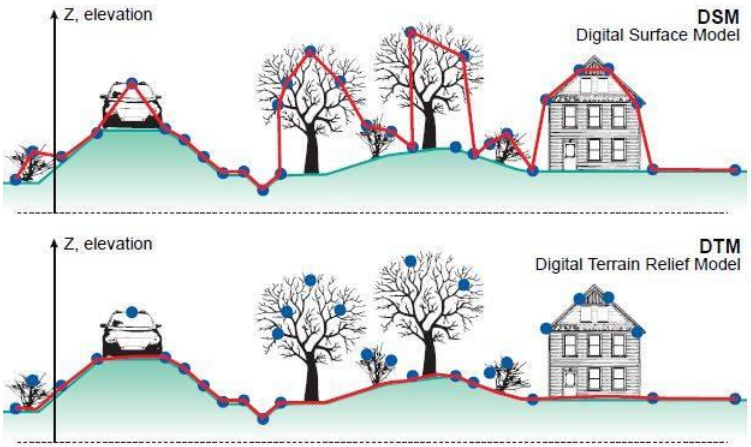


Figure 4: The difference between DTM and DSM; source: [21]

3. System for automatic detection and geolocation of the person in the picture

This paper proposes a system for automatically detecting and geolocating people in photographs taken by a drone camera in non-urban areas during search and rescue operations. If a person is detected in several photos, the system determines the speed of the person's movement and suggests a search area based on that. The obtained information is recorded in a file (.gpx) that can be displayed visually in GIS programs.

The system for detecting and determining geolocation consists of several modules working together. Figure 5 illustrates the processes that make up the system.

During the drone's flight in a search and rescue operation, the drone captures the monitored area and stores metadata with each image. It is recommended that in a search and rescue operation, the drone has settings for automatic imaging of the area every 2 seconds so that a set of data can be formed on which an additional offline search can be made in the command center. After the flight, the recorded images are analyzed on a more powerful computer with detection algorithms, and a missing person is searched for. This research uses a person detection model based on YOLOv4 architecture, and fine-tuning is done for search and rescue scenes (YOLOV4-SARD-832-1024) [1].

To estimate the geolocation of the detected person, metadata recorded with the image and the height of the ground at the place where the drone took off (home point) are used. Each image in which a person is automatically detected is marked and saved with all metadata in a set of images for verification by mission operators. If a person is detected in two or more photos, the system estimates the speed of the person's movement in the observed interval and accordingly determines a new search area.

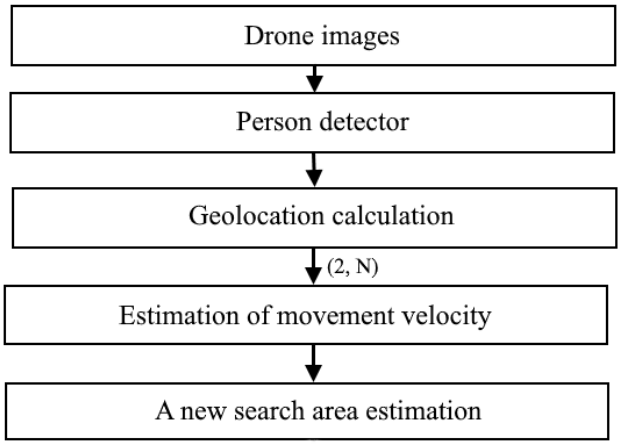


Figure 5: System for automatic detection and geo-localization of the missing person.

3.1. Person detection

In earlier research, we investigated and analyzed existing detectors based on deep convolutional neural networks and showed that the starting model YOLOv4 [9] pre-trained on the COCO base [22] gives the best results for our needs [1]. YOLOv4 uses CSPDarkNet53 [23] as the backbone. To the basic DarkNet53, a

deep residual network with 53 layers, CSPNet (Cross Stage Partial Network) was added. Also, the authors of YOLOv4 added Spatial Pyramid Pooling (SPP) [24] as a neck to increase the receiving (receptive) field without causing a decrease in inference speed performance. YOLO divides the image into a grid of dimensions S×S, each cell providing frames for the object. The probability, calculated for each frame, tells how confident the model is when there is an object inside the frame and how confident it is in the accuracy of the bounding box.

The model is fine-tuned on the custom dataset SARD [1] which contains and simulates scenes from search and rescue missions in non-urban areas. A DJI Phantom 4A drone took the SARD images in FHD resolution. Several examples of images from the SARD dataset can be seen in Fig. 6. There are 1,981 images in this set, with 6,532 marked people. The set was divided in a ratio of 60:40 into train and test datasets.

Figure 6: Examples of images from the SARD dataset.

Several YOLOv4 models were trained on the SARD dataset, each with a different network resolution, starting from the original 512×512 [1], to higher of 832-832 and 1024x1024. SARD-832-1024 model trained on a network resolution of 832x832 and tested on with a network resolution of 1024x1024 was chosen as the best because it achieves a high rate of person detection with sufficient robustness and a speed of inference that meets the needs of SAR missions (AP 65%, APs 51.2%, ROpti 93.9 %).

For comparison purposes in, Table 1 are shown the results achieved after training and testing the model on different network resolutions. For example, SARD-512-832 is a model trained on a network resolution of 512x512 and tested with network resolution 832x832, SARD-832-832 trained and tested on a network resolution of 832x832 and SARD-832-1024 is a selected model for a further experiment that was trained on network resolution of 832x832 and tested on with network resolution of 1024x1024.

Table 1. Detection results for YOLOv4 model (%).

Train	Test	AP	IMP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_S	AR_M	AR_L	ROPTI
SARD-512-832 [1]	SARD	61.3	37.9	95.7	71.1	45.0	66.4	72.6	52.3	72.1	77.8	92.8
SARD-832-832	SARD	63.4	40.0	95.8	73.9	46.5	68.3	**78.1**	52.9	73.8	**82.4**	92.4
SARD-832-1024	SARD	**65.0**	**41.6**	**95.9**	**77.1**	**51.2**	**69.1**	76.9	**57.7**	**74.7**	82.1	**93.9**

4. The proposed method of determining the geolocation and speed of movement of a person automatically detected in the image

The assumption for determining the geolocation of a person is that a drone takes images and that there is a person detection model that has detected people in the recorded images. For the detection of persons, we used the model we defined during the previous research, however, it is important to point out that the proposed method of geolocation and determining the speed of movement of a missing person does not depend on a specific detector, and that any detector that gives good results on tracking recordings can be used to detect a person and rescues filmed by a drone.

The input data for the proposed method of offline geolocation of a person in an image taken during a drone flight is metadata that is stored with each captured image and the height of the ground at the place where the drone took off (home point).

From many metadata recorded during the drone's flight for localization of the detected person, we used their subset, which consists of data related to the drone's trajectory, image identification, and camera parameters at the time of

taking the photo. The data used and a specific example of values are shown in Table 2.

Table 2. Used data and their measurement units.

Variable	Description	Example data	Unit
Time	Flight point's timestamp	2022-09-26 19:35:42	
File_Name	Filename of the image recorded in the flight point	DJI_0265.JPG	
Img_Width	Recorded image's width	5472	px
Img_Height	Recorded image's height	3648	px
FOV	Camera's diagonal field of view	84	degrees
Relative_Altitude	Drone's altitude relative to the take-off point's height	30.1	m
Gimbal_Pitch_Degree	Camera's pitch	-45.8	degrees
Gimbal_Yaw_Degree	Camera's yaw	15	degrees
Gimbal_Roll_Degree	Camera's roll	0	degrees
GPS_N	Latitude of the flight point	45.5107911388	degrees
GPS_E	Longitude of the flight point	16.7602712222	degrees

In order to simplify the problem of detecting/tracking a person in images, the dataset in the analyzed case was formed so that there is only one person in the images that represent the target for detection.

4.1. Geolocation algorithm

This section describes the procedure for estimating the distance between the detected person and the camera mounted on the drone, i.e., determining the position of the person in the coordinate system of the earth in the image taken from a bird's eye view. The method takes as input one RGB image captured by the drone during the flight and a bounding box obtained because of person detection with a corresponding confidence value. The output is the GPS (WGS84) coordinates of the detected person in the image.

The detection bounding box is used to estimate the relative position of the person, so the middle of the lower edge of the bounding box is taken as the reference point for determining the person's distance. By the same principle, the distance of other detected objects could be calculated.

Algorithm 1 Post-flight detection

Input: A set of images taken by a drone in a non-urban area

Output: Detected persons marked with a bounding box and data on the geolocation of the detected person

1: Take a set of drone shots

2: Search for the desired object using a convolutional neural network

3: If the object is detected, determine the center of the bottom edge of the bbox as the point where the object is located

4: Calculate the position of the object in the 3D coordinate system of the Earth

5: If the same object appears in more than one image in the set, determine the speed and direction in which the object is moving

5.1.: From the obtained data, propose a new search area

6: Save the calculated object positions as a .gpx file

The proposed algorithm for determining the distance of the detected object from the camera uses camera parameters and data obtained from sensors installed in the drone (metadata). The retrieved telemetry consists of the drone's GPS position, height relative to the ground takeoff position, gimbal_roll, gimbal_pitch and gimbal_yaw angles. If gimbal_yaw = 0° and the camera is looking to the ground (i.e. nadir) it means that the top of the image points to the north, for gimbal_yaw = 90° and camera is looking nadir, it means that the top of the image points to the east and if gimbal_yaw = 270° and camera is looking nadir, it means that the top of the image points to the west. If gimbal_pitch = 0°, it means that the camera is looking forward or if gimbal_pitch = -90°, it means that the camera is looking down (i.e. nadir). The relationships between camera positions and image orientation are shown in Table 3.

Table 3. Relationships between camera position and orientation in the Earth system.

Gimbal YAW	Gimbal PITCH /Camera position	Top of the image orientation
0°	-90° (nadir)	north
90°	-90° (nadir)	east
180°	-90° (nadir)	south
270°	-90° (nadir)	west

As in our case the camera is on a gimbal stand, gimbal_roll always takes the value 0. For this reason, we used a simpler mathematical calculation that eliminates the roll value from the calculation, being aware that this can lead to an error in the calculation, especially in cases of sudden movements of the drone

(e.g., due to the gust of wind) when the gimbal does not manage to react in such a short time and level the camera.

For the detected object, as we described earlier, we take only one point on the image in pixels (u, v) and transform these coordinates into (x, y) coordinates on the ground. The starting point of the (x, y) coordinate system is the nadir point, the point located directly below the drone (Fig. 7).

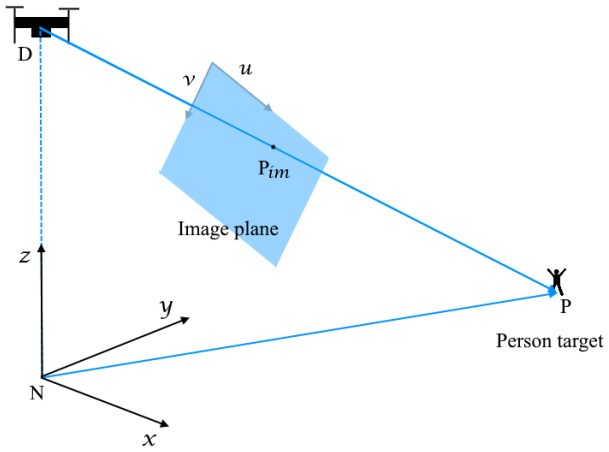


Figure 7: The coordinate system of the image (*u,v*) and the earth's coordinate system (*x,y,z*), where the origin of that system is the nadir point N (GPS_N, GPS_E).

We determine the distance of the detected object from the camera on the drone, i.e., the distance and azimuth of the detected object in relation to the nadir point based on the GPS coordinates of the drone, the FOV of the camera, the image resolution and the known AGL ("Above Ground Level") height of the drone, which is enough to determine the GPS coordinate of the detected object.

Using the FOV and aspect ratio, we determine the VFOV vertical field of view and the HFOV horizontal field of view.

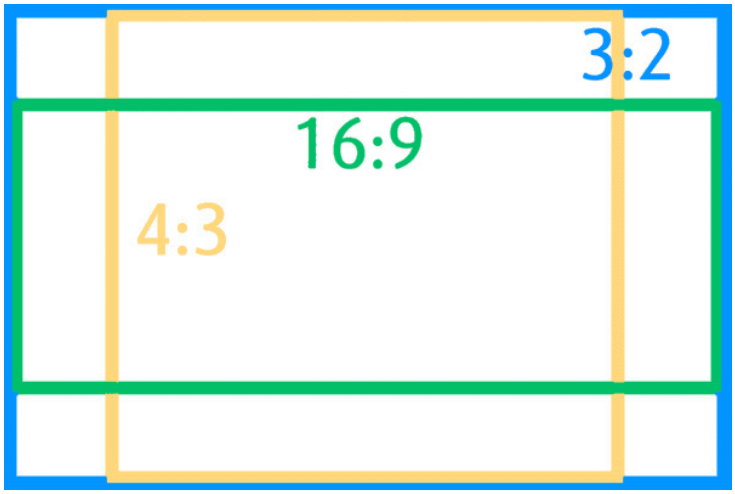


Figure 8: Image sensor – different aspect ratio

In Fig. 8. shows how the sensor area used during recording depends on the aspect ratio (image size). In our case, the FOV of the camera Field [25] is 84° and represents the diagonal angle for the 3:2 ratio. For different aspect ratios we have different FOV compared to the one defined in the settings or read from the EXIF data, i.e., 4:3 and 3:2 use the same sensor height and therefore have the same VFOV, while in the case of 16:9 and 3: 2 we have the same width or HFOV.

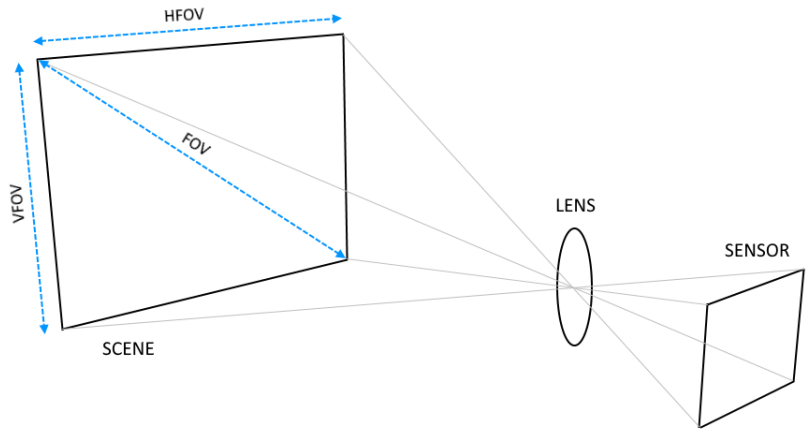


Figure 9: Horizontal and vertical FOV

From the data on the height and width of the image, we can determine the aspect ratio, i.e., how much of the sensor is used and thus correctly determine the VFOV and HFOV (Fig. 9).

Dividing the VFOV by the image height and the HFOV by the image width yields the angle in one image pixel by height and width. If we multiply the value of the angle per pixel by the position where the person is, we get the angle at which we see the person in the image τ (Fig. 11). This angle is measured from the upper edge of the image since the origin of the coordinate system of the image is in the upper left corner. The mathematical expression for determining the angle τ is:

$$\tau = \frac{v \cdot VFOV}{h} \tag{6}$$

Where (u, v) is the coordinate of the position of the object in the image, and h is the height of the image in pixels. Then the angles (α and β) at which the object is seen are determined (Fig. 10).

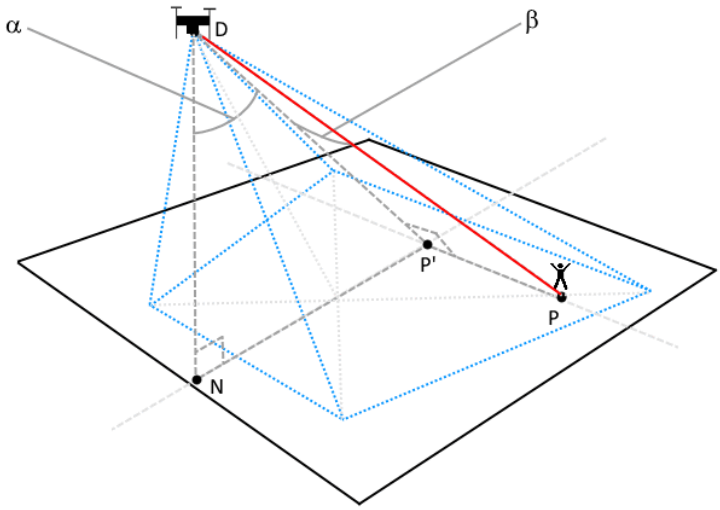


Figure 10: Display of aircraft and person in 3D space. α and β are the angles at which we see the person in the picture. The distance from the aircraft to the person is marked with a red line.

In vertical and low oblique shots, i.e., when the camera angle is less than or equal to VFOV / 2, we must check whether the object is in front of or behind the nadir point. If the object is in front of point N, the angle at which we see the object is equal to:

$$\alpha = \varphi + \left(\frac{VFOV}{2} - \tau\right) \tag{7}$$

Where φ is the angle of the camera (90 - Pitch). In the case when the object is behind the point N, the angle α is equal to:

$$\alpha = \tau - \frac{VFOV}{2} - \varphi \tag{8}$$

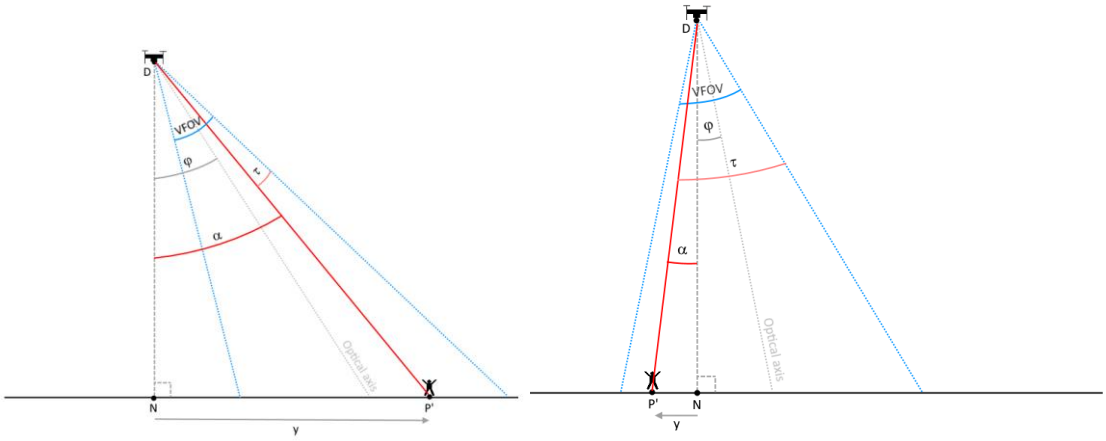


Figure 11: Side view of Fig 10. a) the person is in front of the nadir point (N) b) the person is behind the nadir point, in the pictures the distance from the drone (D) to the point (P') is marked with a red line, α is the angle at which we see the person in the y-z plane. The y represents the distance to the person from the nadir point

The following equation calculates the distance to the detected object on the y-axis:

$$y = h_{drone\_AGL} \cdot \tan \omega \tag{9}$$

After determining the distance in the y direction, we calculate the distance in the x direction. As with determining the distance in the y direction, in this case we consider two possibilities that the object is located on the left of the image (viewed from the center of the image - Fig. 12.):

$$\beta = \frac{HFOV}{2} - \sigma \tag{10}$$

or on the right:

$$\beta = \sigma - \frac{HFOV}{2} \tag{11}$$

where is:

$$\sigma = \frac{u \cdot HFOV}{w} \tag{12}$$

β is the angle at which we see the detected object in relation to the *y*-axis, if β = 0 then *x* = 0.

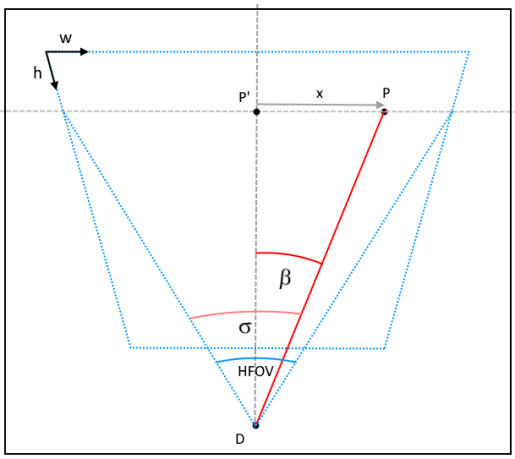


Figure 12: Top view, the case in which the person is located to the right of the center of the image (point P). σ – the angle at which we see the detected person measured from the left edge of the image.

From the AGL height at which the drone is located and the y distance (3) using Pythagoras, we can determine the distance from the drone to the detected object in the y plane ($dist_y = \overline{DP'}$). The distance along the x-axis of the detected object is obtained according to the equation:

$$x = dist_y \cdot \tan\beta \tag{13}$$

The distance from the nadir point to the location of the detected object ($\overline{NP}$) is obtained from:

$$D = \sqrt{x^2 + y^2} \tag{14}$$

While the azimuth (the angle at which we see the detected object in relation to the north pole of the earth) is:

$$\theta = gimbal\_Yaw \pm \beta \tag{15}$$

$$P = \ Geodesic.WGS84.Direct(GPS\_N, GPS\_E, \theta, D) \quad (16)$$

where gimbal_Yaw is the orientation of the drone in relation to the north, while the + or - sign is taken depending on where the object is located concerning the center of the image.

To determine the GPS coordinates of the object, we use the "Direct" function from the geographiclib.geodesic library [26] Equation 16. As input parameters, the function receives the latitude and longitude (lat, lng) of the drone position, azimuth, and distance to the object.

The height referred to as the relative height of the drone refers to the height concerning the ground at take-off, which can cause a significant error in the case of sloping terrain when there is a difference in height between the take-off position (home point) and the position where the drone is currently located, and between the position of the drone and the position where the person was detected (Fig. 13).

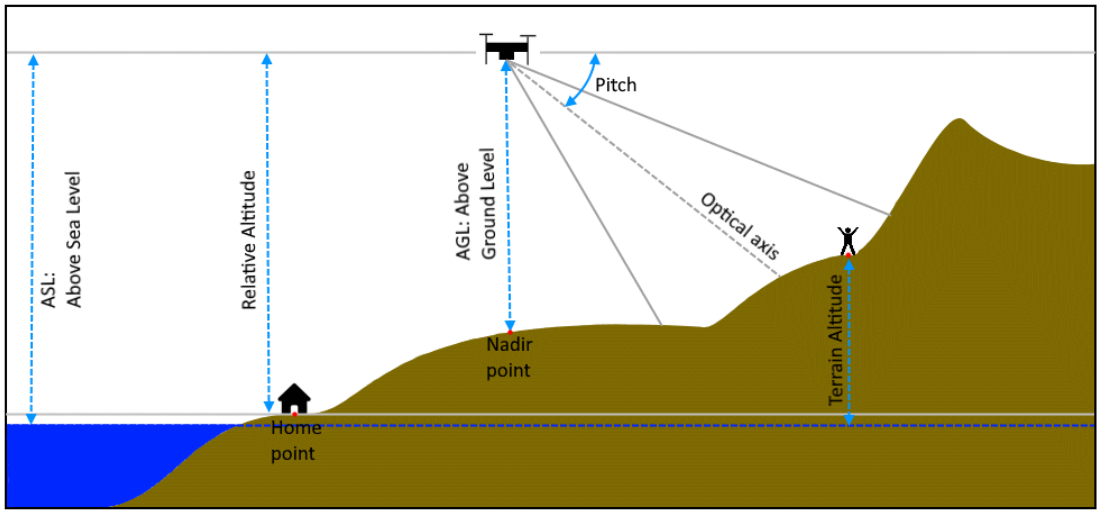


Figure 13: The different types of altitude

To reduce such an error, we introduce EU-DEM into the calculation. The EU-DEM is a 3D raster dataset with elevations captured at one arc second postings (2.78E-4 degrees) or about every 30 meters. The downloaded DEM is 4.56 GB, the area of the Republic of Croatia was cut using the QGIS program, which is 1.03 GB, while the area of Moslavina where the drone footage was taken, is 46.8 MB.

4.2. Geolocation algorithm with DEM calculation

In the first step, as in the previously proposed geolocation algorithm, we mathematically determine at what angle the person is seen from the drone, and from this data we get the distance at which the person is located (if the surface of the earth were flat). After that, we correct the position of the object, depending on the home point position, the position of the drone and the position where the person is located, i.e., depending on the height (which we read from the DEM file) for these positions. In particular, the relative height of the drone Fig. 13. which is written in the metadata is reduced/increased (depending on whether the drone is higher or lower in relation to the home point) by the absolute amount of the height difference between the DEM height of the home point and the DEM height of the nadir point of the drone. In this way, we more precisely determine the height at which the drone is located, which we use later in the algorithm as a constant. This step is necessary because the drone measures the relative height as the height in relation to the starting point from which it took off (home point), i.e. the relative height in the meta data is not the height that refers to the difference between the ground (nadir point) and the drone at the moment when is the image taken, i.e. in the case of terrain with a slope, the relative height of the drone is not the AGL height at which the drone is located.

The second parameter we use in the algorithm is the difference in the Nadir point's DEM height and the located object's (target) DEM height. The first step calculates the X position of the target (point X in Fig. 14). In this step, we check the Terrain Altitude in the DEM record for the GPS coordinate of the X position. If there is no difference in height between the nadir point and the X point, then the location is correct. If not, we look for the intersection of the line $\overline{DX}$ and the line $\overline{NY}$, and for the location of that intersection, we check the Terrain Altitude in the DEM database, which gives us the point Y_1. Suppose the height of the terrain of point Y1 is different from the height of the intersection of the lines. In that case, we look for a new intersection between the line $\overline{DX}$ and the new line $\overline{NY_1}$. We repeat this procedure until the values of the terrain height from the DEM and the calculated height of the intersection match with the desired accuracy (in our case to three decimal places).

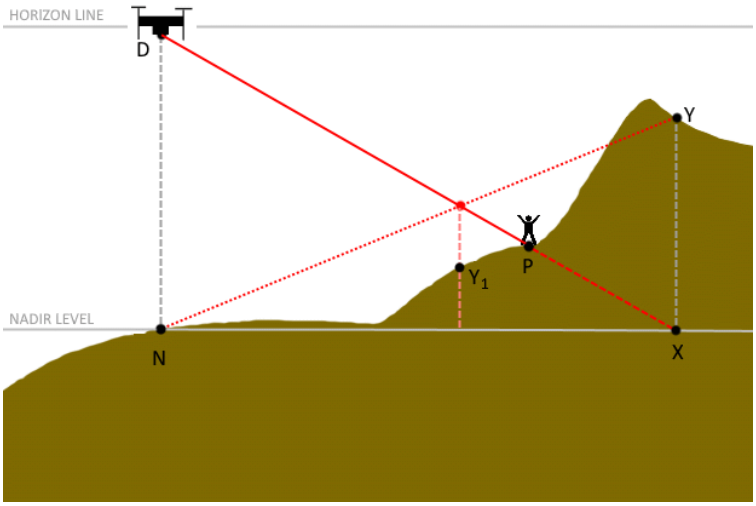


Figure 14: Terrain with a slope

4.3. Algorithm for determining the speed and direction of movement of the detected person

When a person is detected in two or more images, it is possible to determine the person's velocity from the obtained locations, which narrows the search zone and shortens the time needed for the ground teams to find the person. The speed of a person is calculated by first determining the distance between the two GPS locations dist(P_2, P_1) where the person was detected, and dividing that distance by the time that passed between the creation of the images where the person was detected. The direction of movement is determined as the geographic azimuth between the initial and final detection points.

$$v = \frac{\Delta s}{\Delta t} = \frac{dist(P_2,P_1)}{t_{img\_2}\text{-}t_{img\_1}} \quad (17)$$

Our prototype application has the option of saving the person's track (the position of the person where it was detected) in gpx format that can be displayed in some GIS programs. The Croatian Mountain Rescue Service uses a modified version of the Qgis program for search and rescue operations (Fig. 16).

Our application, after determining the traveled path and the direction of the person's movement, creates a search area for the search by adding and subtracting 30 degrees to the person's direction. The cartographer/search leader

can display the new search area in the GIS program and forward it to the field teams in their mobile applications.

Figure 16 shows an example of the initial phase of the search for a missing demented person that took place in the Moslavina region. The circles in the picture represent the statistical areas of previous finds of persons of the same type (e.g., dementia, child, mountaineer, mushroom picker...). The first circle, according to statistical data, represents a 25% probability that a missing person will be found in that area (which according to [27] is 300 m), the green circle is 50% (1000 m radius), while the probability of finding a missing person within the blue circle area is 75% (radius 2,400 m).

The subjective search zone in the picture is marked with a red line and it is the area that the rescuers will search, the subjective zone is then divided into zones, which are marked with the letters A, B, C, and D. Zone A is further divided into segments A1 - A10 where search teams are sent.

The star in the Fig. 16 (the center of the circles) indicates the IPP (Initial Planning Point), which is usually the point of last sighting or the last known location of the missing person.

If a missing person is detected using a drone, his location on the map is determined (yellow dot in the Fig. 16 within segment A3) and marked based on the calculated speed, the elapsed time since the image was created and the azimuth of the person's movement, marking a new segment of the search (red triangle in Fig. 16).

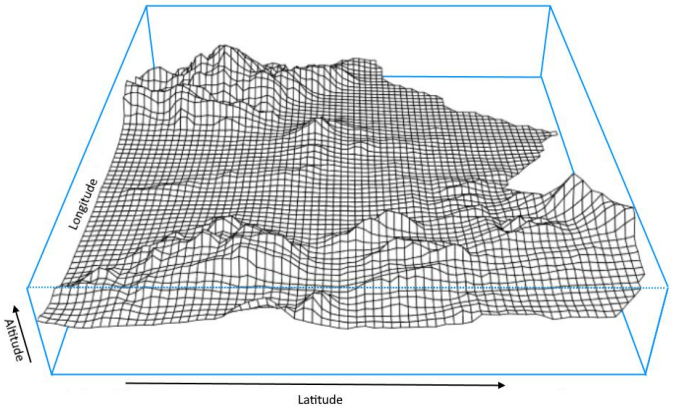


Figure 15: 3D view of DEM terrain [21]

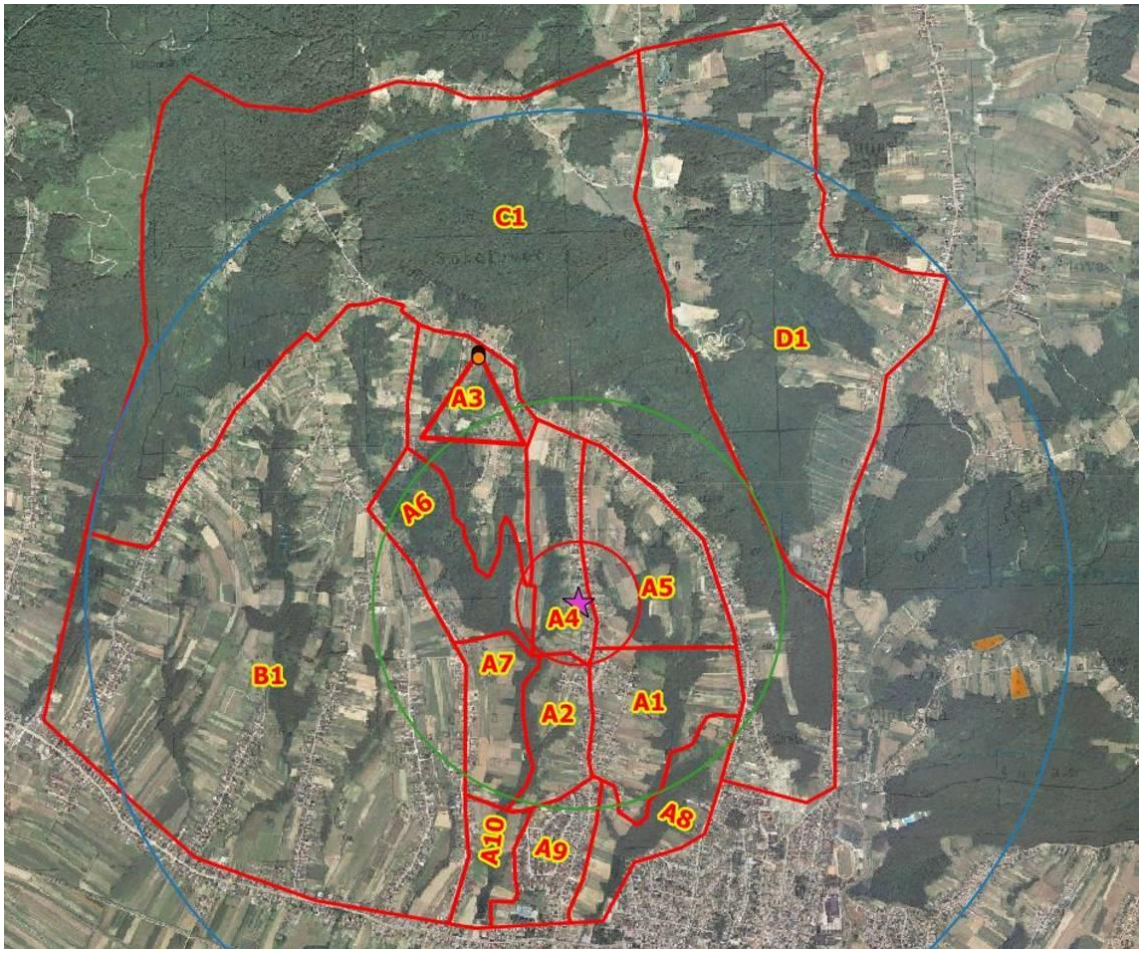


Figure 16: GIS in search and rescue operations

4.4. Experimental setups

For the experiment, we used the drone Phantom 4 Advanced [25] and Mavic 2 Enterprise Advanced [27], which were manually controlled. For the purposes of localization of the detected person, we used metadata that is recorded with the images taken by the drone during the flight and is related to the trajectory of the drone, the identification of the images and camera parameters at the time of taking the photo as shown in Table 1, and the height of the ground at the place where the drone took off (home point).

For automatic person detection in images, the SARD-832-1024 model, which was previously trained in search and rescue scenes for person detection, is used. In a further step, the geolocation is determined for the person detected in the images taken by the drone.

To evaluate the proposed method of localization of a person and prediction of the movement of person in real conditions, several experiments were carried out, which include the following relationships between the movement of a person and a drone:

1. the person and the drone are stationary

2. the person is stationary, and the drone is moving

3. the person moves while the drone is stationary/hovering

4. the person and the drone move

An experiment in which a person and a drone are stationary is presented in chapter 5.1. and the goal is to check the accuracy of the method, i.e., how much is the deviation from the actual spatial coordinate of the person. The experiment was carried out in two locations, flat terrain without a slope (meadow) and with a slope (vineyard). In the case of shots taken with the Phantom 4 Advance drone, the FOV of the camera is 84° while the image resolution is 5472 x 3648 px, the drone flew at a height of 30 m. The Mavic 2 Enterprise Advanced drone also flew at a height of 30 m above the home point, the result of the images taken by this drone is 8000 x 6000 while the FOV of the camera is also 84°.

The experiment in which the person is stationary, and the drone is moving is shown in 5.2. and in this case, the recording was also made in two locations. Two sets were filmed in the meadow, while three were filmed in the vineyard. A realistic scenario was applied for the case of searching for a missing person with this type of drone, which means that the drone flew over the terrain taking pictures in a certain time interval.

The case where a person moves while the drone hovers in place was filmed in two sets in a meadow and two sets in a vineyard. This method of searching is typical for drones and platforms like (DJI M30, DJI Matrice 210, or DJI Matrice 300) where the aircraft are located at heights between 100 and 300 m and inspect the space using zoom and movement of the camera only.

In the fourth experiment, both the person and the drone move. In order to have accurate data that can be compared with that which will be automatically estimated by applying the proposed method, a moving person has a navigation device, i.e., a navigation device that recorded the position where the person was. When filming with the Phantom 4 Advanced drone, the person has a Garmin GPSMAP 78 watch, and when filming with the Mavic 2 Enterprise Advanced drone, a hand-held navigation device GPSMAP 65s with support for multi-frequency systems / multiple GNSS that recorded the person's position every second and saved the data in a .gpx file which was later used for data comparison. In the same way, the position of the person's movement is monitored in the case of the person's movement while the aircraft is stationary.

The fifth experiment deals with the speed of a person's movement, i.e., estimating the speed and direction of a person's movement, which serves as a basis for determining a new search area.

In all cases, the results were also checked using the DEM file.

4.5. OBS AI Detector

To demonstrate and test the use of the proposed methods, a desktop application prototype (Fig. 17) was developed for use in search and rescue operations. During fieldwork, the controller of the drone is connected via HDMI cable (or wirelessly via Wi-Fi) to a computer on which the detector is running, which performs detection on the images obtained from the video during the flight (Fig. 18). The second part of the application is intended for off-line detection and for detection on recorded video or on recordings made during the flight in the field.

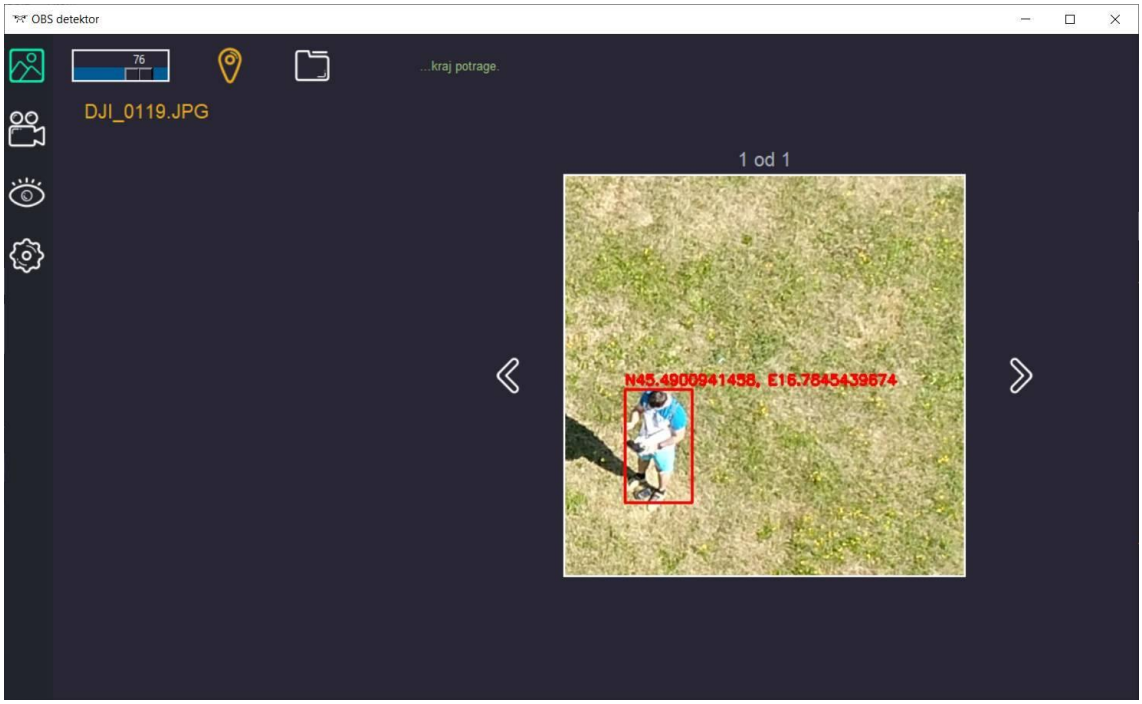


Figure 17: Presentation of the operation of the "OBS detector" application. In the picture, we see a detected person with the GPS coordinates where the person is located.


a)

b)

Figure 18: Display of the system ready for terrain search. a) The controller of the Phantom 4 Advanced aircraft is connected via an HDMI cable to the computer on which the image from the controller's screen is projected. b) Mavic 2 Enterprise Advanced aircraft controller connected to a computer via wifi.

5. Experimental Results

In the experiments, the accuracy of locating people using the proposed method is tested. The result is displayed as the distance between two points measured in meters. In Fig. 19 shows the GT location of a person (yellow pin) and the location of the same person determined by the proposed Algorithm 1 (blue pin) for one photo from the set. Mean Error (Equation 19) represents the mean value of all distances ΔP_i between the points determined according to (Equation 16) and the GT point for each image in a certain set of recordings. Max Error is the greatest distance, i.e. the largest error the method made in that set, and Min Error is the smallest error. Both data are equally important, because it is difficult for remote pilots to detect a stationary object, and it is possible that in the set that is subsequently analyzed, there will be only one shot of the desired object from which the position in the Earth's coordinate system will be determined.

$$\Delta P_i = \ Geodesic.WGS84.Inverse(P_i, P_{GT_i}) \tag{18}$$

$$Mean\ Error = \ \frac{\sum_{i=1}^{n} \Delta P_i}{n} \tag{19}$$

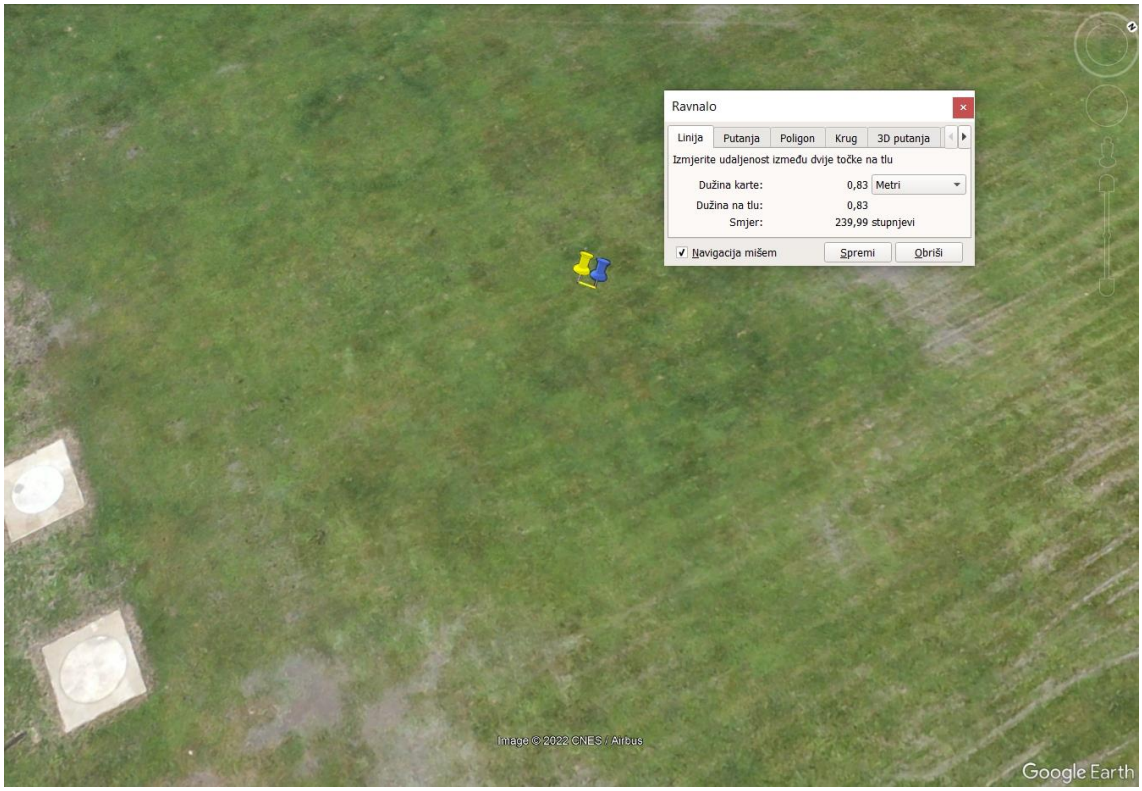


Figure 19: Displaying the GPS location of a person in the Google Earth program. The yellow pin represents the person's GT location, while the blue pin is the location obtained by the proposed algorithm. The yellow direction represents the distance, which in this case is 0.83 meters.

5.1. The person and the drone are stationary

Table 4 shows the results for sixteen sets. The second column represents the number of shots taken two seconds apart for each set. The drone hovers in one position using "P-mode" positioning. This mode works best with a strong GPS signal and will provide the most stable flight. "P-mode" allows the drone to maintain its position and altitude even in moderate wind. Each set is recorded at a new position where the drone hovers, which is what we see in the difference in the results. Also, the person is located at different coordinates for the Phantom sets compared to the Mavic, as well as the home point for each aircraft Fig. 18. The differences in the results arise from the inaccuracy of the sensors in the spacecraft, we can also address a significant role in the accuracy of the results to the influence of the wind, i.e. slow response of the gimbal system and positioning of the aircraft in such cases. Part of the inaccuracy can also be

addressed to the resolution of the DEM file, but even such a DEM improves our results by more than 50% compared to the results when no DEM was used.

Table 4. Coordinates calculation of person standing on known location.

#	Data Set	Number of images in set	Mean Error (m)	Max Error (m)	Min Error (m)
1	Phantom - livada 1	10	0.903	1.040	0,790
2	Phantom - livada 2	10	3.392	4.181	2,732
3	Phantom - livada 3	10	2.872	4.344	1.327
4	Mavic - livada 1	6	2.729	2.865	2.563
5	Mavic - livada 2	10	4.361	4.836	4.102
6	Mavic - livada 3	10	1.645	1.804	1.462
7	Phantom - vinograd 1	10	6.326	6.494	6,239
8	Phantom - vinograd 2	10	7.149	7.438	6.919
9	Mavic - vinograd 1	10	24.701	25.219	24.236
10	Mavic - vinograd 2	10	27.084	27.370	26.850
11	Mavic - vinograd 3	9	19.704	20.006	19.386
12	Phantom - vin 1 DEM	10	5.339	5.587	5.234
13	Phantom - vin 2 DEM	10	2.270	3.782	1.301
14	Mavic - vin 1 DEM	10	11.026	11.541	10.605
15	Mavic - vin 2 DEM	10	12.377	12.604	12.170
16	Mavic - vin 2 DEM	9	9.102	9.367	8.836

Figure 20: Display the positions where the person is in the case of recordings made

with the Phantom 4 Advanced and Mavic 2 Enterprise Advanced aircraft, as well as the position from which the aircraft took off (home point) for cases when the person is stationary.

5.2. The person is stationary, the drone is moving

In this scenario, the drone flies at a speed that allows the pilot to search the target area well. The person is stationary, which is a very common case in search and rescue operations. Missing persons very often not moving after some period. Most often, the person is exhausted and can no longer move independently, also an injury to the person can be a reason for not moving.

Table 5 shows the results related to the estimation of the distance of a person from a drone in flight. The smallest error was achieved in the meadow where the terrain is flat so that the most accurate results were achieved. In the case of vineyards where the terrain is sloping, it was shown that the use of DEM significantly improves the precision of distance determination compared to when DEM was not used. For the Phantom VP 1a set the improvement is 40%, for the Phantom VP 2a 53%, and for the Phantom VP 3a case 46%.

We believe that the calculated location on which the person was detected is important data to help the rescuers on the ground to reach the located person in the shortest time (Fig. 19 and Fig. 20). However, it should be investigated what is the maximum acceptable error in locating the person and that for different cases of terrain configuration and flight height, i.e., different cases of detection range (AMDR - Average Maximum Detection Range [28]).

Table 5. Coordinates calculation of person standing on known location.

#	Data Set	Number of images in set	Mean Error (m)	Max Error (m)	Min Error (m)
1	Phantom LP 1	10	8.963	10.539	7.870
2	Phantom LP 2	10	8.704	11.595	6.212
3	Phantom VP 1a	4	18.388	29.283	8.424
4	Phantom VP 2a	7	50.540	73.028	14.447
5	Phantom VP 3a	9	51.267	98.108	22.783
6	Phantom VP 1a DEM	4	10.935	15.833	5.630
7	Phantom VP 2a DEM	7	23.604	34.681	7.327
8	Phantom VP 3a DEM	9	27.911	66.887	14.762

5.3. The person moves, the drone hovers in one position

This search method setup is more often used with platforms where cameras with different lenses and different zoom capabilities are installed. Then the remote pilot places the aircraft at a higher altitude to hover and uses the controller to survey the area just by moving the camera and using the zoom. Four sets were made with the Mavic 2 EA drone, two sets on non-sloped terrain (Mavic LH 1 and Mavic LH 2) and two sets on sloped terrain (Mavic LH 1 and Mavic LH 2). Table 6 shows the obtained results.

Table 6. Coordinates calculation of moving person recorded from a hovering drone.

#	Data Set	Number of images in set	Mean Error (m)	Max Error (m)	Min Error (m)
1	Mavic LH 1 DEM	10	7.682	10.913	4.650
2	Mavic LH 2 DEM	21	4.994	10.375	2.216
3	Mavic VH 1 DEM	11	10.010	14.107	7.131
4	Mavic VH 2 DEM	13	4.613	9.257	2.035

Fig. 21 shows the points/traces for each image in the set using the QGIS program at a scale of 1:505. The yellow lines/dots represent the calculated positions of the detected person in the image, while the orange lines/dots represent the positions recorded by the handheld GPS device (Garmin GPSMAP 65s) worn by the person/target during the recording set and used as a ground truth. The GPS device is set to record a person's position every second. And for comparison, i.e., to check the accuracy, the positions that were created at the same moment (the moment of the image creation) are taken.

Figure 21: Representation of a person's movement using tracks and dots, the orange line/dots represent the calculated positions of the person, while the yellow lines/dots represent the points measured using a hand-held GPS device (Garmin GPSMAP 65s) for set a) Mavic HL 1 DEM, b) Mavic HL 2 DEM, c) Mavic HV 1 DEM and d) Mavic HV 2 DEM. In each image, there is also the beginning of the proposed search segment (a pointed red corner, the top of which is located at the first detected position of the missing person), from which the direction of the person's movement can be seen.

5.4. The person and the drone are moving

A realistic scenario, especially in the initial phase of the search. Table 7 shows the results of the locating error for the case in which a person and a drone are moving. Results are shown for both aircraft, Phantom 4A and Mavic 2 EA. The test was performed out in two locations. Mean error ranges from 2.298 m to 13.950 m. Which is a good result if we consider that the field teams for the 75% circle need to search 19 625 000 m^2.

Table 7. Coordinates calculation of moving person.

#	Data Set	Number of images in set	Mean Error (m)	Max Error (m)	Min Error (m)
1	Phantom LM 1 DEM	13	2.298	4.406	0.996
2	Phantom LM 2 DEM	20	7.452	13.808	2.395
3	Phantom LM 3 DEM	10	4.588	8.434	2.085
4	Mavic LM 1 DEM	10	6.943	7.951	5.943
5	Mavic LM 2 DEM	5	6.692	9.623	4.200
6	Mavic LM 3 DEM	7	8.486	11.065	7.754
7	Mavic LM 4 DEM	3	5.031	5.259	4.732
8	Phantom VM 1 DEM	6	12.388	20.923	7.133
9	Phantom VM 2 DEM	5	13.950	25.072	4.439
10	Mavic VM 1 DEM	5	12.054	16.445	8.210
11	Mavic VM 2 DEM	6	10.413	12.595	7.690
12	Mavic VM 3 DEM	5	8.484	9.580	7.283
13	Mavic VM 4 DEM	12	9.266	10.977	6.635

Figure 22: A set of photos taken in real conditions on a sloped field ("vineyard")

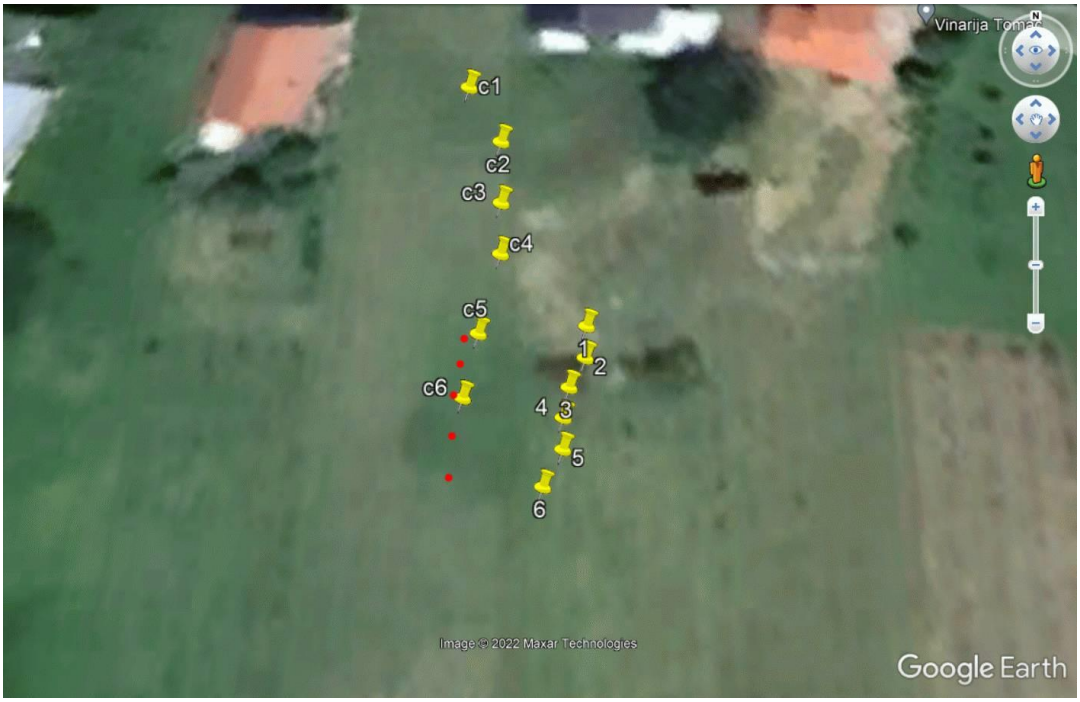


Figure 23: Display of a person's GPS location in the "Google Earth" program. Points 1-6 are coordinates recorded with a hand-held GPS device, points c1-c6 are coordinates determined using the proposed algorithm. Red points indicate the estimated position of the person with regard to the photos shown in Fig. 22.

5.5. Velocity calculation

The speed of a person's movement is calculated as the quotient of the distance and the time in which that distance was covered. To calculate the route, the initial position of the GPS location where the person was first detected is used, and as the final position the GPS location where the person was last detected, and the distance between these two points representing the route is calculated. To determine the time in which the person has passed that route, the time from the creation of the first image in which the person was detected to the creation of the last image is taken.

Additionally, from the two GPS coordinates related to the initial and final location of the person, using the geographiclib.geodesic [26], along with the distance between the points, the azimuth between those points is obtained, i.e. the direction of a person's movement in relation to the north pole of the earth.

Further, it is necessary to check how much time has passed from the moment the last image was taken to the moment of detection, i.e., statistically determine how far the person could have moved if he continued to move at the calculated speed. The program outputs a .gpx file that can be displayed in a GIS application and is used to define a new proposed search area and the trail along which the missing person moved. For each set (Table 8), we determined the person's speed per gpx track as the ground truth and speed concerning detection.

Table 8. Movement speeds of the detected person in more than two photos.

#	Data Set	Number of images in data set	Calc speed (m/s)	Garmin GPSMAP 78/ Garmin GPSMAP 65s speed (m/s)	Speed error (%)
1	Phantom LM 1 DEM	13	1,172	1,085	+ 8.01
2	Phantom LM 2 DEM	20	1,639	0,799	+ 105.13
3	Phantom LM 3 DEM	10	1,199	1,257	- 4.61
4	Mavic LM 1 DEM	10	1.088	1.344	- 19.04
5	Mavic LM 2 DEM	5	1.172	1.259	- 6.91
6	Mavic LM 3 DEM	7	1.617	1.284	+ 25.93
7	Mavic LM 4 DEM	3	1.141	1.166	- 2.14
8	Phantom VM 1 DEM	6	2,571	1,096	+ 134,58
9	Phantom VM 2 DEM	5	1,053	1,192	-11.66
10	Mavic VM 1 DEM	5	2.062	1.029	+ 100.38
11	Mavic VM 2 DEM	6	0.421	0.854	- 50.70
12	Mavic VM 3 DEM	5	0.563	0.371	+51. 75
13	Mavic VM 4	12	1.386	1.250	+ 10.88

Table 8 shows the speed of motion of the detected person in the image. Calc speed is the average speed by taking the calculated points of the person's position. In contrast, Garmin speed is calculated using the points determined by the Garmin handheld GPS device (for the Phantom, it is GPSMAP 78, while for the Mavic, set GPSMAP 65 was used). The number of images in the data set represents the number of images in which the person was detected.

The example of data for the motion of a person and a drone in the case of the Mavic LM 1 DEM set is shown in Fig. 24. The yellow track shows the movement of the person from the calculated coordinates, while the orange track shows the movement of the person from the coordinates measured by the GPS device. The blue track is the motion of the drone.

Fig. 22 shows images from the Phantom VM 1 DEM set, whit marked locations where the person is. By comparing the calculated positions and the grown truth we get using the handheld GPS Garmin GPSMAP 78 device, the estimated positions are shifted to the right about the actual position observed in the Google Earth application (Fig. 23). In the same image, the positions of the person are marked with red points according to the assessment of an expert who considers the configuration of the terrain, his experience, and the visible environment in the images of the Google Earth application. The goal was to show that even the GPS devices used in search and rescue operations as the grown truth have an error of ± 3 m. Hence, the results presented in this paper provide sufficiently precise data on the missing person's location and enable ground teams to access a missing person quickly.

Figure 24: The display of points in the QGIS program for the movement of a person, and the yellow track/points represent the calculated coordinates, while the next points/track show the coordinates determined by manual navigation, the blue track/points represent the positions of the drone at the time the photos were taken.

6. Conclusion and future work

This paper presents a complete framework for person detection and geolocation using a single image captured by a drone camera. The person in a non-urban area, with a small pixel size in the image, was detected using the model trained by Yolov4 named SARD-832-1024. A passive geolocation method is presented

to calculate the GPS coordinates of the detected person. Suppose there are several images in which the detected person is located. In that case, the proposed system calculates the speed at which the person moves using the distance traveled between the detected positions of the person and the time elapsed between the photos.

We established an experimental system consisting of a DJI Phantom 4 Advance aircraft and a laptop computer to analyze the images after the flight. The experiment results show that with this approach, the missing person can be located precisely enough so that ground teams can approach the person in the shortest possible time in case the person is injured/not moving or reduce the search area in case the person is moving. Moreover, all the methods mainly depend on the low-accuracy sensors on the drone. The results show that the person detection system is cost-effective and efficient. In future work, the system needs to be adapted for tracking multiple people, i.e., distinguishing them in case of multiple detections in one photo.

References

[1] S. Sambolek and M. Ivasic-Kos, “Automatic person detection in search and rescue operations using deep CNN detectors,” *IEEE Access*, vol. 9, pp. 37905–37922, 2021, doi: 10.1109/ACCESS.2021.3063681.

[2] K. Phillips *et al.*, “Wilderness Search Strategy and Tactics,” *Wilderness Environ. Med.*, vol. 25, no. 2, pp. 166–176, 2014, doi: 10.1016/j.wem.2014.02.006.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” 2014, doi: 10.1109/CVPR.2014.81.

[4] R. Girshick, “Fast R-CNN,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, 2015, doi: 10.1109/ICCV.2015.169.

[5] W. Liu *et al.*, “SSD: Single shot multibox detector,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9905 LNCS, doi: 10.1007/978-3-319-46448-0_2.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-December, doi: 10.1109/CVPR.2016.91.

[7] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in *Proceedings*

- *30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, vol. 2017-January, doi: 10.1109/CVPR.2017.690.

[8] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *Tech Rep.*, 2018.

[9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” 2020.

[10] S. Martínez-Díaz, “3D Distance Measurement from a Camera to a Mobile Vehicle, Using Monocular Vision,” *J. Sensors*, vol. 2021, 2021, doi: 10.1155/2021/5526931.

[11] C. Xu, D. Huang, and F. Kong, “Small UAV passive target localization approach and accuracy analysis,” *Yi Qi Yi Biao Xue Bao/Chinese J. Sci. Instrum.*, vol. 36, no. 5, 2015.

[12] DJI, “DJI Matrice 30.” https://www.dji.com/hr/matrice-30/specs.

[13] S. Sambolek and M. Ivasic-Kos, “Person Detection in Drone Imagery,” 2020, doi: 10.23919/SpliTech49282.2020.9243737.

[14] S. Sohn, B. Lee, J. Kim, and C. Kee, “Vision-based real-time target localization for single-antenna GPS-guided UAV,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 4, 2008, doi: 10.1109/TAES.2008.4667717.

[15] X. Zhao, F. Pu, Z. Wang, H. Chen, and Z. Xu, “Detection, tracking, and geolocation of moving vehicle from UAV using monocular camera,” *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2929760.

[16] J. Sun, B. Li, Y. Jiang, and C. Y. Wen, “A camera-based target detection and positioning UAV system for search and rescue (SAR) purposes,” *Sensors (Switzerland)*, vol. 16, no. 11, 2016, doi: 10.3390/s16111778.

[17] Y. Zhang, C. Lan, Q. Shi, Z. Cui, and W. Sun, “Video image target recognition and geolocation method for UAV based on landmarks,” in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 2019, vol. 42, no. 2/W16, doi: 10.5194/isprs-archives-XLII-2-W16-285-2019.

[18] Y. Pi, N. D. Nath, and A. H. Bezhadan, “Deep Neural Networks for Drone View Localization and Mapping in GPS-Denied Environments,” 2020, doi: 10.46421/2706-6568.37.2020.paper001.

[19] A. El Habchi, Y. Moumen, I. Zerrouk, W. Khiati, J. Berrich, and T. Bouchentouf, “CGA: A New Approach to Estimate the Geolocation of a Ground Target from Drone Aerial Imagery,” *4th International Conference on Intelligent Computing in Data Sciences, ICDS 2020*. 2020, doi: 10.1109/ICDS50568.2020.9268749.

[20] “EU-DEM.” https://www.eea.europa.eu/data-and-maps/data/copernicus-land-monitoring-service-eu-dem (accessed Oct. 10, 2022).

[21] “Digital Elevation Models.” https://www.cdema.org/virtuallibrary/index.php/charim-hbook/data-management-book/3-base-data-collection/3-2-digital-elevation-models.

[22] T. Y. Lin *et al.*, “Microsoft COCO: Common objects in context,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8693 LNCS, no. PART 5, doi: 10.1007/978-3-319-10602-1_48.

[23] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, “CSPNet: A new backbone that can enhance learning capability of CNN,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2020, vol. 2020-June, doi: 10.1109/CVPRW50498.2020.00203.

[24] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, 2015, doi: 10.1109/TPAMI.2015.2389824.

[25] “Phantom 4 Advanced - Specs.” https://www.dji.com/hr/phantom-4-adv/info#specs. (accessed May 05, 2021).

[26] “GeographicLib.” https://geographiclib.sourceforge.io/ (accessed Oct. 10, 2022).

[27] “Mavic 2 Enterprise Advanced - Specs - DJI.” https://www.dji.com/hr/mavic-2-enterprise-advanced/specs (accessed Dec. 16, 2022).

[28] R. J. Koester, “Sweep width estimation for ground search and rescue,” *POTOMAC Manag. Gr. ALEXANDRIA VA*, 2004.

RAD 8. APPLICATION OF RAYCAST METHOD FOR PERSON GEOLOCALIZATION AND DISTANCE DETERMINATION USING UAV IMAGES IN REAL-WORLD LAND SEARCH AND RESCUE SCENARIOS

Ovaj rad je objavljen kao: Paulin, Goran, Sasa Sambolek, and Marina Ivasic-Kos. "Application of raycast method for person geolocalization and distance determination using UAV images in Real-World land search and rescue scenarios." *Expert Systems with Applications* 237 (2024): 121495.

https://www.sciencedirect.com/science/article/abs/pii/S0957417423019978

1. Introduction

People are adventurous creatures. For numerous reasons, they enjoy spending time in the wilderness. Unfortunately, occasionally they get lost or injured. When this happens, their life is at stake, and their survival depends on being efficiently found and rescued in the shortest possible time. A search and rescue operation (SAR) is launched after the accident is reported, and all possible resources are activated (people, search dogs, vehicles, helicopters, and drones).

Today, drones and unmanned aircraft systems (UAS) are ubiquitous in various SARs due to the many benefits that their use provides. Drones equipped with cameras (RGB and/or thermal) are potent reconnaissance systems (Doherty & Rudol, 2007) and area mapping tools (Boccardo et al., 2015) in SAR operations. They can provide crews with almost real-time situational awareness helpful for locating a missing person, speeding up the rescue mission, and increasing the survival rate. However, sophisticated drones are often not available in SAR practice. Instead, low-cost commercial drones are used without Real-Time Kinematic (RTK) modules, laser rangefinders, and stereo cameras. In order to make them usable for participation in SAR missions, it is necessary to research and design a system that includes an efficient method for person geolocalization and distance determination using monocular unmanned aerial vehicle (UAV) images and basic metadata such as drone GPS position and orientation.

Thanks to the recent development of photogrammetry and computer vision, especially the methods for automatically detecting and tracking objects and for geolocalization (Paulin et al., 2021), fast and automated processing of drone data recorded from a bird's eye view has been enabled (Sambolek & Ivasic-Kos, 2021). A ground target (person) within an image frame can be detected automatically in real-time using machine learning techniques. A model can be built to detect a person in images recorded in the same conditions as the images used to train the model. Nowadays, detection models are based on deep learning and deep convolutional neural networks, and they achieve very good results in detecting people (even from a top-down perspective). Data such as the drone's current position and the camera's position and orientation in relation to the

drone's body are useful for determining the target's or the victim's coordinates. The built-in GPS receiver provides the drone's position, which is stored in image metadata along with other data, such as camera orientation.

The accuracy of location determination depends on the accuracy of the data obtained from the drone sensors. In the past, the photogrammetric process of extracting accurate geometric information was based almost exclusively on images in which the camera's position was perpendicular to the recording surface. However, in SAR operations, the drone pilot usually captures the area with the camera set to an angle that covers as much area as feasible, and possibly when the person is detected, the drone is just above him or in an ideal position perpendicular to the ground.

Oblique aerial images are taken with the camera axis intentionally inclined with respect to the vertical. They are characterized as either high oblique if tilted sufficiently to show the horizon or low oblique if they do not include the horizon (Verykokou & Ioannidis, 2018). Some of the basic characteristics of oblique photographs are their trapezoidal footprint, the significant change of scale, the coverage of a larger ground area compared to vertical images taken from the same altitude by the same camera, and the intuitive interpretation by people because they are accustomed to seeing ground features from a similar perspective (Verykokou & Ioannidis, 2015).

Distance determination, as a part of the geolocalization process, has an important role in many areas, including SAR and 3D reconstruction (Hosseinpoor et al., 2016; Sambolek & Ivasic-Kos, 2021). The fastest way to measure distance is with a laser. The drone-mounted laser measures the distance for a single point, which usually needs to be manually marked on the remote controller screen. However, due to the small screen size, the drone pilot may not see the object of interest at the given moment. For that reason, images are further analyzed after flight operations (offline) (Sambolek & Ivasic-Kos, 2020), but measuring the object's distance in the image with a laser is no longer possible. In the case of SAR operations, it is also impossible to use Ground Control Points (GCP) as a method

of indirect geolocation (F. He et al., 2018) since placing a GCP on inaccessible terrain is impractical.

There are alternative methods that can be divided into single-image and multi-image methods. Algorithms such as Structure from Motion (SfM) and Simultaneous Localization and Mapping (SLAM) are approaches using multiple images. The general principle behind determining depth from multiple images is finding corresponding features in them and resolving the depth by looking at the change in their positions (Forlani et al., 2019; Vidal et al., 2018; von Stumberg et al., 2017). SLAM implementations require large amounts of processing power and memory to generate accurate maps.

Another approach often applied in the automotive industry and used to estimate distances to objects and potential obstacles is the use of stereo cameras, where the depth is determined by triangulation on the obtained images and binocular disparity between the image of the object in the left and right cameras (Leu et al., 2012; Zhang et al., 2018).

Also, image metadata, which includes the position and orientation of the drone and camera in space, is frequently used (Haseeb et al., 2018; Sun et al., 2016; Verykokou & Ioannidis, 2015; Zhao et al., 2019; Zhu & Fang, 2019) for determining object distance. However, the problem with this approach is mainly the small number of sensors installed on the drone and their low reliability. Accordingly, the information and signals obtained are often inaccurate and deficient.

The target geolocation algorithm based on a single image and the Earth ellipsoid model (Cai et al., 2022) is still the mainstream of current research and the basis of other relatively extensive researched algorithms. However, when the distance determination problem is observed in 3D, it may be reduced to locating a point of interest on the ground using image metadata. A similar problem occurs in computer graphics, specifically raytracing, when searching for line-surface intersections (Roth, 1982). It is also encountered in 3D computer games, where is known as raycasting (Pietroszek, 2018), and used to determine collisions between 3D objects. Once we find the intersection point, determining the distance

is then a simple task of calculating the distance between the camera position and the point of intersection. Sheng (Sheng, 2004, 2005) tests three different methods to solve the optic ray-DEM intersection: Iterative Photogrammetry (IPG), Ray-Tracing (RT), and Iterative Ray-Tracing (IRT). He concludes that the RT method, essentially based on raycasting, is more accurate than the others. However, he rejects it in favor of the IRT method because it is considered too computationally demanding. It was a meaningful criticism in 2004 but no longer is today.

For the above reasons, building upon previous simulation experiments (Paulin et al., 2021), we propose using the raycast method for person geolocalization and distance determination in different real-world scenarios and offer recommendations for its successful use. The proposed approach allows using low-cost commercial drones with a monocular camera and no RTK module while enabling laser rangefinder emulation during offline image analysis. It overcomes problems encountered in previously published methods and achieves the best result (geolocation error of 0.7 m) in actual SAR mission conditions. We also proposed a new evaluation metric (ErrDist) for person geolocalization and prepared and released our SAR-DAG_raycast dataset.

The main contributions of this paper are:

- prototype of a system for automatic person detection and geolocation in search and rescue missions (SAR-DAG);
- proposed geolocating method based on raycast for use in SAR missions;
- proof that the proposed geolocation method can be adapted for real-world scenarios with recommendations for use;
- proposed ErrDist evaluation metric for person geolocalization;
- SAR-DAG_raycast dataset.

The rest of the paper is organized as follows: Section 2 provides an overview of previous research and papers related to CNN-based object detection, distance, and geolocation determination. Section 3 describes the SAR-DAG system prototype. In Section 4 proposed geolocation method based on the raycast is detailed, describing the algorithm, 3D terrain generator, and the raycaster tool. Section 5 describes experiments, including datasets, evaluation metrics, 3D

terrain generation, and raycaster management, and analyzes the obtained results. Section 6 gives recommendations for using the proposed geolocation method in real-world scenarios. The paper ends with the conclusion and directions for future research.

2. Related work

Today, drones and unmanned aircraft systems (UAS) are almost inevitably used in various search and rescue operations (SARs) to reconnoiter areas and locate missing or injured persons. As a rule, drones are equipped with RGB cameras, and more and more often with thermal cameras, which allows them to capture a situation image or video of the monitored terrain at all times of the day. The operator can view the recorded material in real-time or, subsequently, offline. For SAR operation, it is important to detect the person in the image but also to estimate the distance of the person from the drone and thus to geolocate the person, which is not easy to determine since offline analyzed images are primarily taken from an oblique perspective.

With the development of laser radar and machine vision, non-contact active and passive distance measurement methods have emerged (Aki et al., 2016; Bradshaw et al., 2005). Active UAV methods for object geolocalization are based on the laser rangefinder. DJI M30 drone (*DJI Matrice 30*, n.d.) has a laser rangefinder that can provide precise coordinates of objects up to 1,200 meters away. However, these devices are not useful when the detection of objects is done offline in captured images (Sambolek & Ivasic-Kos, 2021).

For UAVs without a laser rangefinder, GPS and inertial measurement units (IMU) can provide the location and altitude of the UAV, so the passive methods are widely chosen. This data is also recorded in the image metadata. In general, the mathematical transformation methods are based on the measurement of the intrinsic and external orientation parameters of the aerial image and the 3D coordinates of the UAV to calculate the 3D coordinates of the ground object.

The object geolocation method described in (Zhao et al., 2019) first transforms the pixel coordinate frame to the East-North-Up (ENU) frame, using the intrinsic

camera parameters (image width, image height, and focal length of the camera). Then the depth estimation of the object is performed, assuming that the observed space is relatively flat. The object distance is calculated from the drone's altitude and cosine of the angle at which the object is seen. Finally, the ENU frame is converted to GPS coordinates. For a flight height of 100 m, an accuracy of 5 m was achieved.

Leira et al. (Leira et al., 2015) try to find the North-East-Down (NED) coordinates of the object from the pixel coordinates of the object in the thermal image using a scaling factor obtained from known intrinsic and external camera parameters. In order to improve localization accuracy, they developed a method of calibrating thermal cameras. For flights at heights between 50 and 100 m, they achieve an accuracy of 7.8 m.

A different approach to obtaining GPS coordinates of the object is to create orthophoto images for a specific target area (Suziedelyte Visockiene et al., 2016). The problem with this method is that it takes a lot of time.

A method using georeferenced images is described in (Conte et al., 2008). The authors seek to match the image taken by the aircraft with existing georeferenced images such as Google Earth. In the experiment, a ground object was geolocalized with an accuracy of 2.3 meters from a flight altitude of 70 meters.

The object positioning by using the camera's FOV and UAV's altitude is given in (Sun et al., 2016). The ratio between the distance and pixels is assumed to be a linear relationship for vertical aerial photography (*Types of Aerial Photograph*, n.d.). The calculation of ground coordinates from a single low oblique aerial image is given in (Verykokou & Ioannidis, 2015). Tilt angle is used for the calculation, in addition to the focal length and the drone altitude.

Stereo cameras are used in the automotive industry for collision prevention (Leu et al., 2012) by estimating depth using a binocular disparity map between the image of the object in the left and right cameras, which is then segmented based on pixel intensity to detect different objects in the scene. High-speed cameras of 100 fps and low latency below 0.1 are processed with computationally expensive

algorithms. For each detected object, its distance to the front of the vehicle is calculated, and the collision warning module estimates the degree of danger. In (Zhang et al., 2018), stereo vision is used in simulations and actual flight to geolocate a target based on the relative height between the drone and the target using image information retrieved from the drone metadata. The goal of the proposed method is to avoid using georeferenced terrain databases and position sensors used to determine the drone's turning angle, given that the sensors on the drone are mostly of low quality and provide incorrect information.

Additional data recorded in the image metadata, which includes the position and orientation of the drone and camera in space, is used in (Haseeb et al., 2018; Sun et al., 2016; Verykokou & Ioannidis, 2015; Zhao et al., 2019; Zhu & Fang, 2019) in case of single image. However, the higher the desired data accuracy, the higher the hardware's cost and/or size. The problem with this approach is the inaccuracy of the measured data because, due to payload constraints, UAVs typically use smaller and error-prone senzors.

Paper (Pan et al., 2023) presented a framework for geolocating a moving target using images captured from a UAV. Unlike traditional approaches, the proposed framework relies solely on monocular vision and does not require laser rangefinders or multiple UAVs. It transforms the problem of moving target geolocation into stationary target geolocation by matching corresponding points. The framework utilizes Siamese-network-based models for point matching and introduces compensation values to improve matching accuracy. The experiment's results demonstrated successful geolocation of the moving target on the ground, with mean absolute errors of 0.046 m, 0.044 m, and 0.165 m for the X, Y, and Z coordinates, respectively.

Cellular-based drone search and rescue geolocalization system SARDO (Albanese et al., 2022) aims to find victims' locations by keeping track of their mobile phone signals in disaster areas with the information collected by a single UAV that acts as a portable cellular base station.

The use of the raycast method for distance determination was tested in a simulated 3D scenario corresponding to terrain images recorded from a height of

60 meters. The accuracy of geolocation of a point in space with a location error of 1 m was achieved using 150 iterations of raycast (Paulin et al., 2021). The achieved results were an incentive for further development of the method and research with real data.

A distance estimation using the bounding box of the detected object from the monocular camera, DistNet (Haseeb et al., 2018), is an approach in which authors used the object's bounding boxes resulting from the YOLO object classification, processed to calculate the features and bounding box parameters. The ratios of the object bounding box dimensions to the image dimensions Bh, Bw, and Bd and the values of average height, width, and breadth Ch, Cw, and Cb of an object of the particular class are input features. DisNet's input layer consists of 6 neurons corresponding to 6 features, followed by 3 hidden layers with 100 parameters. The output layer consists of a single neuron. The output of this node is the estimated distance between the camera and the object viewed with the camera. The training of DisNet is performed using a set of RGB images collected from a railway scene with possible static obstacles on the railway track.

The authors (Zhu & Fang, 2019) introduced the framework to directly predict distances (in meters) from a given RGB image and object bounding boxes. The enhanced model contains 4 parts: a feature extractor, a keypoint regressor, a distance regressor, and a multiclass classifier. The model only uses camera projection matrix P, keypoint regressor, and classifier for training. The feature extractor, the keypoint regressor, the distance regressor, and the classifier are trained simultaneously. To construct the train/test dataset, the authors appended the ground truth of the object-specific distance and keypoint to object detection labels of the training samples of the KITTI/nuScenes(mini) dataset (because only they contain ground truth labels), together with the RGB images. Generated ground truth object-specific distances are varied from 0 to 80 m for KITTI and from 2 to 105 m for the nuScenes(mini) dataset. For the evaluation metrics, absolute relative difference, squared relative difference, the root of mean squared errors, and root of mean squared errors computed from the log of the predicted

distance and the log ground truth distance were used. The results demonstrated that the base model could predict distances with superior performance over alternative IPM (Inverse Perspective Mapping algorithm) and SVR (Support Vector Regressor) approaches, while the enhanced model obtained the best performance over all methods compared.

Table 1. Chronologically sorted overview of papers reporting their best results and flight altitude.

Author	**Method**	**Determined distance error (m)**	**Shot height (m)**	**Reliability % (Shot height / Distance error)**	**UAV model**
(Conte et al., 2008)	Geolocation of ground targets using aerial image registration	2.3	70	30.43%	PingWing, in-house developed fixed-wing MAV
(Leira et al., 2015)	Earth ellipsoid model + Camera calibration and distortion model	7.8	50 - 100	6.4-12.8%	X8 Skywalker fixed-wing
(Zhao et al., 2019)	Earth ellipsoid model (Passive geolocation method)	5	100	20%	DJI M100
(Paulin et al., 2021)	Distance determination using raycast	1	60	**60%**	Simulated
(Pan et al., 2023)	Earth ellipsoid model + Learning-based corresponding point matching model	MAE X 0.046 Y 0.044 Z 0.165	2.8	-	Laboratory product

Of all the mentioned works, only a small number specify the flight altitude and the achieved result as an error in determining the geolocation in meters (Table 1). (Pan et al., 2023) achieves the best result, but only in laboratory conditions and at a flight altitude (2.8 m) that is not suitable for use in actual SAR missions. (Conte et al., 2008) achieves an excellent result that would be applicable in actual flight conditions, but for use in non-urban SAR missions, it is limited by requiring georeferenced images because of the continuous change in the appearance of the terrain due to the change of seasons and vegetation growth. Both (Zhao et al., 2019) and (Leira et al., 2015) achieve similar results, but the former for flat terrain exclusively and the latter using a thermal camera. Our previous method for distance determination based on the raycast method (Paulin et al., 2021), overcomes all the listed limitations for use in a SAR scenario in a simulated

environment and achieves the best result with a reliability of 60% (error of 1 m on recording from a 60 m height).

Computer vision techniques can help operators detect a person or object in an image or video. Today, deep learning methods, specifically convolutional neural networks (CNN), are the most often used to detect objects in images. CNN-based object detection methods are usually divided into two-stage and single-stage detectors. In general, single-stage detectors achieve higher inference speed but with lower precision. As a prominent example of the two-stage detector, Faster R-CNN (Ren et al., 2017) uses a region proposal network to create boundary boxes and utilizes those boxes to classify objects. Its derivatives, such as R-FCN (Dai et al., 2016) and Mask R-CNN (K. He et al., 2017), are proposed to improve detection accuracy further. The single-stage detectors discard the phase of generating proposals and detect objects in a dense manner, e.g., YOLO (Bochkovskiy et al., 2020; Redmon et al., 2016; Redmon & Farhadi, 2017, 2018) and SSD (Liu et al., 2016). YOLO and SSD have adopted a lightweight neural network as a backbone to obtain faster inference speed with state-of-the-art comparable accuracy. Networks such as VGG (Simonyan & Zisserman, 2015) or MobileNet (A. Howard et al., 2019; A. G. Howard et al., 2017; Sandler et al., 2018) pre-trained on the ImageNet (Russakovsky et al., 2015) or OpenImages (Kuznetsova et al., 2020) dataset, are most commonly used as backbones. YOLOv4, used in our experiments, uses CSPDarkNet53 (Wang et al., 2020) as the backbone. To the basic DarkNet53, a deep residual network with 53 layers, CSPNet (Cross Stage Partial Network) was added. Also, the authors of YOLOv4 added Spatial Pyramid Pooling (SPP) (K. He et al., 2015) as a neck to increase the receiving (receptive) field without causing a decrease in inference speed performance. YOLO divides the image into a grid of dimensions S×S, each cell providing frames for the object. The probability, which is calculated for each frame, tells us how confident the model is when there is an object inside the frame and how confident it is in the accuracy of the bounding box. From 2015 to today, YOLO has developed into one of the key models for real-time object detection (Li et al., 2022; Wang et al., 2023). The current version of YOLOv8 (Jocher et al., 2023) provides 5 scaled versions: YOLOv8n (nano), YOLOv8s (small),

YOLOv8m (medium), YOLOv8l (large), and YOLOv8x (extra large). YOLOv8 supports multiple vision tasks such as object detection, segmentation, pose estimation, tracking, and classification. Evaluated on the MSCOCO dataset's test-dev2017 subset, YOLOv8x achieved an Average Precision (AP) of 53.9% with an image size of 640 pixels.

Recently, as an alternative to CNN, there has been increased interest in using Vision Transformer (Dosovitskiy et al., 2021) in various computer vision tasks. Transformer models are originally proposed for natural language processing and use self-attention mechanisms. In computer vision tasks, the transformer model represents an input image as a series of image patches, like the series of word embeddings used when using transformers to text, and directly predicts class labels for the image. The transformer architecture has been shown to be a promising option for computer vision tasks as it requires significantly less computing resources than CNNs. However, it requires huge data sets to train the model to achieve the same performances as today's CNN models (Liu et al., 2023).

The performance of object detection methods significantly depends on the use case and application and the set of images on which they are applied, so different algorithms may prove to be the most successful in different cases. One of the benchmarks for comparing detection methods is the Microsoft COCO dataset (Lin et al., 2014), where different models are typically evaluated by the Mean Average Precision (MAP) metric and the inference time (Frames per Second). The results of object detection on the COCO set achieved by all relevant models are shown on the COCO Detection Leaderboard page (COCO - Detection Leaderboard, n.d.) and show that top detectors reach a mean accuracy of 56%. Successful detection also depends on the size of the object in the image because it is more difficult to detect objects that occupy only a few pixels than those that are in the foreground and occupy a large area of the image, so the detection results, in this case, range from 41% for small objects (APS) to 72% for large objects (APL).

For the use of detectors in various applications that require decision-making in real-time, an important parameter is the inference time, which should be as short

as possible and is measured in FPS (Frames per Second), that are better the higher they are. A comparison of the performance of baseline real-time object detectors trained from scratch on the COCO train 2017 dataset with the same settings is given in (Wang et al., 2023), and here only models relevant to our experiment are presented (Table 2).

Table 2: Comparison of baseline real-time object detectors.

Model	#Param.	Size	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv4 (Bochkovskiy et al., 2020)	64.4M	640	49.7%	68.2%	54.3%	32.9%	54.8%	63.7%
YOLOv4-CSP (Wang et al., 2021)	52.9M	640	50.3%	68.6%	54.9%	34.2%	55.6%	65.1%
YOLOv7 (Wang et al., 2023)	36.9M	640	51.2%	69.7%	55.5%	35.2%	56.0%	66.7%

A graphical presentation of the comparison of real-time object detectors performances considering inference time is shown on page Real-Time Object Detection on COCO (Real-Time Object Detection on COCO, n.d.), Fig. 1.

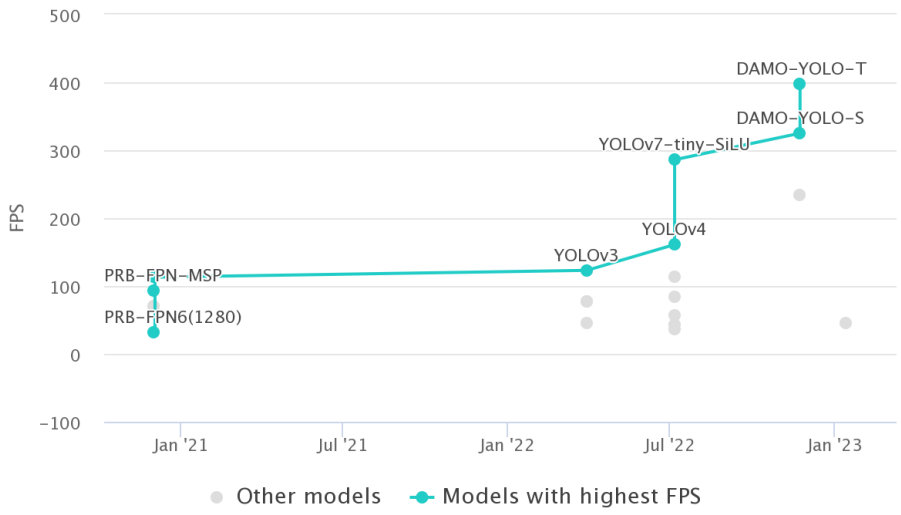


Figure 1: The state-of-the-art object detectors by Frames per Second (FPS) considering the development timeline. The leading computer vision algorithm for real-time object detection on COCO can process 400 frames per second (DAMO-YOLO-T).

3. Prototype of a system for automatic person detection and geolocation in search and rescue missions

We propose a prototype of a system for automatic person detection and geolocation in search and rescue missions (SAR-DAG) specifically designed to automatically detect and geolocate missing persons by using UAVs in SAR missions. The system consists of 3 units: data acquisition, detection module, and geolocation module (Fig. 2).

Satellites and drones participate in the data acquisition phase. Satellites periodically record digital elevation maps (DEM) and supply the drone with GPS data during flight. The drone flies over the search area, performs online detection, and captures images and metadata. Metadata includes the position and orientation of the drone and camera data. Images and metadata are recorded on an SD memory card and become available for offline processing upon the drone's return to base.

The detection module is used in the phase of offline data processing for analysis of images taken during drone flight and automatic detection of persons in the images using a deep neural network model that has been previously trained and fine-tuned for the detection of injured and missing persons.

As part of the geolocation module, the resulting detection data is combined with the positional data of the drone in a single CSV file. The DEM of the terrain over which the drone was flown is used to generate the 3D terrain. By using 3D terrain and the data in the CSV file, the raycaster can determine the geolocation of the detected person. Geolocation information is forwarded to the rescue team, who accesses the person on the ground and successfully completes the SAR mission.

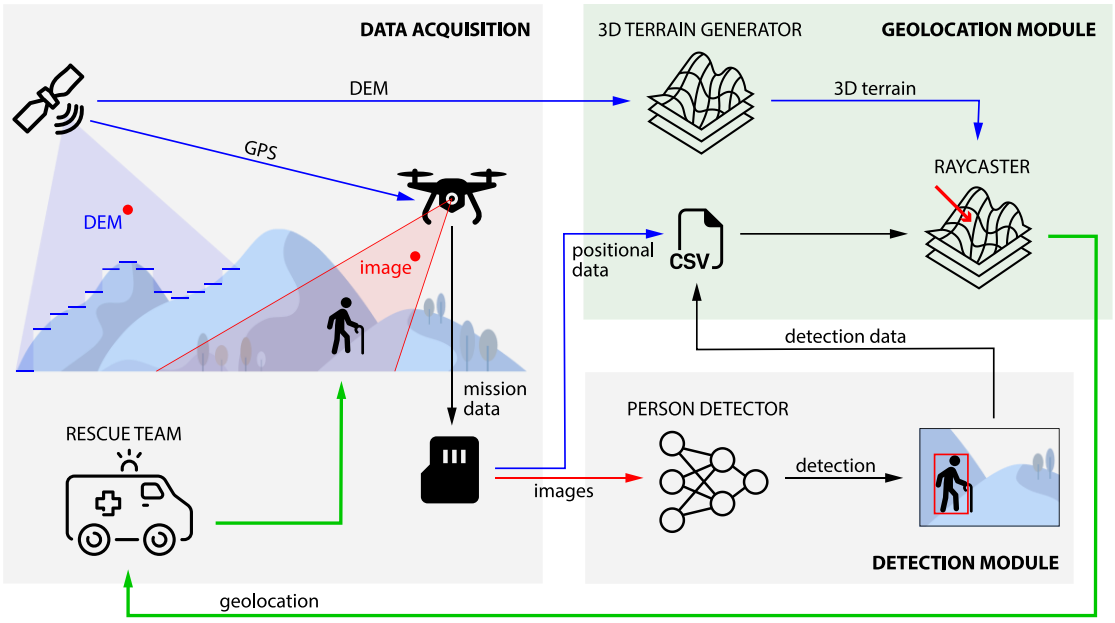


Figure 2: Search and Rescue - Detection and Geolocation (SAR-DAG) system overview.

3.1. Data acquisition

The images obtained during the drone flight are used to create datasets for training the person detector. With each image taken by a drone, a series of metadata is also recorded, such as the GPS location of the drone from which the image was taken and the telemetry of the drone and camera. This metadata, together with the drone's camera optics specification, is used when determining the distance of the person detected in offline mode.

In order for the person detector to be specially prepared to facilitate the detection of missing or injured persons in non-urban areas, it is necessary to additionally train the detector on precisely those images that correspond to real scenes and situations that could occur in search and rescue operations. This actually means that the dataset on which the detector is trained should include footage that simulates different positions of injured individuals located in challenging terrains, such as inaccessible areas, hills, forests, etc. It should also include different lighting conditions, different drone heights, and scenarios where people may be partially obscured by tree branches or placed in the shade of trees. These

different scenarios in search and rescue operations should include different activities and poses, such as walking, running, sitting, and others.

We have prepared the SARD dataset (Sambolek & Ivasic-Kos, 2021) with footage on challenging and inaccessible terrains and simulations of various poses of injured individuals found in non-urban areas. The dataset is additionally augmented by adding different weather conditions. Images containing persons are manually annotated with the bounding box marked around each person in the image to be used later as ground truth (GT) for training or testing the person detectors (Fig. 3).

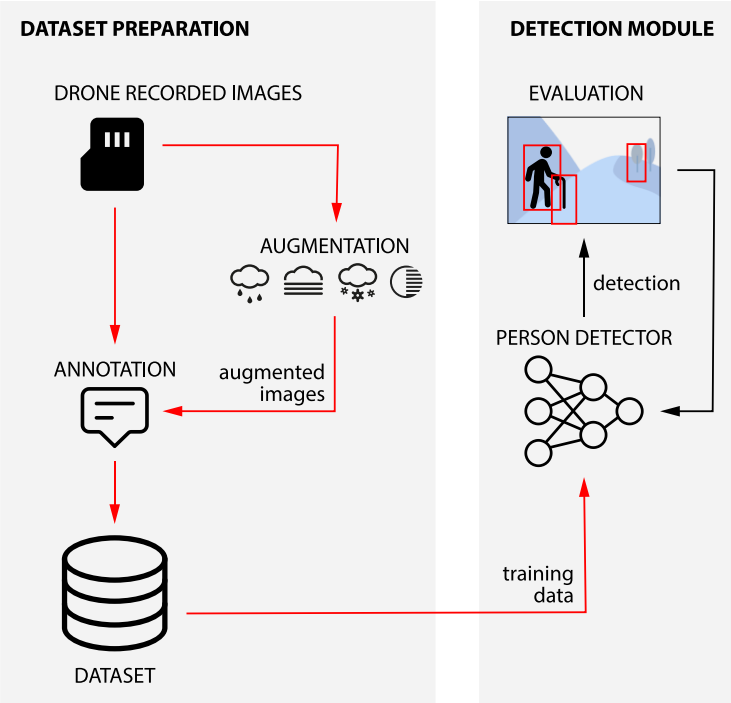


Figure 3: Dataset preparation for training and evaluating deep neural network person detector model.

3.2. Detection module

The detection module is able to perform object detection during the flight, in online mode, and after the flight, in the offline mode, but for this research, we are using it solely for offline detection. We have in detailed explained the process of training the deep neural network object detector for person detection in SAR missions in (Sambolek & Ivasic-Kos, 2021), and here we will only point out the most important

characteristics of this module so that the functionality of the proposed system can be understood more easily.

The deep learning method was used to train a person detector on drone imagery acquired in non-urban areas during search and rescue operations. Here we use the YOLOv4 model pre-trained on the MS COCO dataset (Lin et al., 2014), and the model intended for the detection of people in the SAR missions additionally trained and fine-tuned on the SARD dataset as in the original work (Sambolek & Ivasic-Kos, 2021). The YOLOv4 model was trained using a batch size of 64 and a subdivision of 32, with a total of 6000 iterations. The training process employed a learning rate of 0.001, momentum of 0.949, and decay of 0.0005. The width and height were set to 512 px for the network resolution. For testing purposes (person detection), the same environment was used as in (Sambolek & Ivasic-Kos, 2021), Dell G3 i7-9750H CPU, 16 GB RAM, GeForce GTX 1660 Ti 6 GB.

3.3. Geolocation module

In this paper, we focus on the last required component of the SAR-DAG system, the geolocation module, by using the proposed geolocation method based on the raycast (Section 4) and evaluating its potential for use in real-world SAR scenarios (Section 5).

4. Proposed geolocation method based on the raycast

Raycasting is well-known in computer graphics, being around since 1982 (Roth, 1982). It is a process of tracing geometric rays from the camera and finding line-surface intersections (Fig. 4). It has been invented as the methodological basis for a CAD/CAM solid modeling system. In other words, raycast is the ray-solid evaluator, which finds where a given ray enters and exits a given solid. Here, the solid is assumed to be a 3D terrain.

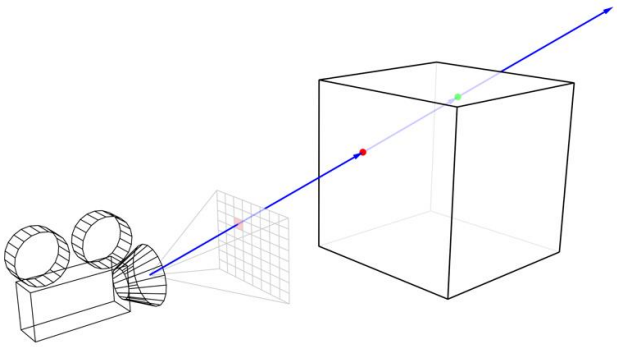


Figure 4: Ray is projected from the camera (on the left) through the point on the screen (pale red square) to the object (cube) in 3D space. The point of intersection is where the ray hits the object's surface (red dot). Ray may continue through the object, exit (green dot), and then continue further, trying to hit another surface in the scene.

Besides 3D terrain, input in the form of a digital elevation map, meshed LiDAR point cloud, or sculpted 3D model, the proposed geolocation method based on the raycast requires a sequence of monocular aerial images, known specification of drone's camera optics, drone, and camera telemetry for each recorded image, a dedicated neural network for the object (person) detection, and normalized device coordinates of the center of the bounding box of the detected person.

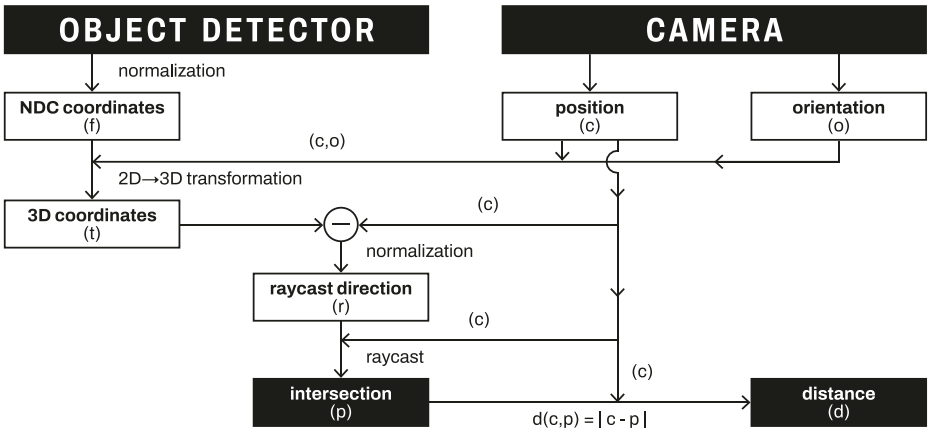


Figure 5: Diagram of the raycasting part of the proposed geolocation method.

The method takes normalized device coordinates (NDC; **f**, ranging from 0 to 1 in screen space) from the detector's output and transforms them to 3D coordinates (**t**) using the camera's position (**c**) and orientation (**o**) from the image's metadata (Fig. 5). Then it subtracts these 3D coordinates from the camera's position to obtain a raycast direction vector (**r**). When shot from the camera's position to 3D terrain, this vector gives us a point of intersection (**p**) whose 3D coordinates are

exact geo-coordinates where a person is located (Fig. 6). Calculating the distance (**d**) to the detected person is then a simple task of calculating the distance between the camera position and the point of intersection (Eq. 1):

$$d(c, p) = |c - p| \tag{1}$$

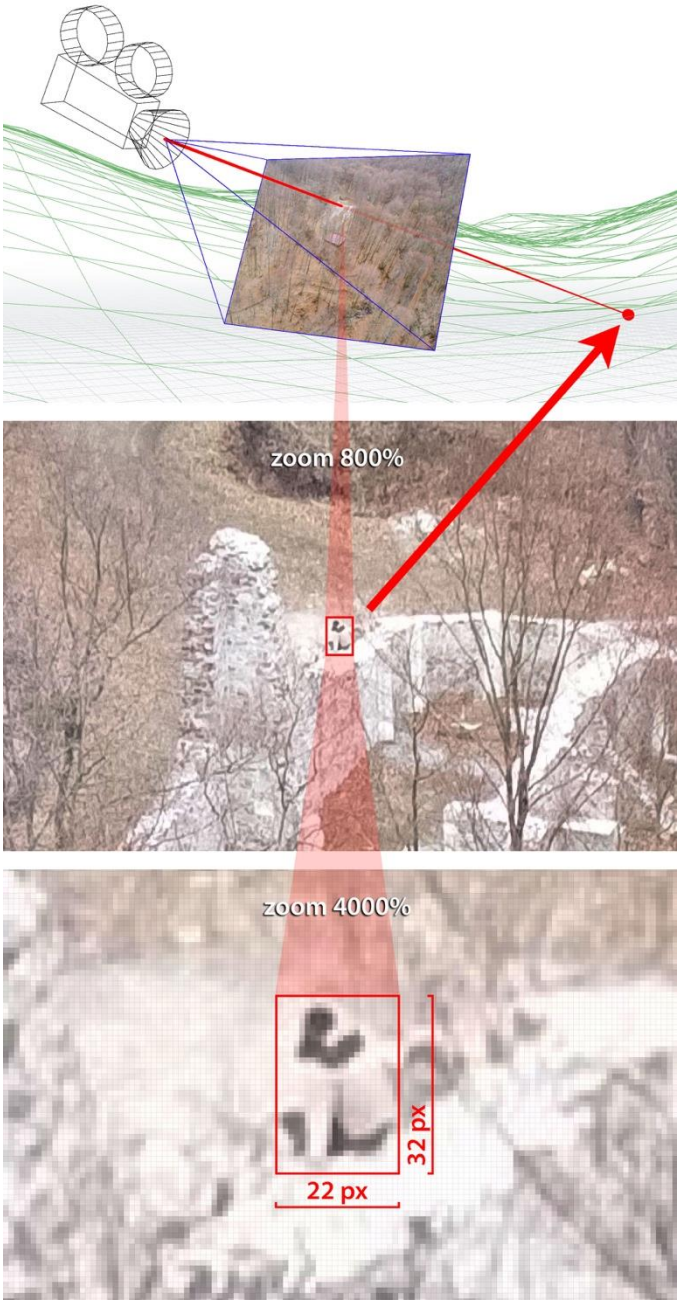


Figure 6: Top: The drone's virtualized camera with a visible 4:3 aspect ratio frustum

(blue) casts a single ray (red) thru the person detector's NDC coordinates to the 3D terrain (green). The red dot indicates a point of intersection (p): a person's location in the 3D space that can be easily remapped back to GPS coordinates. Middle: 800% zoomed 4864×3648 px image with visible detection bounding box (red) covering 22×32 px area with a sitting person inside. NDC coordinates are located in the center of this bounding box. The red arrow points to NDC coordinates projected on the terrain. Bottom: 4000% zoomed image with the pixelated person.

To successfully apply the proposed geolocating method based on raycast for use in SAR missions, Algorithm 1 should be followed. First, the tools for 3D terrain generation and raycasting need to be developed. Then, using a 3D terrain generator, a specific 3D terrain is generated for each SAR mission while acquiring mission data (images captured by the drone and the drone's telemetry). For each image in mission data, person detection is performed using pre-existing person detector. Also, for each image, drone telemetry and image metadata are extracted from the drone's log and image. After processing all images, detection data, telemetry, and metadata are combined into a single CSV file. This file is further processed with a raycasting tool and used in a preprocessing step, where the flight path is divided into individual flight sequences. The algorithm's core is geolocating the detections for which the raycast is used. And finally, sequence and flight results are processed, and the precise GPS coordinates of the target (e.g. injured person) are determined and forwarded to the rescue team.

Algorithm 1. Proposed geolocation method algorithm.

```
create 3D terrain generator
create raycasting tool
for each SAR mission
    generate 3D terrain using 3D terrain generator
    acquire mission data using drone
    for each image in mission data
        perform person detection using pre-existing person detector
        extract drone telemetry from drone log
        extract image metadata from image
    combine detection results, telemetry, and metadata to Input CSV file
    with raycasting tool
        process Input CSV file
        preprocess flight path data
        geolocate detections
        process sequence and flight results
```

4.1. 3D terrain generator

The 3D terrain model has an important role in the proposed geolocation method. Its accuracy (compared to the actual terrain), along with the accuracy of the telemetry data of the drone, dictates whether the projected ray will intersect it in the appropriate location and thus allow reading the exact GPS coordinates of the detected person. If the geometry of the 3D terrain deviates from the geometry of the actual terrain, which is often the case due to the relatively low resolution of the 3D terrain, the read coordinates will contain the deviation (calculated Error_2D and Error_3D, which represent the distance from the target to the ground truth, GT, expressed in meters, will be greater than zero).

A limiting resource for creating 3D terrain models is the availability of elevation data and their resolution (m/px). Our terrain generator uses two types of input data: metadata of the selected OpenStreetMaps area and NASA elevation data (1-arcsecond N45E016.hgt tile, resolution 30 m/px which equals 3601×3601 px) in a latitude/longitude projection (EPSG: 4326 (*EPSG:4326*, n.d.)) from the Shuttle Radar Topography Mission (SRTM), downloaded using 30-Meter SRTM Tile Downloader (*30-Meter SRTM Tile Downloader*, n.d.).

From OpenStreetMaps data (*OpenStreetMap > Export*, n.d.), which we download using the Overpass API (*Overpass API*, n.d.), we use metadata from which we build 3D objects and 3D roads and rails infrastructure. These 3D objects are not used for raycasting, but they significantly help with orientation when evaluating raycasting results, given the monotony of the display of the 3D terrain itself (Fig. 7). In addition, they can be used for automatic pathfinding so that when determining the location of the detected person, an access route proposal is created. The proposal might include the shortest route to the nearest road, considering different terrain properties included in the data that we used to generate 3D terrain.

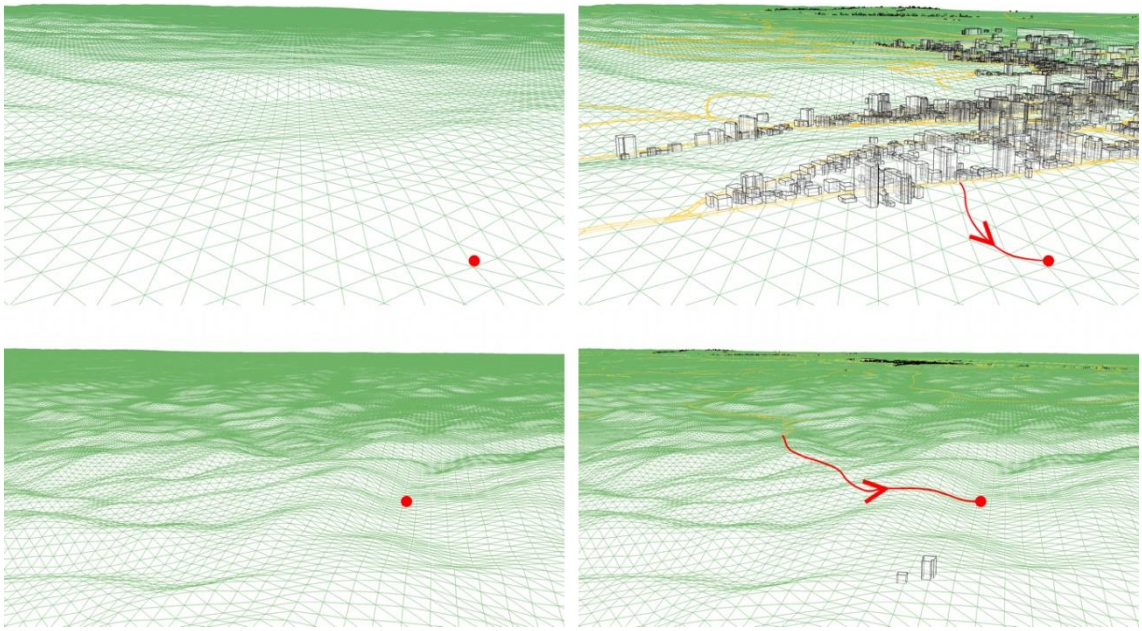


Figure 7: Top-left: monotonous 3D terrain (near the city) that is difficult to navigate without reference 3D geometry. The red dot represents the location of the detected person. Top-right: 3D terrain with additional 3D geometry (buildings and traffic infrastructure). The automatically generated red curve shows the shortest and safest route from the nearest road to the detected person. Bottom-left: monotonous 3D terrain (wilderness). Bottom-right: even sparse geometry (2 structures at the mountain peak) and contours of the city (black) far away help in orientation.

The generator allows us to vary the output resolution of the 3D terrain model. The output of the 3D terrain generator is a triangulated polygonal mesh that contains data on the boundaries of the geographical bounding box.

4.2. Raycaster tool

The raycasting tool is the heart of the geolocation module. It transforms and projects transformed 2D coordinates obtained from the object detector onto 3D terrain, finding the point of intersection whose 3D coordinates are exact geo-coordinates of a person's location in the real world.

Before creating the raycasting tool, its input (Table 3 and 4) and output (Table 5) formats are defined. Today's drones record a large amount of metadata in their logs with each image they capture during the flight. Their subset (Table 3), which consists of data related to the drone trajectory and identification of captured

images at individual flight points (Table 3, #1-12, and #25-28), is sufficient for applying the proposed geolocation method. We refer to this subset as "Flight log".

Table 3. Flight log variables, their description, and example data with corresponding units.

#	Variable	Description	Example data	Unit
1	Time	Flight point's timestamp	2022-01-18 14:28:44	-
2	File_Name	Filename of the image recorded in the flight point	DJI_0265.JPG	-
3	GPS_N	Latitude of the flight point	45.51064608	degrees
4	GPS_E	Longitude of the flight point	16.76035542	degrees
5	Absolute_Altitude	Drone's absolute altitude (unreliable)	30.82	m
6	Relative_Altitude	Drone's altitude relative to the take-off point's height	29.7	m
7	Flight_Pitch_Degree	Drone's pitch	-3.8	degrees
8	Flight_Yaw_Degree	Drone's yaw	-11.7	degrees
9	Flight_Roll_Degree	Drone's roll	-0.6	degrees
10	Gimbal_Pitch_Degree	Camera's pitch	-44.3	degrees
11	Gimbal_Yaw_Degree	Camera's yaw	-11.7	degrees
12	Gimbal_Roll_Degree	Camera's roll	0	degrees
13	GT_CX	Horizontal center of GT bounding box	1620	px
14	GT_CY	Vertical center of GT bounding box	3068	px
15	DET_CX	Horizontal center of detection's bounding box	1612	px
16	DET_CY	Vertical center of detection's bounding box	3068	px
17	GT_X_min	Upper-left X coordinate of GT bounding box	1506	px
18	GT_Y_min	Upper-left Y coordinate of GT bounding box	2912	px
19	GT_X_max	Lower-right X coordinate of GT bounding box	1735	px
20	GT_Y_max	Lower-right Y coordinate of GT bounding box	3224	px
21	DET_X_min	Upper-left X coordinate of detection's bounding box	1510	px
22	DET_Y_min	Upper-left Y coordinate of detection's bounding box	2911	px
23	DET_X_max	Lower-right X coordinate of detection's bounding box	1715	px
24	DET_Y_max	Lower-right Y coordinate of detection's bounding box	3226	px
25	Drone_Model	Drone model designation	Phantom4A	-
26	Camera_Model	Camera model designation	FC6310	-
27	Resolution_X	Recorded image's width	5472	px
28	Resolution_Y	Recorded image's height	3648	px
29	FOV	Camera's diagonal field of view	84	degrees
30	Focal_Length	Camera's focal length	8.8	mm
31	Wind	Wind speed	3	m/s
32	Wind_Direction	Wind direction	SE	-

The drone's absolute altitude (#5) is not necessarily reliable. If this measurement during the raycaster setup proves unreliable, the absolute altitude of the drone should be calculated from the drone's relative altitude (relative to the home point's altitude). The home point is the place from where the drone took off. Data related to the bounding box of detected objects (#15-16 and #21-24) is obtained as the output of the person detector. Variables #13-14 and #17-20 are reserved for storing manually annotated ground truth, which is required for testing the raycaster's implementation. In both cases, if detection is not present in the image or if the GT is not manually added, -1 is used instead of the coordinate.

Data related to camera optics (#29-30) is taken from the technical specifications of each drone (identified by #26). Wind data (#31-32) is registered before taking off and, if the drone has no anemometer, refers to the wind measured at the ground level. Wind at the flight altitude can deviate significantly from this measurement, so the possibility is left to be treated as a variable and recorded with each image, using a drone's anemometer like the one built into the MATRIX 300 RTK (*MATRICE 300 RTK User Manual, n.d.*). If the wind is not measured, -1 is registered (both for wind speed and direction).

Recording the data combination #25-30 allows for varying the settings of the drone camera optics during the flight, as well as the image resolution and aspect ratio. For the experiments, fixed focal length (8.8 mm), FOV (84°), and resolutions of 5472×3648 px (aspect ratio 3:2) and 4864×3648 px (aspect ratio 4:3) were used.

Flight log data is used to create an Input CSV file (Table 4). Input CSV file consists of a header row with variable names and Flight log data for the Home point, N data points, and optional GT required for testing the raycaster's implementation. A single flight point can have multiple detections (in a single image). Therefore we store Flight log data for each detection separately and refer to it as a "data point" in the Input CSV file.

Table 4. Input CSV file structure.

Row	Content	Type of content
1	Header	Variable names
2	Home point	Flight log data

3...N	Data points	Flight log data
N	GT	Flight log data

Based on the defined input and output data formats, we developed a raycasting tool, using, as in the case of the 3D terrain generator, procedural 3D modeling. We chose this approach because it allows rapid prototyping with immediate visual feedback. We can monitor the performance of each scenario over the entire observed area (3D terrain) but also from the perspective of a virtualized drone camera, making it easier to spot problems and adjust tools faster (Fig. 8).

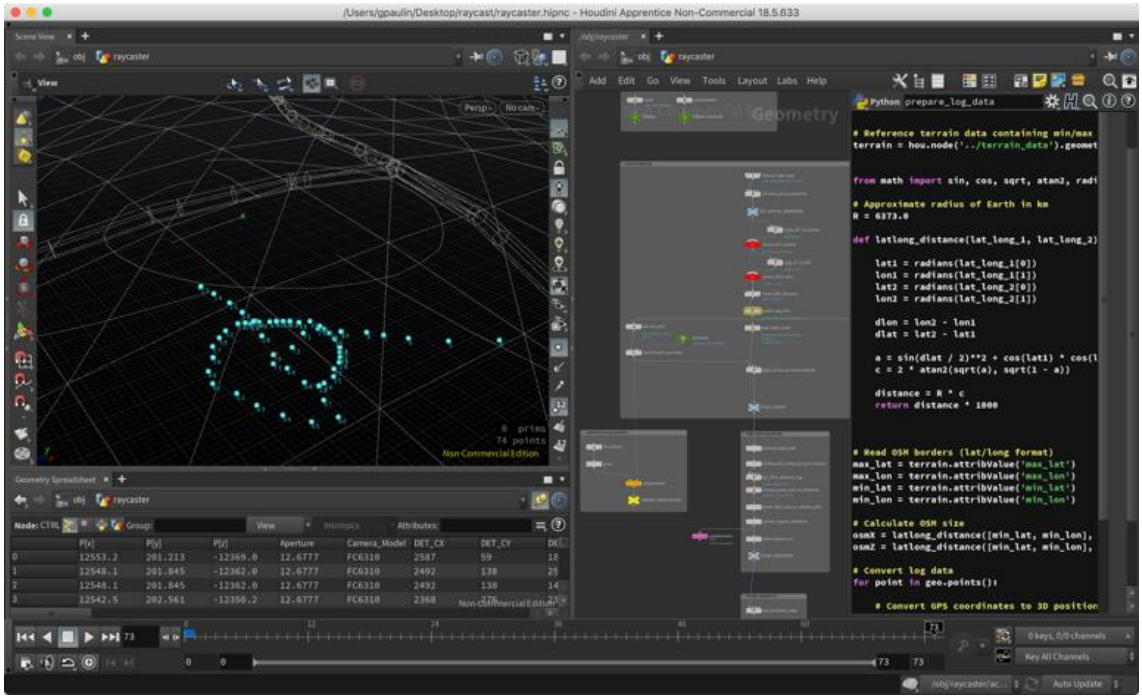


Figure 8: Raycaster implementation in Houdini with visible Scene View (upper-left: showing 3D terrain, flight path, drone's home point, GT, and multiple detections), Geometry Spreadsheet (bottom-left: showing numerical results of the raycasting), procedural nodes (middle area) and Python code of one of the nodes (right).

Raycaster sequentially performs input CSV processing, flight path data preprocessing, geolocating detections, and sequence and flight results processing.

4.2.1. Input CSV processing

In the first step of Input CSV processing, the pre-known (according to drone/camera specifications) diagonal drone camera's FOV is converted to a horizontal FOV, using image dimensions, at each flight point. This data, combined with the known focal length, is used to calculate the aperture value of the 3D camera, which aligns the optics of the drone camera and the 3D camera (Fig. 9).

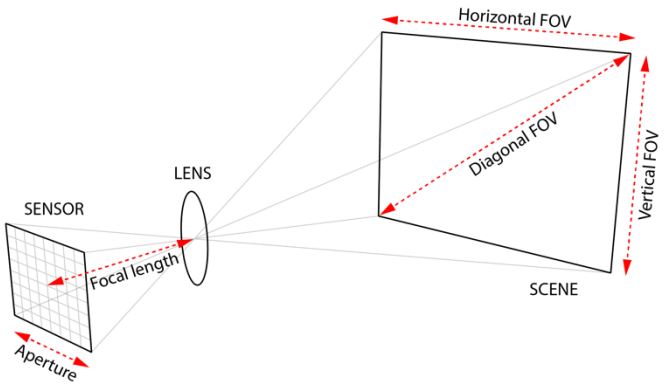


Figure 9: Relevant elements for camera optics alignment.

In the second step, using the geographical bounding box boundary data stored in the previously generated 3D terrain, the GPS coordinates of the drone trajectory are converted to 3D coordinates. At the same time, the YOLO detection coordinates are translated into NDC coordinates (normalization is performed, and the Y coordinates are mirrored from the 4th to the 1st quadrant).

From the data thus prepared, the home point location is taken, and a raycast is performed by which its XZ position is projected onto the 3D terrain. The home point's absolute height (Y) is read from the intersection point. The optional GT (actual location of the target) is processed in the same way.

In the last step of processing the Input CSV, the drone's altitude (h_{ABS}) at all flight points is corrected. Its known relative altitude (h_{REL}) is added to the home point's height (h_{HOME}) which completes the positioning of the virtual drone in 3D space (Fig. 10).

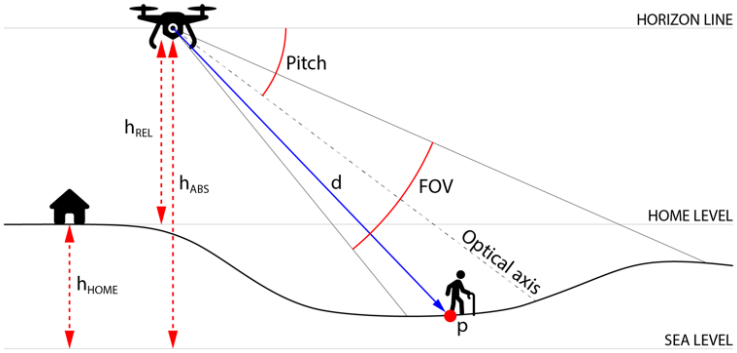


Figure 10: Virtual drone positioned in 3D space with marked camera's optical axis, FOV, and pitch in relation to the horizon line. d represents the distance from the drone to the person we are searching for (located at point p).

4.2.2. Flight path data preprocessing

Since not all flight points contain detections, the raycaster's second task is to divide the flight path into individual flight sequences. A flight sequence contains a series of detections in continuous chronological order.

The first step of this process is to remove flight points that do not contain detections (head and tail trimming). The remaining points in the sequence are then assigned a unique sequence identifier that defines the association of a particular flight point to only one sequence.

In the case of multiple detections in the same image (e.g., a group of people), each detection is registered as a separate data point, treating the group as a single person detected in multiple places (within a single flight point). Future work can improve this by identifying (and subsequently reidentifying) each person and averaging raycast results per unique person per flight sequence.

Although the goal of developing this method was to build a system that can detect an injured person from a single image (detection) in just one flight over the area, initial experiments indicated that single flight point sequences are better ignored when determining location. In all test cases, these were false detections that corrupted the result during averaging. Consistent with this conclusion, the final step of sequencing is to remove sequences containing a single flight point.

4.2.3. Geolocating detections

The raycaster's main task is to determine the location of each detection in 3D space and, during calibration, calculate the error relative to the GT and, incidentally, the distance of the GT from the drone. These operations are performed separately for each flight point that belongs to one of the previously formed sequences.

The previously prepared NDC coordinates are transformed into 3D coordinates, and a raycast vector (**r**) is formed using the drone's position. Raycasting is performed from the camera's position in the direction of the raycast vector, and the point of intersection with the 3D terrain is determined. This point is the location of the detected person in the 3D space. Then XZ (Error_2D) and XYZ distances (3D_Error) from GT are calculated. Error_Difference (Eq. 2) is calculated as the difference between these distances:

$$Error\_Difference = |Error\_2D - Error\_3D| \quad (2)$$

Error_2D ignores the height difference (it behaves like a distance in map reading). Error_3D takes height into account (by calculating the actual distance in 3D space) and can differ significantly from Error_2D if the GT is located, e.g., at the foot of a cliff, and detection is (due to raycast error caused by insufficient 3D terrain resolution or incorrect drone telemetry) located at the top of the cliff.

4.2.4. Sequence and flight results processing

The results of individual sequences are calculated after all individual points within them have been processed. The average pitch, relative height, wind speed, and drone distance from the determined geolocation are calculated for each sequence. The average Error_2D, Error_3D, and Error_Difference of the sequences are not calculated by averaging previously calculated Error_2D, Error_3D, and Error_Differences of all points. Instead, all detection locations within the sequence are first averaged to a single point, and then Error_2D, Error_3D, and Error_Differences are calculated for that point. Sequence processing ends with converting 3D coordinates back to GPS coordinates.

The same procedure is repeated to calculate the average results of all sequences for the entire flight. The obtained results are recorded in Results CSV file (Table 5). Individual flight sequences (#24) are numbered from zero, and the entire flight is marked with -1.

Table 5. Results CSV variables, their description, and example data with corresponding units.

#	Variable	Description	Example data	Unit
1	flight	Filename of the flight's data in Input CSV format	vinograd.csv	-
2	terrain resolution	Resolution of 3D terrain model	30	m/px
3	DET location	Use center (of the detection's bounding box) or bottom-middle (bottom) position as the center of detection	center	-
4	DET	Use detection (DET) or GT (GT) data	DET	-
5	P.x	X coordinate of person location (in 3D space)	12521.66602	m
6	P.y	Y coordinate of person location (in 3D space)	204.705368	m
7	P.z	Z coordinate of person location (in 3D space)	-12317.81055	m
8	Aperture	3D camera's aperture (in Houdini)	13.18559551	mm
9	Average_Distance	Averaged drone to GT distance	34.55657578	m
10	Camera_Model	Camera model designation	FC6310	-
11	Detections	Number of flight points (in a single sequence or during entire flight)	15	-
12	Drone_Model	Drone model designation	Phantom4A	-
13	Error_2D	Averaged 2D distance between detection and GT	7.948086739	m
14	Error_3D	Averaged 3D distance between detection and GT	7.957713604	m
15	Error_Difference	Difference between Error_2D and Error_3D	0.009626865387	m
16	Focal_Length	Camera's focal length	8.8	mm
17	FOV	Camera's diagonal field of view	84	degrees
18	Gimbal_Pitch_Degree	Averaged camera's pitch	-44.29999161	degrees
19	Latitude	Latitude of person location (in real world)	45.51074219	degrees
20	Longitude	Longitude of person location (in real world)	16.76032829	degrees
21	Relative_Altitude	Averaged relative altitude	29.70000458	m
22	Resolution_X	Recorded image's width	5472	px
23	Resolution_Y	Recorded image's height	3648	px
24	Sequence	Single sequence ordinal number (or -1 for entire flight)	-1	-
25	Wind	Averaged wind speed	3	m/s
26	Wind_Direction	Wind direction	SE	-

5. Experiments

The purpose of this series of experiments was to test the proposed geolocation method in different real-world scenarios. The scenarios differed according to the terrain configuration. Kutina (Croatia) was chosen as a location for the experiments due to the availability of terrains of varying difficulty in a relatively small area (approx. 30×30 km), with the possibility of access by car. The selected

micro-locations and their terrain configurations were increasingly demanding, both for the application of the proposed method and for SAR. They varied from flat terrain, over terrain with a slight slope, to mountainous terrain and narrow valley (Fig. 11).

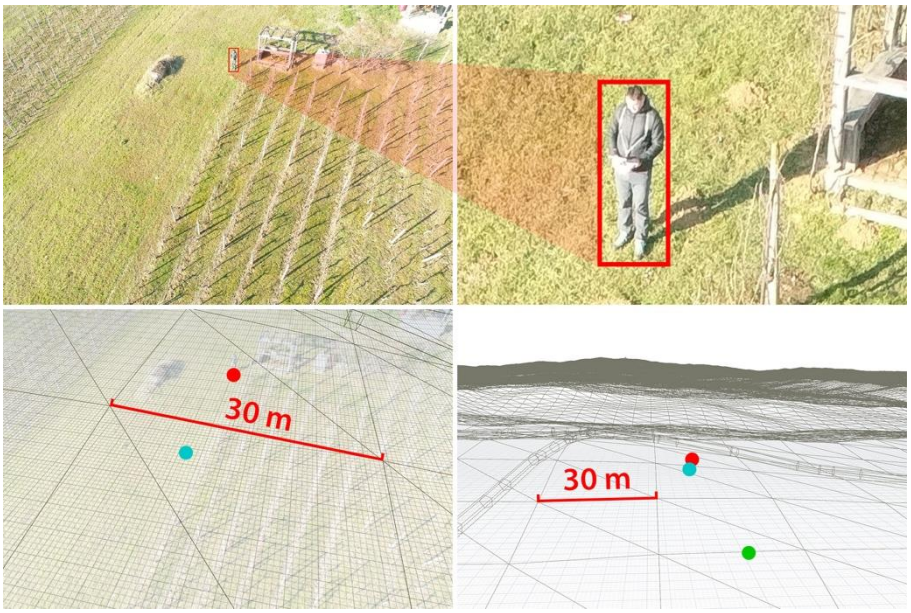


Figure 11: Top-left: example drone imagery from the 2nd experiment (terrain with a slight slope), with visible detection (red bounding box). Top-right: 800% zoomed detection. Bottom-left: generated 3D terrain laid over drone imagery. Each rectangular pair of triangles represents a 30×30 m terrain patch. The red dot indicates the person's position (ground truth), and the cyan dot indicates the raycasted detection's coordinate. Bottom-right: 3D terrain's relief. The green dot indicates the drone's home point.

In the experiments, the hand-operated Phantom 4 Advanced drone (Phantom 4 Advanced - Specs, n.d.) without an RTK module was used in all phases, such as taking images to form a dataset for training and testing the model for detecting persons in SAR missions and for testing the geolocation method. All imagery acquisitions were performed by flying 30 m above the ground. Images were taken at a frequency of 1 image every 2 seconds.

5.1. Dataset for training the person detector

For training the person detector model in the detection module, we have prepared the SARD dataset (Sambolek & Ivasic-Kos, 2021) that is specifically curated to facilitate the detection of missing or injured individuals in non-urban environments

using drone imagery. The dataset includes footage that simulates various poses of injured individuals found in challenging terrains, such as inaccessible areas, hills, forests, and more. It also incorporates different lighting conditions, varying drone altitudes, and scenarios where people may be partially obscured by tree branches or located in the shade of trees. These diverse scenarios encompass search and rescue actions and standard poses like walking, running, sitting, and others.

The SARD dataset includes 1981 images captured with a DJI Phantom 4 Advanced drone at FHD resolution, featuring 6532 annotated instances of people. We divided the dataset into training and testing subsets to assess the model's performance using a 60:40 ratio. Fig. 12 provides a visual representation of a selection of images from the SARD dataset.

In the SARD dataset, there is a greater proportion of small objects than the large ones. The object's size is determined based on the number of pixels within their respective bounding box. According to the COCO standard (COCO - Detection Evaluation, n.d.) there are three object size categories: Small (with an area smaller than 322 px), Medium (with an area between 322 and 962 px), and Large (with an area larger than 962 px). Approximately 29% of the objects in the SARD dataset fall into the Small category, while 64% fall into the Medium category. The remaining 7% of objects belong to the Large category.

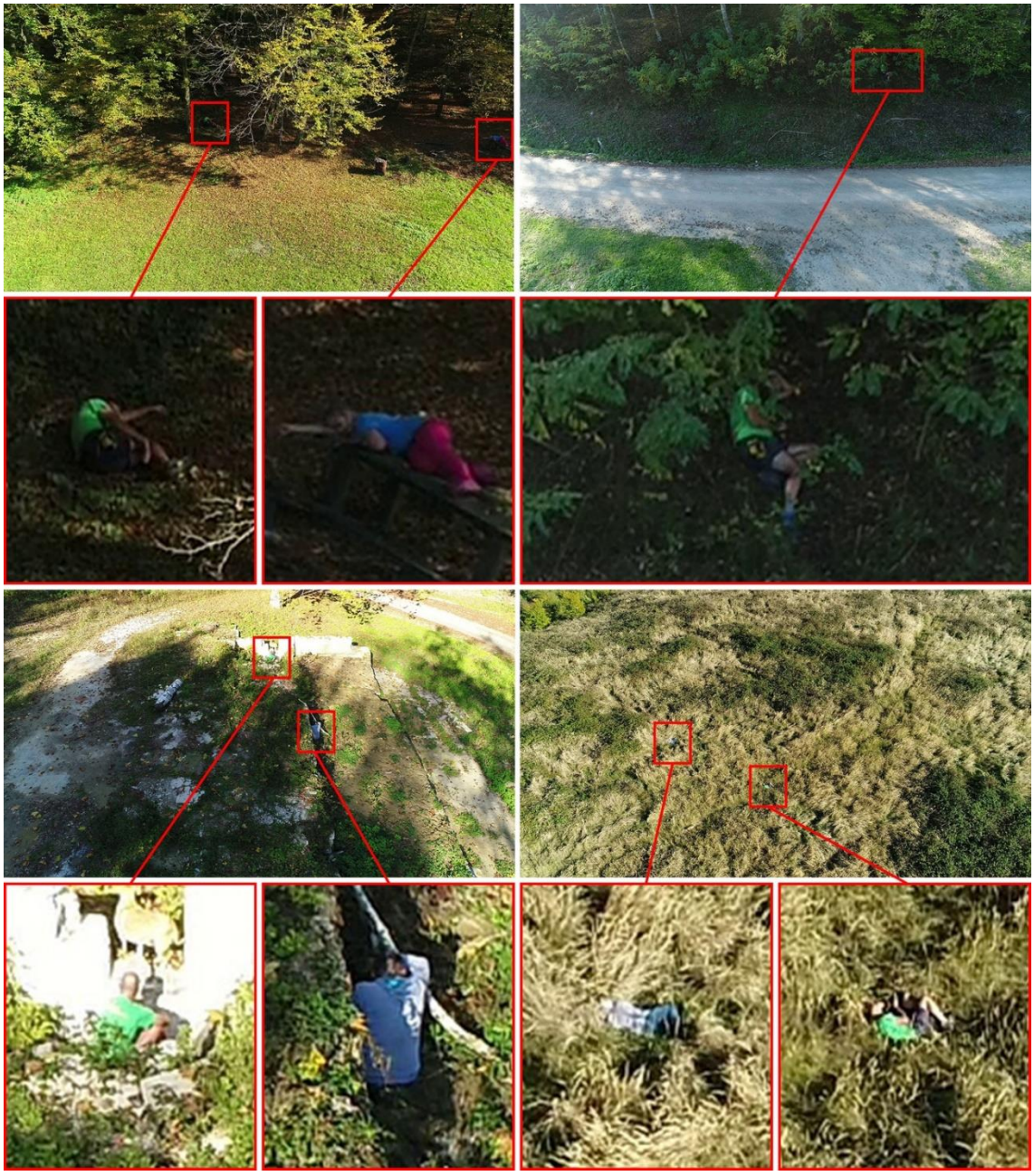


Figure 12: Examples of images from the SARD dataset.

In case of blurred images or challenging weather conditions, the model can be further trained using the Corr-SARD dataset (Sambolek & Ivasic-Kos, 2021). The Corr-SARD dataset is derived from the SARD set and incorporates the effects of snow, fog, frost, and motion blur into the SARD images (Fig. 13).

Figure 13: Examples of images from the SARD-Corr dataset. Top row: ice (left) and fog (right). Bottom row: motion blur (left) and snow (right).

Since this paper focuses on geolocation, the drone imagery for the experiments was recorded in good weather and did not require additional training on the SARD-Corr dataset. An optimized detection model can be used depending on the weather conditions during a particular SAR mission. The geolocation module would work equally well with thermal images, using a model for detecting people in thermal images. The training process can be carried out continuously, using images collected during SAR missions and adjusting the model afterward.

5.2. Datasets used for testing the geolocation method

In 4 conducted experiments, we prepared a total of 7 primary datasets for raycast (Table 6, datasets #1-5 and #7-8) and 2 aggregated (Table 6, datasets #6 and #9), created by combining primary datasets (Table 6, datasets #4+5 and #7+8, respectively).

While preparing these datasets, we eliminated the initial and final parts of the flight that did not contain detections. This way, data processing is sped up, related to the detector and to processing individual Input CSV files as part of the first raycaster's task.

Table 6. Raycast datasets with corresponding statistics.

#	Dataset	Sequences	Flight points	Data points	GT	DET (TP+FP)	TP	FP	FN	ROpti (%)
1	polje	4	72	73	25	25	23	2	2	84.0
2	vinograd	1	15	18	10	15	10	5	0	50.0
3	vinograd2	5	67	86	57	78	55	23	3	55.2
4	Vis_let1	1	12	12	10	1	0	1	10	-10.0
5	Vis_let2	1	9	9	6	1	0	1	6	-16.7
6	Vis_let1+2	2	21	21	16	2	0	2	16	-12.5
7	Brunkovac_let1	2	14	14	10	6	6	0	4	60.0
8	Brunkovac_let2	2	21	21	16	16	15	1	1	**87.5**
9	Brunkovac_let1+2	4	35	35	26	22	21	1	5	76.9

Table 6 shows the number of sequences in each flight, the number of flight points processed (containing at least one detection), the number of data points, the number of available ground truth (GT), the number of detections with YOLOv4-SARD model (true, TP and false, FP), the number of failed detections (FN) and ROpti.

Flight sequences are created during flight path data preprocessing. They consist of a continuous sequence of detections within flight points (the moment during the flight in which the image was taken). Each detection in an individual image is recorded as one data point in Input CSV file (Table 4). In our case, flight sequences contain an average of 30 flight points (median = 21). The small number of flight points is a consequence of the large area covered during the flight in the SAR mission, so there are not many overlaps in the images. For this reason, there are not many (redundant) detections, which, consequently, speeds up the process of raycasting and geolocation.

The SAR-DAG_raycast dataset is available for download at https://urn.nsk.hr/urn:nbn:hr:195:686105.

5.3. Evaluation metrics

5.3.1. Metrics for evaluation of the object detector performances

For evaluating the detection performances of our model, we have used standard evaluation metrics (*COCO - Detection Evaluation*, n.d.) for the average precision (AP) on various IOUs (Intersection over Union) and considering the object size

on the image. It is typically calculated for 10 IOU thresholds ranging from 50% to 95%, with increments of 5%. This range is often reported as AP@50:5:95. Additionally, AP can be evaluated using specific IOU values, most commonly 50% and 75%. These are reported as AP_{50} and AP_{75}, respectively. The AP metric can also be evaluated across different object sizes: AP_S for small objects (with an area smaller than 32^2 px), AP_M for medium size objects (with an area between 32^2 and 96^2 px), and AP_L for large objects (with an area larger than 96^2 px).

When searching for a particular object, any detection that recognizes the object and its location can be considered a positive detection, regardless of the percentage of the IoU between the GT bounding box and the detected bounding box. However, search and rescue purposes require an optimized object detector that makes as few false-positive (FP) detections as possible, as they consume valuable human resources and time. Since human resources are limited in SAR operations, it is essential to use them optimally. Each detection of the person detector is additionally reviewed by the person responsible for checking the detections on the recordings, who then dispatches teams to the field. For this reason, it is crucial to have as few false positive detections as possible not to waste time that can mean the difference between life and death for a missing person. To address this, we evaluate detection performance also with the ROpti (Recall Optimal) metric defined in (Sambolek & Ivasic-Kos, 2021) because ROpti gives us a value that considers all variables necessary for detector evaluation (TP, FP, FN). ROpti is computed as the ratio of the difference between true positive (TP) and false positive (FP) detections, divided by the total number of possible detections (TP+FN) in the dataset (Eq. 3). This metric provides a quantitative measure of the model's performance in terms of minimizing false positives and maximizing true positives, considering the overall potential detections in the dataset. For perfect precision (no false positives), ROpti is equal to recall, and with perfect recall (no false negatives), ROpti is equal to 1, which is a perfect score.

$$ROpti = \frac{(TP - FP)}{(TP + FN)} \quad (3)$$

5.3.2. Metrics for evaluation of the geolocation method

In the experiments, we replicate the conditions of actual use of drones in SAR missions with the camera tilted down (pitch) by -33.7° to -63.8° (relative to the horizon). By taking oblique photographs, we cover a larger angle than when the camera would look perpendicular to the ground (pitch = -90°), taking vertical photographs. The distance of the detected person from the drone can be much higher compared to vertical photographs in which the detection distance in the center of the image corresponds to the drone's height in relation to the ground.

For this reason, in addition to the primary set of measures (Error_2D and Error_3D, which measure the detection distance from GT), we introduce the ErrDist (Eq. 4) measure, which represents the percentage of error of the determined location of the person in relation to the actual distance of the drone from the person.

$$ErrDist = \frac{Error\_2D}{d(c,p)} \quad (4)$$

In real situations, the error (Error_2D) of 50 m does not play a significant role, especially in areas of good visibility (e.g., meadows). However, if the search is performed in thickets or on impassable karst terrain, even a smaller error (Error_2D < 40 m) can be a problem that is then attempted to be solved by looking at a photograph in which detection is visible. Knowing this, we chose ErrDist = 10% for the (rather strict) limit of acceptable error, which means that for a drone 60 m away from the location of the person we are searching (GT), the acceptable result is the one that gives us a location within a radius of 6 m from GT.

5.4. 3D terrain generation

To successfully apply the proposed geolocation method and build the geolocation module for the SAR-DAG system prototype, we developed tools for 3D terrain generation and raycasting using SideFX Houdini Apprentice 18.5.672 (*Houdini Help, n.d.*).

For the experiments, we generated 3D terrain at the maximum available (30 m/px) and lower resolution (100 m/px) to determine the influence of terrain resolution on the detection accuracy. By using 3D terrain in higher resolution, we measured the heights of known elevations and found significant deviations (up to 28 meters, Table 7):

Table 7. Mountain peaks used for measuring the height difference between 3D terrain (based on 30 m/px elevation data) and their actual height.

Mountain peak	Actual height (masl)	Height in 3D (masl)	Error (m)	Latitude, longitude
Vis	444	416	**-28**	45.6001, 16.7558
Humka	489	481	**-8**	45.61409, 16.75402

Since the problem of insufficient accuracy of terrain geometry for detection purposes was noticed when using higher resolution (30 m/px), lower 3D terrain resolution was not used in the experiments. The generated 3D terrain in our case contains 999,740 points and 1,995,480 polygons and data on the boundaries of the geographical bounding box (45.4, 16.6) and (45.7, 17.0) latitude/longitude.

5.5. Raycaster tool management

Raycaster tool management is reduced to parameters sufficient for applying the proposed geolocation method. Input data vary during experiments (Fig. 14).

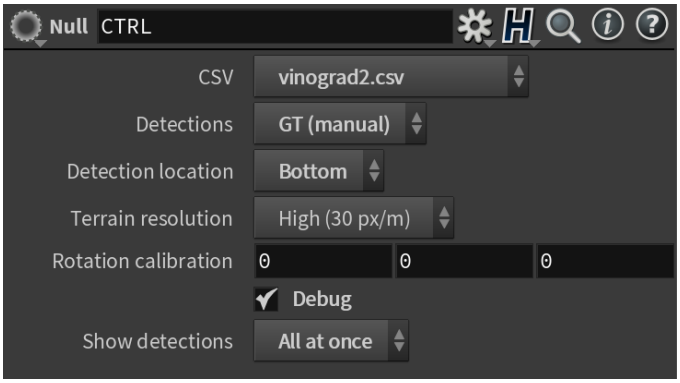


Figure 14: Raycaster's control interface.

The "CSV" parameter selects the flight record (Input CSV file). With the "Detections" parameter, we choose between using YOLO detections or GT. "Detection location" determines the detection coordinate that we target with a raycast (center or middle of the bottom of the bounding box). The center of the bounding box is more suitable for situations where the requested person is lying on the ground, while the bottom of the bounding box is more suitable if we detect a person in an upright position.

The "Terrain resolution" parameter selects the resolution of (previously generated) 3D terrain. The "Rotation calibration" parameter enables subsequent calibration of the drone camera orientation (after the flight is completed) to cancel the errors that occurred during the drone self-calibration. It is possible to adjust all three axes simultaneously (pitch, yaw, and roll) without the danger of a gimbal lock. The "Results" parameter (not visible in Fig. 14 due to the enabled "Debug" option) allows the selection of results: averaged over Sequences or averaged over the entire Flight. Enabling the "Debug" parameter changes the display of results in the 3D viewport from the average coordinates of the localized person (for the entire flight or for a single sequence) to the display of detections at a single flight point. Detections can be displayed as Single (useful when watching a 3D scene from a virtualized drone) or All at once (useful as visual feedback when calibrating the raycaster).

5.6. Results and discussion

5.6.1. Performances of person detector fine-tuned for SAR missions

Table 8 presents the results of person detection on SARD images, specifically focusing on the AP metric. It compares the original YOLOv4 model (referred to as "COCO") with the YOLOv4 model fine-tuned on SARD images (referred to as "SARD"). The "COCO" model was trained on the MS COCO dataset. The results demonstrate a significant enhancement in the detection performances of the "SARD" model for both AP and ROpti. The model trained on the SARD dataset has 37% better AP results than the original "COCO" model (IMP 37.9), and ROpti increased by 57.6%. Specifically, there were 2,611 annotated persons in the testing set, of which the "COCO" model correctly detected 1,068 while detecting 150 false positives. The "SARD" model had 2,512 correct detections and only 97 false positives.

Table 8. Detection results for YOLOv4 model (%).

Train	**Test**	**AP**	**IMP**	**AP_{50}**	**AP_{75}**	**AP_S**	**AP_M**	**AP_L**	**ROpti**
COCO	SARD	23.4	0	40.2	25.3	13.2	26.1	41.1	35.2
COCO + SARD	SARD	**61.3**	**37.9**	**95.7**	**71.1**	**45.0**	**66.4**	**72.6**	**92.8**

Despite using the detector model with high ROpti (93.9%), the average achieved ROpti of all prepared datasets was 50.3%, with a large standard deviation (30.7%) which was mostly contributed by Vis_let1, Vis_let2, and Vis_let1+2 datasets from the 3rd experiment, due to their rigorous conditions (a person in white clothes, in a sitting position, on a white stone).

To simplify the detection problem, a single person in the image was used as the detection target. However, false-positive detections caused multiple detections in some images. We treated such cases by registering each detection in the image as a separate data point during dataset preparation.

5.6.2. Evaluation of the proposed geolocation method

In a previous in silico experiment (Paulin et al., 2021), we simulated a static drone, focusing on measuring the influence of telemetry error on raycast precision in a controlled environment. We concluded that it takes as many as 150 images (raycast iterations) to reduce the detection's location error to less than 1 m (initial

error being 6.26 m and largest ~8 m). These were satisfying results in terms of detection location error but problematic regarding the number of required iterations for application in real-world SAR scenarios. In current real-world experiments, given the drone's movement during which the person we search for enters and exits the frame very quickly, it turned out that we are limited to only 2 to a maximum of 33 images with detections per flight, which converts to the same number of possible raycast iterations. Thus, in the achieved conditions of the experiment, matching those in actual SAR missions, we approached the effort to build a system that can detect an injured person from a single image (detection) in only one flight over the area. Moreover, with the best result (Error_2D = 0.7 m, in a sequence with 4 detections), we found that excellent results can be achieved with a small number of raycast iterations, provided the adequate accuracy of 3D terrain and drone telemetry.

Table 9. Collected results of all experiments.

#	Dataset	DET location	DET/GT	Average_Distance	Detections	Error_2D	Error_3D	Error_Difference	Gimbal_Pitch_Degree	Sequence	Wind	Wind_Direction	ErrDist
1	polje	center	DET	55.14	23	**4.78**	4.78	0.00	-48.15	-1	1	-1	**8.67%**
2		center	DET	46.56	6	**1.88**	1.88	0.00	-51.50	0	1	-1	**4.04%**
3		center	DET	57.06	4	**0.70**	0.70	0.00	-51.50	1	1	-1	**1.23%**
4		center	DET	55.55	7	**8.91**	8.91	0.00	-44.20	2	1	-1	16.05%
5		center	DET	61.41	6	**7.75**	7.75	0.00	-45.40	3	1	-1	12.61%
6	polje	bottom	DET	55.14	23	**4.17**	4.17	0.00	-48.15	-1	1	-1	**7.57%**
7		bottom	DET	46.56	6	**3.06**	3.06	0.00	-51.50	0	1	-1	**6.56%**
8		bottom	DET	57.06	4	**0.86**	0.86	0.00	-51.50	1	1	-1	**1.50%**
9		bottom	DET	55.55	7	**8.47**	8.47	0.00	-44.20	2	1	-1	15.25%
10		bottom	DET	61.41	6	**6.69**	6.69	0.00	-45.40	3	1	-1	10.89%
11	vinograd	center	DET	34.56	15	**7.95**	7.96	0.01	-44.30	-1	3	SE	23.00%
12	vinograd	bottom	DET	34.56	15	**6.57**	6.59	0.02	-44.30	-1	3	SE	19.02%
13	vinograd2	center	DET	32.61	77	**7.47**	7.50	0.03	-48.50	-1	3	SE	22.92%
14		center	DET	48.13	29	**13.77**	13.80	0.02	-38.30	0	3	SE	28.61%
15		center	DET	31.08	33	**23.95**	23.97	0.02	-38.30	1	3	SE	77.06%
16		center	DET	31.62	9	**10.47**	10.54	0.06	-38.31	2	3	SE	33.13%
17		center	DET	26.69	4	**7.39**	7.42	0.03	-63.80	3	3	SE	27.68%
18		center	DET	25.56	2	**6.17**	6.19	0.01	-63.80	4	3	SE	24.16%
19	vinograd2	bottom	DET	32.61	77	**6.56**	6.59	0.03	-48.50	-1	3	SE	20.11%
20		bottom	DET	48.13	29	**11.43**	11.47	0.03	-38.30	0	3	SE	23.75%
21		bottom	DET	31.08	33	**21.78**	21.80	0.02	-38.30	1	3	SE	70.10%
22		bottom	DET	31.62	9	**8.35**	8.40	0.05	-38.31	2	3	SE	26.42%
23		bottom	DET	26.69	4	**5.82**	5.85	0.03	-63.80	3	3	SE	21.82%
24		bottom	DET	25.56	2	**4.77**	4.78	0.01	-63.80	4	3	SE	18.68%
25	Vis_let1	bottom	GT	85.60	10	**41.04**	41.32	0.28	-33.71	-1	5	NE	47.94%
26	Vis_let1	bottom	GT	85.60	10	**3.47**	3.48	0.00	-33.71	-1	5	NE	**4.06%**
27	Vis_let2	bottom	GT	61.59	6	**36.19**	36.22	0.03	-48.90	-1	5	NE	58.76%
28	Vis_let2	bottom	GT	61.59	6	**5.97**	5.97	0.01	-48.90	-1	5	NE	**9.69%**
29	Vis_let1+2	bottom	GT	85.60	16	**4.04**	4.39	0.35	-41.31	-1	5	NE	**4.72%**
30	Vis_let1+2	bottom	GT	85.60	16	**4.68**	4.68	0.00	-41.31	-1	5	NE	**5.47%**
31	Brunkovac_let1	bottom	DET	21.17	6	**11.18**	11.27	0.09	-59.00	-1	4	N	52.80%
32		bottom	DET	19.40	2	**17.89**	18.02	0.13	-59.00	0	4	N	92.22%
33		bottom	DET	22.95	4	**22.81**	22.82	0.01	-59.00	1	4	N	99.40%
34	Brunkovac_let1	bottom	GT	21.56	10	**10.68**	10.80	0.12	-59.00	-1	4	N	49.55%
35		bottom	GT	22.89	5	**19.99**	20.15	0.16	-59.00	0	4	N	87.34%

36		bottom	GT	20.23	5	**19.96**	19.97	0.01	-59.00	1	4	N	98.66%
37	Brunkovac_let2	bottom	DET	27.91	16	**16.73**	16.98	0.25	-48.87	-1	4	N	59.96%
38		bottom	DET	17.98	10	**17.46**	17.46	0.00	-57.00	0	4	N	97.10%
39		bottom	DET	45.53	2	**39.98**	40.34	0.36	-44.80	1	4	N	87.82%
40		bottom	DET	20.22	4	**19.78**	20.03	0.25	-44.80	2	4	N	97.82%
41	Brunkovac_let2	bottom	GT	24.53	16	**10.31**	10.51	0.20	-50.90	-1	4	N	42.05%
42		bottom	GT	19.57	9	**19.11**	19.11	0.00	-57.00	0	4	N	97.67%
43		bottom	GT	29.48	7	**27.33**	27.61	0.28	-44.80	1	4	N	92.69%
44	Brunkovac_let1+2	bottom	DET	24.57	22	**9.61**	9.84	0.23	-52.92	-1	4	N	39.10%
45		bottom	DET	19.40	2	**17.89**	18.02	0.13	-59.00	0	4	N	92.22%
46		bottom	DET	22.95	4	**22.81**	22.82	0.01	-59.00	1	4	N	99.40%
47		bottom	DET	20.05	10	**19.64**	19.64	0.00	-57.00	2	4	N	97.92%
48		bottom	DET	42.88	2	**37.34**	37.68	0.34	-44.80	3	4	N	87.08%
49		bottom	DET	17.57	4	**17.14**	17.38	0.24	-44.80	4	4	N	97.54%
50	Brunkovac_let1+2	bottom	GT	22.92	26	**5.22**	5.48	0.26	-54.95	-1	4	N	22.78%
51		bottom	GT	22.89	5	**19.99**	20.15	0.16	-59.00	0	4	N	87.34%
52		bottom	GT	20.23	5	**19.96**	19.97	0.01	-59.00	1	4	N	98.66%
53		bottom	GT	21.73	9	**21.31**	21.31	0.00	-57.00	2	4	N	98.04%
54		bottom	GT	26.83	7	**24.68**	24.95	0.27	-44.80	3	4	N	92.00%

5.6.3. Flat meadow

The first experiment was conducted at Lonjsko polje (GPS: 45.480357, 16.701824). The terrain was a flat meadow with sparse trees, intersected by a stream and some puddles. Visibility was good, wind minimal (1 m/s). The person we were searching for wore contrasting clothes (dark) compared to the environment (light green). 4 straight flights were performed from different directions over the person (Fig. 15). The minimum distance of the drone from the person was 31 m, and the maximum was 90 m. The images were taken in a resolution of 5472×3648 px (aspect ratio 3:2).

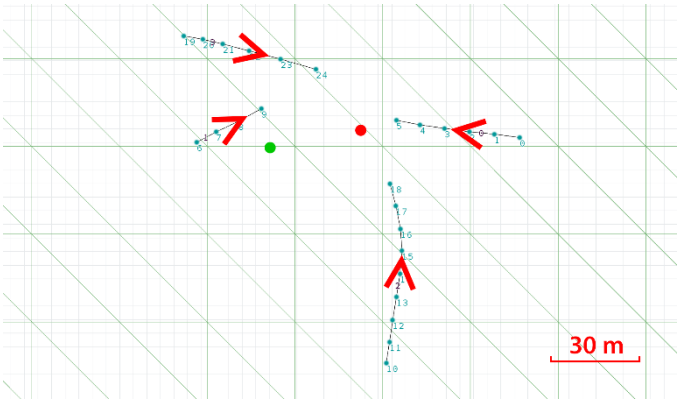


Figure 15: Top view of 4 straight flights from different directions (connected cyan dots) over the person we search for (red dot). The green dot indicates the drone's home point, and each rectangular pair of triangles represents a 30×30 m terrain patch (green mesh in the background).

The detector had a high performance (ROpti = 84.0%; Table 6, dataset #1). It was confused only by the reflections of the sky in the puddles (Fig. 16).

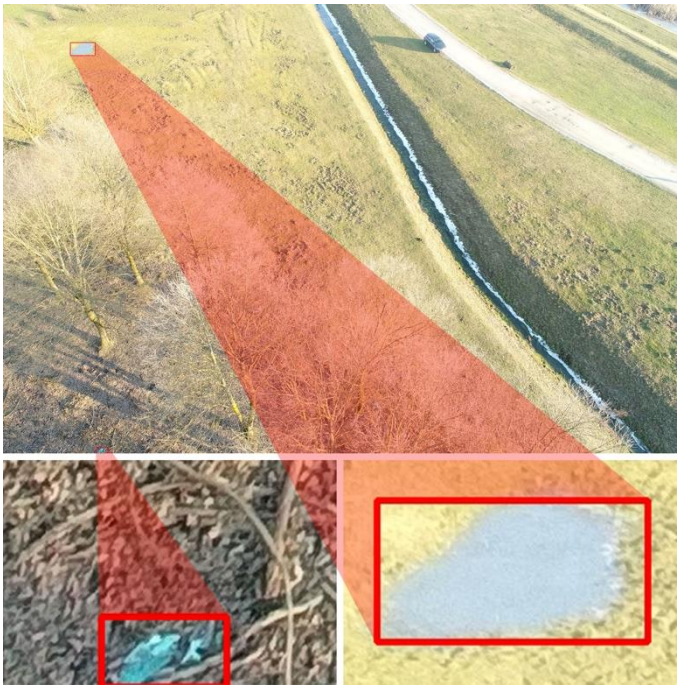


Figure 16: Top: example of false detections in puddles. Bottom: 400% zoomed detections.

Using the center of the bounding box as the detection coordinate, the Error_2D (distance from GT) of averaged detections (23) throughout the flight was 4.78 m (Table 9, result #1). Error_Difference was 0.00 m, in accordance with flat terrain. We consider location determination successful according to ErrDist criteria (8.67% < 10%).

Analysis of individual sequences shows that the results (Table 9, #2-3, Error_2D) were significantly better than the average of the entire flight (Table 9, #1). Among them is the best-achieved result (0.7 m) of all measurements. The results of the next two sequences (Table 9, #4-5) are almost twice as bad, which can be explained by the strong correlation (0.72) found between Error_2D and Gimbal_Pitch_Degree (camera's pitch). Fig. 17 shows the change in Error_2D by sequences. The first two sequences (0 and 1) were flights in opposite directions, using the camera's pitch -51.5°. The next two sequences (2 and 3) were also flights in opposite directions but using the camera's pitch -44.2° and -45.4°, respectively.

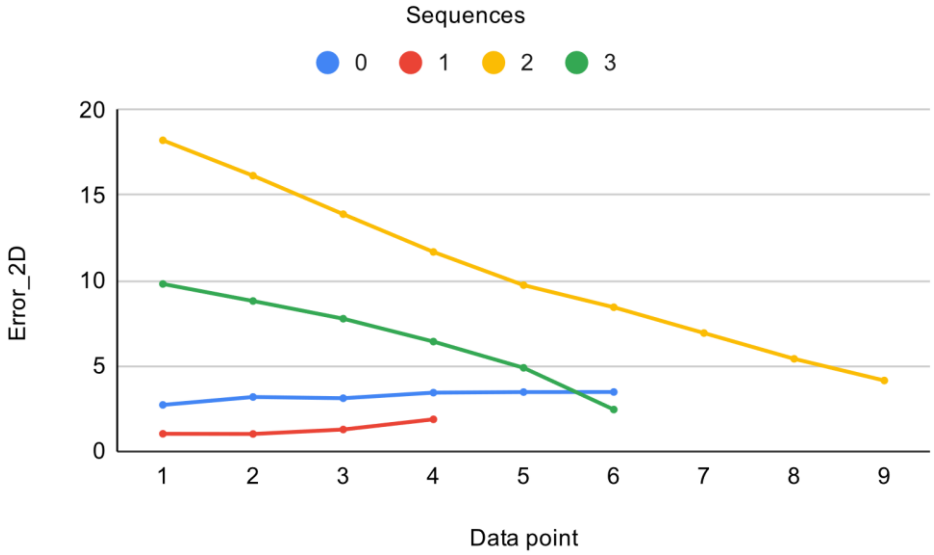


Figure 17: Error_2D results by flight sequences, using the middle-bottom of the bounding box (DET location) as the detection coordinate.

After noticing that the person's complete figure is visible in the images, we added to raycaster the ability to treat the middle of the bottom of the bounding box as the detection coordinate. In this way, repeated measurements (results #6-10) favored sequences recorded with a smaller pitch (in absolute value) and improved the average result of the entire flight. Error_2D fell from 4.78 m to 4.17 m, and ErrDist from 8.67% to 7.57%.

5.6.4. Slight slope

For the second experiment, a vineyard located on a slight slope was selected (GPS: 45.490690, 16.769242). Visibility was good, but the wind increased (3 m/s). The person we were searching for was again contrastingly dressed (dark) compared to the environment, which was primarily green. Two sets of flights were performed (Fig. 18). The first set (vinograd) contains only 1 straight flight over the terrain, with the camera focused exclusively on the area planted with vines. The second set (vinograd2) contains 5 flights, of which 4 were straight in 2 opposite directions and 1 orbital, with various rural elements (houses, auxiliary facilities, garden elements, vehicles) appearing in the images. While recording the second set, the resolution of the photos was reduced to 4864×3648 (changing the aspect

ratio to 4:3), and the raycaster was improved to support variable image resolutions.

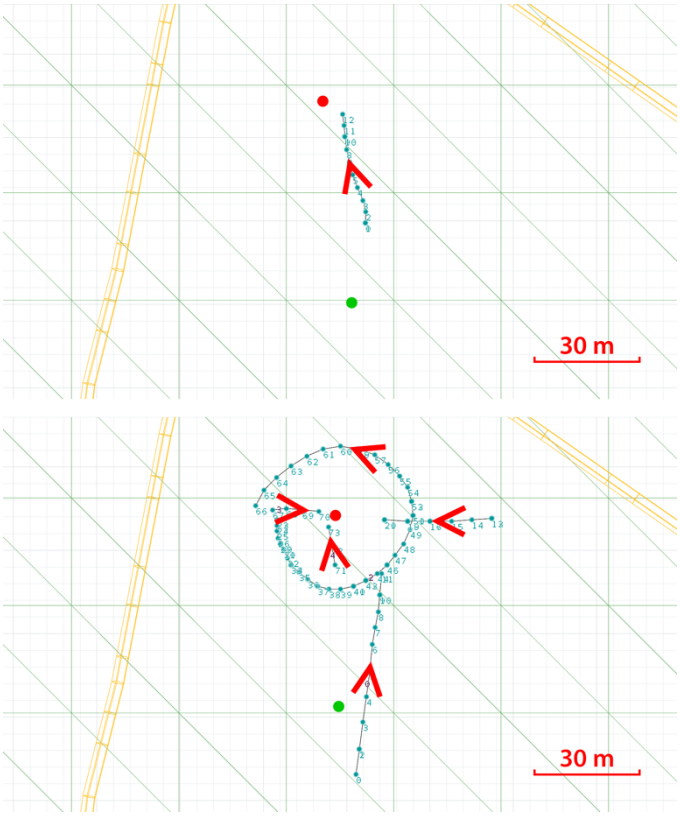


Figure 18: Top: top view of single straight flight (connected cyan dots) towards the person we search for (red dot). The green dot indicates the drone's home point, and each rectangular pair of triangles represents a 30×30 m terrain patch (green mesh in the background). Yellow geometry represents roads. Bottom: top view of 5 flights, of which 2 pairs in opposite directions and 1 orbiting.

The detector had a half success rate (ROpti fell to 50.0% and 55.2%, respectively; Table 6, #2-3), with many false detections, detecting various garden elements as persons (Fig. 19). This suggests that garden elements should be used as negative samples when training detectors for use in rural areas.

Figure 19: Top: example of false detections (garden elements). Bottom: 800% zoomed detections.

Despite many false detections (Table 6, #2-3) and multiple detections on individual images, their averaging yielded good results. Error_2D of the first set, using the center of the detection's bounding box, was 7.95 m (Table 9, result #11), and using the middle of the bottom of the detection's bounding box, 6.57 m (Table 9, result #12). A similar Error_2D was achieved for the entire flight in the second set (7.47 m and 6.56 m, respectively, results #13 and #19). Error_2D of individual sequences ranged between 4.77 m and 23.95 m. Error_Difference ranged between 0.01 m and 0.06 m indicating uneven terrain (corresponding to a slight slope).

The camera's pitch in individual sequences varied between -38.30° and -63.80° and confirmed a strong correlation (0.70) between the camera's pitch and Error_2D (Table 9, results #13-18 and #20-24), observed in the first experiment.

Although all the above results are satisfactory for practical application on a specific type of terrain, we do not consider any of these results successful according to the ErrDist criterion because they range from 18.68% to as much as 77.06%.

5.6.5. Mountain

The third experiment was performed on mountainous terrain (Moslavačka gora: 45.596889, 16.752002). Visibility was good, and the wind was the strongest (5 m/s) compared to other experiments. The person we were searching for was dressed in camouflage (white) in relation to his surroundings (sitting on white masonry ruins in the middle of a deciduous forest in winter). Two flights (Vis_let1 and Vis_let2) were performed in opposite directions, both below the take-off point (Vis peak), as a standard scenario for achieving the highest search coverage by overflights in mountainous conditions (Fig. 20).

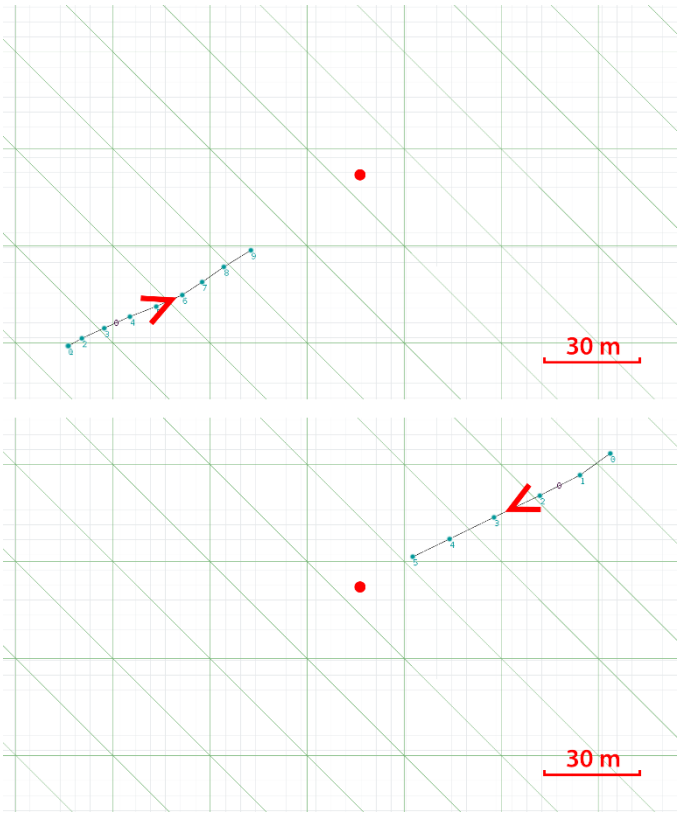


Figure 20: Top and bottom: top view of 2 straight flights from different directions (connected cyan dots) towards the person we search for (red dot). The home point (the green dot) is not visible because it was further away. Each rectangular pair of triangles represents a 30×30 m terrain patch (green mesh in the background).

As expected, due to camouflage conditions, ROpti was negative (-10.0 and -16.7) because no successful detection was achieved. To be able to evaluate the success of the method despite this result, raycaster was given the option to use GT instead of detections (Fig. 21).

Figure 21: Top: example of GT used as detection coordinates. Bottom: 1000% zoomed detection.

Using the middle of the bottom of the GT's bounding box, Error_2D of the first flight (Table 9, result #25) was 41.04 m, and the second (Table 9, result #27) 36.19 m. While these are arguably the worst Error_2D results of all flights, they are still usable in real conditions (especially due to the lack of leaves on the trees). The Error_Difference of individual flights increased (compared to 2nd experiment) to 0.28 m, indicating slightly higher terrain unevenness. However, values are still low, suggesting that all detections (GTs) are located at approximately the same isohypse.

ErrDist is significantly higher than the threshold (47.94% and 58.76%), but, given the large Error_2D, we can consider it moderate due to the large average distance of the drone from the person we search for (85.60 m and 61.59 m).

The camera's pitch varied between -33.71° and -48.90° and, as such, belongs to the zone in which, according to the previous two experiments, we expect worse results (higher Error_2D). This prompted us to refine the raycaster with the ability

to calibrate the drone camera orientation after the flight. Since we used GT, we were able to find the optimal calibration, which in this case was +15° for pitch and +3° for yaw. By measuring after calibration, we achieved Error_2D 3.47 m for the first flight (Table 9, result #26) and 5.97 m for the second flight (Table 9, result #28). This also resulted in satisfactory Err_Dist in both cases (4.06% and 9.69%).

Since such post-calibration is not possible in real conditions, we also tested the ensemble approach, averaging the results of both flights together. We achieved an excellent result: Error_2D dropped from 41.04 m and 36.19 m to only 4.04 m, with a satisfactory ErrDist of 4.72%. When we applied the previously determined calibration to this, the result deteriorated: Error_2D rose to 4.68 m and ErrDist to 5.47%. These are still good results but indicate that averaging the results of flights in opposite directions achieves better results even without calibration, which may be more useful for practical application.

Even more useful is the awareness of an almost perfect negative correlation (-0.996) between Error_2D and the Y coordinate of the detection center (in this case GT_CY), with an almost perfect correlation (0.993) between drone distance from GT and Error_2D. These correlations indicate greater reliability of detections that are positioned lower in the image, and which are therefore closer to the camera. In this case (Vis_let1), Error_2D drops from 49.61 m, when the distance from the drone is 113.92 m, and the Y coordinate detection is 1052nd px (of 3648 px), to 25.96 m, when the distance from the drone is 47.68 m, and the Y coordinate is 2766th px. This means that if we come across a sequence of linearly decreasing Error_2D results and linearly increasing Y detection coordinates (Table 10), we can reliably take the best Error_2D result (25.96 m) within the sequence, thus avoiding a worse result resulting from averaging the results of individual sequences (in this case 41.04 m, Table 9, result #26).

Table 10. Linearly decreasing drone to GT distance (Distance), linearly increasing Y coordinate of the center of the GT's bounding box, and linearly decreasing Error_2D for Vis1_let dataset, resulting in almost perfect correlations.

Distance	GT_CY	Error_2D
113.9183197	1052	49.61310959
113.7919846	1066	50.25457764

108.704155	1147	49.35968018
100.6928329	1277	47.24179459
91.62268829	1433	44.01839066
82.58696747	1618	40.83591843
73.509552	1819	37.62094116
65.97645569	2030	34.58089447
57.5095253	2324	30.9470005
47.68379211	2766	25.95678902

5.6.6. Narrow valley

The last experiment in the series was conducted in a narrow valley (GPS: 45.491297, 16.768484). Visibility was good, wind 4 m/s. 2 sets of flights were performed, with 2 flights in opposite directions in each (Fig. 22).

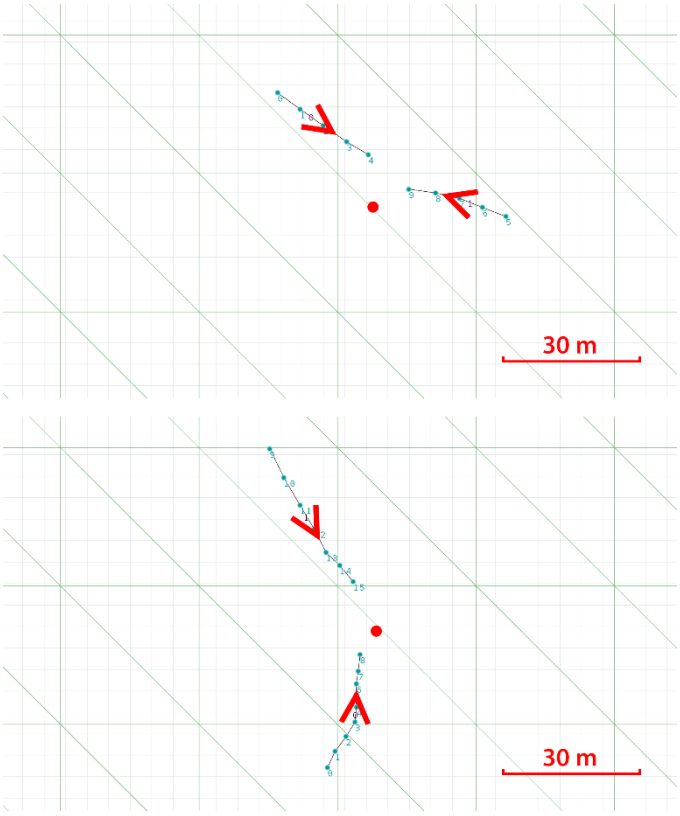


Figure 22: Top and bottom: top view of 2 pairs of straight flights in opposite directions (connected cyan dots) over the person we search for (red dot). The home point (the green dot) is not visible because it was further away. Each rectangular pair of triangles represents a 30×30 m terrain patch (green mesh in the background).

Unlike previous experiments in which the person we were searching for was an adult of average height and build, in this case, it was a 10-year-old child of smaller

build. The person is dressed in a combination of light and dark clothes and is located near a stream (Fig. 23).

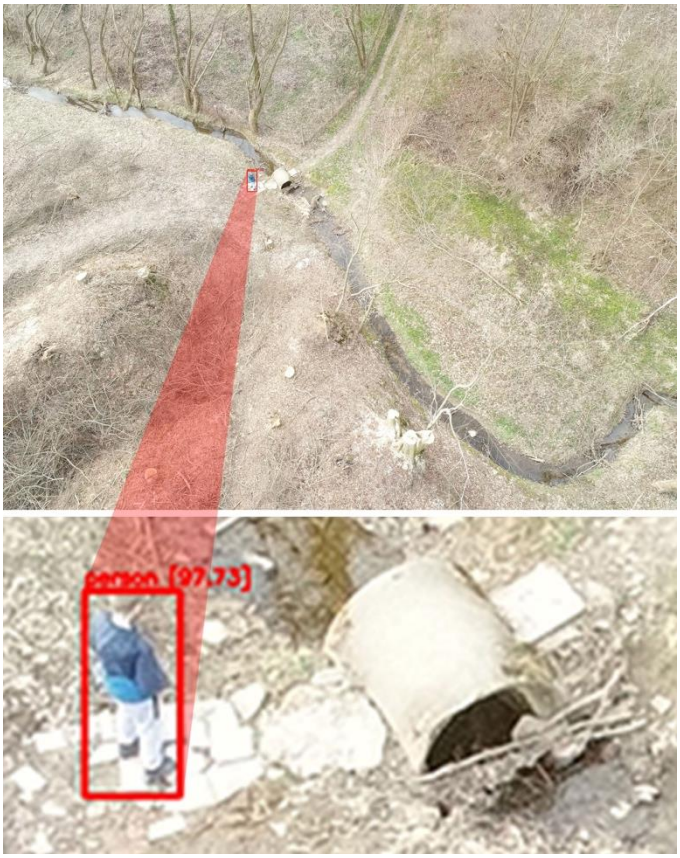


Figure 23: Top: example of successful detection of a 10-year-old child of smaller build. Bottom: 1000% zoomed detection.

Aware of the previous impact of drone telemetry error and the problem of 3D terrain resolution that should become apparent at this location (the valley is narrower than 1 px elevation data), we first tested the reliability of telemetry by repeatedly turning the drone on and off on the ground and recording single control image that contains the recorded GPS location of the drone. It was shown that the drone reports its position with a deflection of 3.6 to 6.4 m in relation to the actual position.

The correlation between wind speed and achieved Error_2D for all flights is moderate (0.56). However, due to the mentioned limitation (measuring wind speed only at the ground level) and the ability of the drone to cope with the wind up to 10 m/s, we did not take into account wind speed and direction when assessing the impact on the reliability of telemetry.

High ROpti (60.0 and 87.5) allowed us to use detection coordinates for raycasting, but due to testing the impact of telemetry error on Error_2D, we performed all measurements using GT as well. Error_2D of entire flights, using detection coordinates, was 11.18 m for the first and 16.73 m for the second flight (Table 9, results #31 and #37), and using GT, 10.68 m and 10.31, respectively (Table 9, results #34 and #41). In all 4 cases, the mentioned Error_2D is almost twice as good as the Error_2D of individual sequences, which indicates a positive effect of averaging sequences.

Insufficient 3D terrain resolution, combined with telemetry error, led to the positioning of the virtual drone (during raycast) below the 3D terrain (Fig. 24). Instead of producing poor results, this situation revealed the robustness of the system and another advantage of the proposed geolocation method. Trying to cast a ray that can not hit the ground (because the camera is below it) results in the starting coordinate (position of the drone/camera) being treated as a detection location. Although this is a faulty result, it guarantees that the drone's distance from the GT is the largest Error_2D we can get in this case. Since this is the best result we have had before raycasting, any successful detection paired with successful raycasting can only improve it, which happened during this experiment.

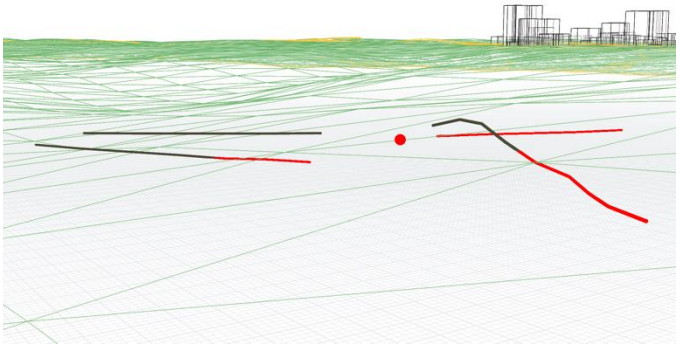


Figure 24: Perspective view of 4 flight paths towards the person we search for (red dot), simultaneously shown. The black part represents above the ground segment of the flight path. The red part represents below the ground segment of the flight path, indicating insufficient 3D terrain resolution, combined with telemetry error.

Finally, we tested the ensemble approach on this data (Brunkovac_let1+2). We averaged the results of both flights and achieved Error_2D for the entire flight of 9.61 m, using detections (Table 9, result #44), and 5.22 m, using GT (Table 9, result #50). Although the achieved ErrDist varies between 22.78% and 59.96% (and, as such, does not exceed the success threshold), Error_2D between 5.22 and 16.73 m represents excellent results for practical application, especially given the specificity of the terrain and pre-known problems of telemetry and 3D terrain resolution.

6. Recommendations for using proposed geolocation method in real-world scenarios

Following the conclusions of the experiments, we make recommendations that include the setup of a drone, person detector, 3D terrain generator, raycaster, and the applicability of the SAR-DAG system to real-world SAR missions.

6.1. Drone setup

After calibrating the drone and before take-off, it is recommended to measure the deviation range of the recorded position of the drone (at the home point) in the flight log in relation to its actual position. By our measurements, low-cost commercial drones without RTK module report this deviation within a 5 m radius from the actual location (average being 4.79 m, with 4.69 m median). Therefore we consider this error intrinsic to the system and, as seen from the results of our experiments, confirm that it does not negatively influence our proposed method. During the raycast, it is possible to correct the home point of the virtual drone with the obtained average deviation value. However, this is unnecessary for an average deviation of up to 5 m since good results could be achieved without correction.

Before take-off, it is mandatory to record the correct GPS coordinate of the home point because the absolute altitude at that point is used to calculate the absolute altitude of the drone during the flight, using the relative altitude recorded by the drone in its log for each flight point.

It is recommended to take low oblique photos, using a pitch between -50° and -60°, for which the proposed method gives good results. A recording frequency of at least 1 image every 2 seconds is recommended, at the maximum resolution of a camera that supports that frequency. This way, a sufficient number of images will be recorded during the flight, enabling a successful detection of the person using a well-trained detector.

The recommended flight altitude is 30 m, but in practice, it depends on the combination of the camera resolution, the detector model, and the dataset on which it was trained. Since several drones are often in the field during SAR missions, vertically spaced from each other by 20 m, it is recommended to train detectors with augmented data corresponding to different flight altitudes.

If possible, it is recommended to perform pairs of flights in opposite directions (for the ensemble approach when using the proposed geolocation method) as they can increase the accuracy of geolocalization 2-10 times.

6.2. Detector setup

For a more reliable operation of the person detector, it is recommended to additionally train the model with negative examples of water bodies (with reflections in them) and specific pre-known objects located in search areas, which could be otherwise falsely detected as persons.

For particularly challenging detection cases (e.g., a person dressed in white, walking in snow), it is recommended to train the detector model with examples of specific equipment that the person may carry (e.g., backpack, walking sticks) according to information available to rescuers.

6.3. 3D terrain generator setup

When generating 3D terrain, we recommend using elevation data with at least 30 m/px resolution to minimize raycasting error. If we prepare the system for a search in the area of more complex relief, we recommend purchasing or creating a DEM in a much higher resolution (up to 1 m/px). The 3D terrain model created from this data can be adaptively resampled to minimize the number of points and

polygons while keeping essential features of the relief. Minimizing the number of points and polygons improves raycasting speed.

In addition to elevation data, we recommend using OpenStreetMap metadata to add reference 3D objects (buildings and traffic infrastructure) that are not used for raycasting but can help the raycasting operator to navigate otherwise monotonous 3D terrain.

If the proposed geolocation method is used as part of a system that, in addition to the GPS coordinates of the located person, returns a suggested route to guide the rescuers from the nearest road, it is recommended to incorporate additional attributes such as soil type information (in the context of passability) and potential hazards (e.g., minefield) in the 3D terrain model.

6.4. Raycaster setup

The established correlations indicate higher reliability of detections that are positioned lower in the image, and which are therefore closer to the camera. Thus, in the linear detection series, it is recommended to use the raycast location of the lowest coordinate instead of the averaged coordinates.

To match the camera's optics with the 3D camera, it is necessary, in addition to focal length and image dimensions (image width and image height), to know the exact horizontal FOV (HFOV) in degrees. Drone user manuals list the camera's FOV but often do not explicitly state what type of FOV it is. In such cases, it is usually a diagonal FOV from which it is necessary to calculate the horizontal FOV. The conversion can be done using Eq. 5 and 6:

$$h = \frac{\sqrt{{image_w}^2 + {image_h}^2}}{\tan\left(\frac{FOV \times \pi}{360}\right)} \tag{5}$$

$$HFOV = \frac{360}{\pi} \times \tan^{-1}\left(\frac{image_w}{h}\right) \tag{6}$$

6.5. Applicability of the SAR-DAG system to real-world SAR missions

For this research, the data acquisition phase (Section 3.1.) used a low-cost commercial DJI Phantom 4 Advanced drone whose battery allows up to 30 minutes of flight time. In practice, for safety reasons (considering the condition of the batteries and the strength of the wind), we set the alarm to sound at 30% battery capacity, and at 20% capacity, the drone lands. Therefore, the average effective flight time is 21 minutes.

During 21 minutes of flight (Fig. 25), the drone can cover ~100,000 m2 (10 ha) area, recording 2 images per second for a total of 630 images (~5 GB of data). When the drone returns to the base, this data is copied to the computer for offline processing. The time required to copy 5 GB of data using USB 3.0 SuperSpeed transfer type is 77 seconds.

The detection module (Section 3.2.) processes images at a speed of 268.1 ms/image using a Dell G3 laptop (2.6 GHz i7-9750H CPU, 16 GB RAM, and GeForce GTX 1660 Ti 6 GB). It takes 169 s to process the entire set (630 images).

The geolocation module downloads the necessary DEM and OSM data within 1 minute (depending on the speed of the Internet connection). Generating 2M polygons for 3D terrain (30×30 km2 area) using a MacBook Pro laptop (2.7 GHz i7 CPU, 16 GB RAM, and GeForce GT 650M 1 GB) takes ~4 minutes. Both tasks (downloading data and generating 3D terrain), considering that they take shorter than the drone flight (5 min < 21 min), can be done before or during the drone flight.

The raycast speed depends on the number of polygons (Paulin et al., 2021), and in our case (2M polygons), it takes 1.15 ms per single detection. The determined median number of detections during a single flight in our experiments is 21 (Section 5.1.), meaning that resolving geolocation using the proposed geolocation method (Section 4.) takes only ~24 ms.

Table 11. Duration of individual tasks within the SAR-DAG system prototype.

Task	Duration
Drone flight	21 min
Downloading DEM and OSM data (during flight)	1 min
Generating 3D terrain (during flight)	4 min
Total data acquisition time	**21 min**
Data copying	77 s
Person detecting	169 s
Person geolocating	1 s
Total data processing time	**247 s**
Minimum time required to complete 1 flight of maximum duration and offline data processing (by determining geolocation)	**~25 min**

The minimum time required to complete 1 flight of maximum duration (21 minutes) and offline data processing (by determining geolocation) is 25 minutes and 7 seconds (Table 11). Using a single computer for offline processing and 4 drones simultaneously during an actual SAR mission, it is possible to achieve optimal utilization of the SAR-DAG system while covering a 40 ha search area within the same time (~25 minutes).

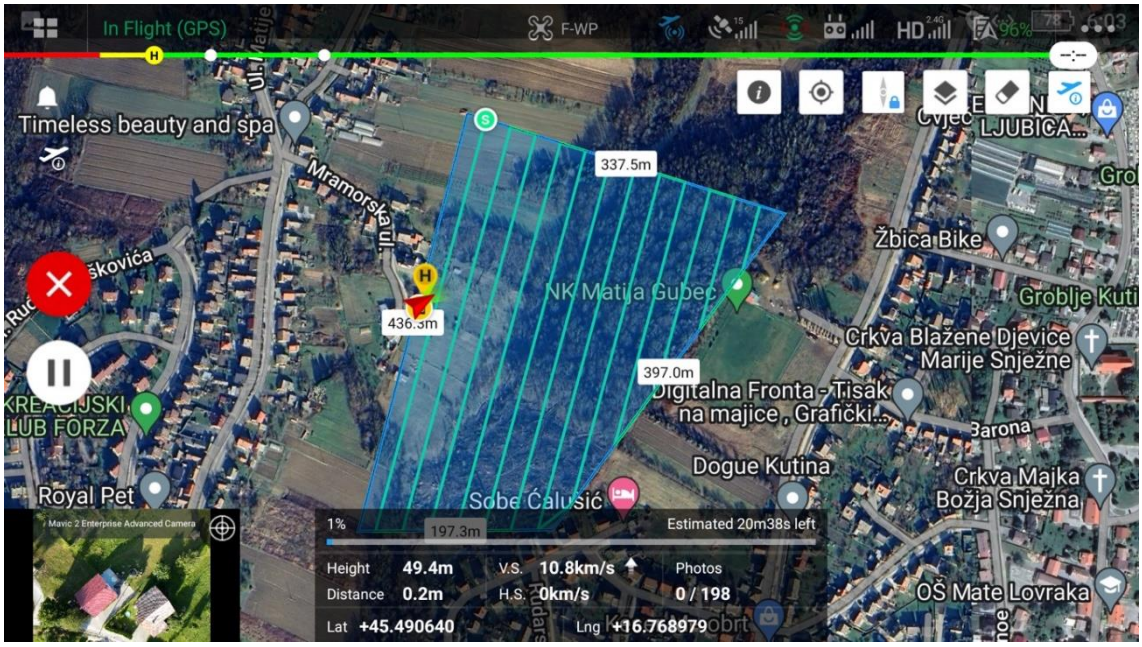


Figure 25: The flight path of the drone (green lines) and the 10 ha covered area (blue) during the 21-minute flight, shown in the DJI Pilot software.

7. Conclusion

In this paper, we explored the possibility of applying the geolocation method based on the raycast in different land search and rescue real-world scenarios, using low-cost commercial drones with a monocular camera and no RTK module, enabling laser rangefinder emulation during offline image analysis.

To test the usability and efficiency of the proposed method in actual SAR missions, we built a Search and Rescue - Detection and Geolocation (SAR-DAG) system, with a goal to precisely geolocate persons automatically detected in offline processed images recorded during the SAR mission.

For the experiments, we selected 4 locations of different configurations and terrain requirements ranging from flat terrain, over terrain with a slight slope, to mountainous terrain and narrow valley. We performed drone flights at 30 m altitude and recorded sequences of monocular oblique aerial photographs at these locations.

Using drone and camera telemetry data from flight logs along with person detection results, we prepared multiple datasets in a proposed Input CSV format that combines them. We used YOLOv4 trained on our SARD dataset for person detection. The achieved AP of the model was 61.3%, and the corresponding ROpti was 92.8%.

Since the application of the proposed geolocation method, besides a sequence of aerial images and person detections, requires 3D terrain, we built a procedural 3D terrain generator that allowed us, using NASA's terrain elevation data (at 30 px/m resolution) and OpenStreetMaps metadata, to obtain, respectively, appropriate polygonal mesh for raycasting and better orientation of human operator evaluating the raycasting results. We applied the proposed method to 9 datasets using a custom-made raycaster that allowed us to monitor each scenario performance in 3D, including from the perspective of a virtualized drone camera. This approach allowed us to notice problems as they occurred and solve them by continuously adjusting and improving the raycaster.

For geolocation evaluation metrics, we chose the distance of the location determined by the proposed method from GT (Error_2D), which, based on previous SAR experience, assesses the practical usability of the results on certain types of terrain. To objectify the assessment of the achieved result, we introduced the ErrDist measure as a percentage of the error of the determined location in relation to the actual distance of the drone from the person we are searching for.

By adjusting the raycaster, we achieved good results on all types of terrain, with Error_2D ranging from 3.47 m to 16.73 m and ErrDist ranging from 4.06% to 59.96%, using all sequences of each flight. The lowest Error_2D (0.7 m) and ErrDist (1.23%) were achieved using a single flight sequence with only 4 consecutive detections, outperforming the previously best result (2.3 m) by 1.6 m (69.57% decrease) and proving that the proposed geolocation method can be adapted for in vivo use. Also, by being able to process offline data acquired during each 21-minute flight over a 10 ha area in 247 seconds, we proved that the proposed method can be efficiently used in actual SAR missions. Raycaster proved robust even in the combined unfavorable conditions of drone telemetry error and insufficient 3D terrain resolution. Following the experiment analysis, we made recommendations related to the setup of a drone, person detector, 3D terrain generator, raycaster, and the applicability of the SAR-DAG system to real-world SAR missions.

In future work, we plan to replicate this series of experiments using more advanced drones with better telemetry, generate synthetic RGB and thermal datasets for model training to improve detection performance, and improve handling multiple detections by identifying (and subsequently reidentifying) each person and averaging raycast results per unique person per flight sequence.

Table 12. Abbreviations.

Abbreviation	Description
AP	Average Precision
API	Application Programming Interface
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CNN	Convolutional Neural Network
COCO	Common Objects in Context

CSV	Comma-Separated Values
DEM	Digital Elevation Model
DET	Detection
ENU	East-North-Up
EPSG	European Petroleum Survey Group
FHD	Full High Definition
FN	False Negative
FOV	Field of View
FP	False Positive
GCP	Ground Control Points
GPS	Global Positioning System
GT	Ground Truth
HFOV	Horizontal Field of View
IMU	Inertial Measurement Unit
IMP	Improvement
IOU	Intersection over Union
IPG	Iterative Photogrammetry
IRT	Iterative Ray-Tracing
LiDAR	Light Detection and Ranging
MAV	Micro Air Vehicle
NDC	Normalized Device Coordinates
NED	North-East-Down
RGB	Red Green Blue
ROpti	Recall Optimal
RT	Ray-Tracing
RTK	Real-Time Kinematic
SAR	Search and Rescue
SAR-DAG	Search and Rescue - Detection and Geolocation
SD	Secure Digital
SfM	Structure from Motion
SLAM	Simultaneous Localization and Mapping
SRTM	Shuttle Radar Topography Mission
TP	True Positive
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
YOLO	You Only Look Once

Table 13. YOLOv4 training parameters.

Parameter	**Value**
Batch size	64
Subdivision	32
Iterations	6000
Learning rate	0.001
Momentum	0.949
Decay	0.0005

Network resolution	512×512

Table 14. Proposed geolocation method variables.

Variable	Description
f	normalized device coordinates
t	3D coordinates
c	camera's position
o	camera's orientation
r	raycast direction vector
p	point of intersection
d	distance

References

30-Meter SRTM Tile Downloader. (n.d.). Retrieved February 9, 2022, from https://dwtkns.com/srtm30m/

Aki, M., Rojanaarpa, T., Nakano, K., Suda, Y., Takasuka, N., Isogai, T., & Kawai, T. (2016). Road Surface Recognition Using Laser Radar for Automatic Platooning. *IEEE Transactions on Intelligent Transportation Systems*, *17*(10), 2800–2810. https://doi.org/10.1109/TITS.2016.2528892

Albanese, A., Sciancalepore, V., & Costa-Perez, X. (2022). SARDO: An Automated Search-and-Rescue Drone-Based Solution for Victims Localization. *IEEE Transactions on Mobile Computing*, *21*(9), 3312–3325. https://doi.org/10.1109/TMC.2021.3051273

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*.

Bradshaw, R. C., Schmidt, D. P., Rogers, J. R., Kelton, K. F., & Hyers, R. W. (2005). Machine vision for high-precision volume measurement applied to levitated containerless material processing. *Review of Scientific Instruments*, *76*(12), 125108. https://doi.org/10.1063/1.2140490

Cai, Y., Zhou, Y., Zhang, H., Xia, Y., Qiao, P., & Zhao, J. (2022). Review of Target Geo-Location Algorithms for Aerial Remote Sensing Cameras without Control Points. *Applied Sciences*, *12*(24). https://doi.org/10.3390/app122412689

COCO - Detection Evaluation. (n.d.). Retrieved July 9, 2023, from https://cocodataset.org/#detection-eval

COCO - Detection Leaderboard. (n.d.). Retrieved August 11, 2023, from https://cocodataset.org/#detection-leaderboard

Conte, G., Hempel, M., Rudol, P., Lundstrom, D., Duranti, S., Wzorek, M., & Doherty, P. (2008). High Accuracy Ground Target Geo-location Using Autonomous Micro Aerial Vehicle Platforms. *AIAA Guidance, Navigation and Control Conference and Exhibit*. https://doi.org/10.2514/6.2008-6668

Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object Detection via Region-Based Fully Convolutional Networks. *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 379–387.

DJI Matrice 30. (n.d.). Retrieved April 9, 2022, from https://www.dji.com/hr/matrice-30/specs/

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.* https://doi.org/https://doi.org/10.48550/arXiv.2010.11929

EPSG:4326. (n.d.). Retrieved February 9, 2022, from https://epsg.io/4326

Forlani, G., Diotri, F., Cella, U. M. di, & Roncella, R. (2019). Indirect UAV Strip Georeferencing by On-Board GNSS Data under Poor Satellite Coverage. *Remote Sensing*, *11*(15). https://doi.org/10.3390/rs11151765

Haseeb, M. A., Guan, J., Ristić-Durrant, D., & Gräser, A. (2018). DisNet : A novel method for distance estimation from monocular camera. *10th Planning, Perception and Navigation for Intelligent Vehicles*, 139–144.

He, F., Zhou, T., Xiong, W., Hasheminnasab, S. M., & Habib, A. (2018). Automated Aerial Triangulation for UAV-Based Mapping. *Remote Sensing*, *10*(12). https://doi.org/10.3390/rs10121952

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2980–2988. https://doi.org/10.1109/ICCV.2017.322

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(9), 1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824

Hosseinpoor, H. R., Samadzadegan, F., & Dadrasjavan, F. (2016). Pricise Target Geolocation and Tracking Based on Uav Video Imagery. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLI-B6*, 243–249. https://doi.org/10.5194/isprs-archives-xli-b6-243-2016

Houdini Help. (n.d.). Retrieved February 9, 2022, from https://www.sidefx.com/docs/houdini/

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR*, *abs/1704.0*.

Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L.-C., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Adam, H., & Le, Q. (2019). Searching for MobileNetV3. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 1314–1324. https://doi.org/10.1109/ICCV.2019.00140

Jocher, G., Chaurasia, A., & Qiu, J. (2023). *Ultralytics YOLOv8*. https://github.com/ultralytics/ultralytics

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T., & Ferrari, V. (2020). The

Open Images Dataset V4. *International Journal of Computer Vision*, *128*(7), 1956–1981. https://doi.org/10.1007/s11263-020-01316-z

Leira, F. S., Trnka, K., Fossen, T. I., & Johansen, T. A. (2015). A ligth-weight thermal camera payload with georeferencing capabilities for small fixed-wing UAVs. *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 485–494. https://doi.org/10.1109/ICUAS.2015.7152327

Leu, A., Aiteanu, D., & Gräser, A. (2012). High Speed Stereo Vision Based Automotive Collision Warning System. In R.-E. Precup, S. Kovács, S. Preitl, & E. M. Petriu (Eds.), *Applied Computational Intelligence in Engineering and Information Technology*. Springer Berlin Heidelberg.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Wei, X., & Wei, X. (2022). *YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer Vision -- ECCV 2014* (pp. 740–755). Springer International Publishing.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision -- ECCV 2016* (pp. 21–37). Springer International Publishing.

Liu, Y., Zhang, Y., Wang, Y., Hou, F., Yuan, J., Tian, J., Zhang, Y., Shi, Z., Fan, J., & He, Z. (2023). A Survey of Visual Transformers. *IEEE Transactions on Neural Networks and Learning Systems*, 1–21. https://doi.org/10.1109/TNNLS.2022.3227717

MATRICE 300 RTK User Manual. (n.d.). Retrieved March 17, 2022, from https://dl.djicdn.com/downloads/matrice-300/20211125UM/M300_RTK_User_Manual_EN_v3.0.pdf

OpenStreetMap > Export. (n.d.). Retrieved February 9, 2022, from https://www.openstreetmap.org/export#map=11/45.5502/16.8000

Overpass API. (n.d.). Retrieved February 9, 2022, from https://wiki.openstreetmap.org/wiki/Overpass_API

Pan, T., Deng, B., Dong, H., Gui, J., & Zhao, B. (2023). Monocular-Vision-Based Moving Target Geolocation Using Unmanned Aerial Vehicle. *Drones*, *7*(2). https://doi.org/10.3390/drones7020087

Paulin, G., Sambolek, S., & Ivasic-Kos, M. (2021). Person localization and distance determination using the raycast method. *2021 6th International Conference on Smart and Sustainable Technologies (SpliTech)*, 1–5. https://doi.org/10.23919/SpliTech52315.2021.9566329

Pietroszek, K. (2018). Raycasting in Virtual Reality. In N. Lee (Ed.), *Encyclopedia of Computer Graphics and Games* (pp. 1–3). Springer International Publishing. https://doi.org/10.1007/978-3-319-08234-9_180-1

Real-Time Object Detection on COCO. (n.d.). Retrieved August 11, 2023, from https://paperswithcode.com/sota/object-detection-on-coco?tag_filter=3%2C15%2C17%2C4

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. https://doi.org/10.1109/CVPR.2016.91

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6517–6525. https://doi.org/10.1109/CVPR.2017.690

Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*.

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), *IEEE Transactions on Pattern Analysis*

and Machine Intelligence (Vol. 39, Issue 6, pp. 1137–1149). Curran Associates, Inc. https://doi.org/10.1109/TPAMI.2016.2577031

Roth, S. D. (1982). Ray casting for modeling solids. *Computer Graphics and Image Processing*, *18*(2), 109–144. https://doi.org/10.1016/0146-664X(82)90169-1

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

Sambolek, S., & Ivasic-Kos, M. (2020). Person Detection in Drone Imagery. *2020 5th International Conference on Smart and Sustainable Technologies (SpliTech)*, 1–6. https://doi.org/10.23919/SpliTech49282.2020.9243737

Sambolek, S., & Ivasic-Kos, M. (2021). Automatic Person Detection in Search and Rescue Operations Using Deep CNN Detectors. *IEEE Access*, *9*, 37905–37922. https://doi.org/10.1109/ACCESS.2021.3063681

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

Sheng, Y. (2004). Comparative evaluation of iterative and non-iterative methods to ground coordinate determination from single aerial images. *Computers & Geosciences*, *30*(3), 267–279. https://doi.org/https://doi.org/10.1016/j.cageo.2003.11.003

Sheng, Y. (2005). Theoretical Analysis of the Iterative Photogrammetric Method to Determining Ground Coordinates from Photo Coordinates and a DEM. *Photogrammetric Engineering & Remote Sensing*, *71*(7), 863–871. https://doi.org/10.14358/PERS.71.7.863

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In Y. Bengio & Y. LeCun (Eds.), *3rd International*

Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings.

Sun, J., Li, B., Jiang, Y., & Wen, C. (2016). A Camera-Based Target Detection and Positioning UAV System for Search and Rescue (SAR) Purposes. *Sensors*, *16*(11). https://doi.org/10.3390/s16111778

Suziedelyte Visockiene, J., Puziene, R., Stanionis, A., & Tumeliene, E. (2016). Unmanned Aerial Vehicles for Photogrammetry: Analysis of Orthophoto Images over the Territory of Lithuania. *International Journal of Aerospace Engineering*, *2016*, 4141037. https://doi.org/10.1155/2016/4141037

Types of Aerial Photograph. (n.d.). Retrieved April 9, 2022, from https://www.ccsuniversity.ac.in/ccsu/Departmentnews/2020-09-15_200.pdf

Verykokou, S., & Ioannidis, C. (2015). *Metric Exploitation of a Single Low Oblique Aerial Image*.

Verykokou, S., & Ioannidis, C. (2018). Oblique aerial images: a review focusing on georeferencing procedures. *International Journal of Remote Sensing*, *39*(11), 3452–3496. https://doi.org/10.1080/01431161.2018.1444294

Vidal, A. R., Rebecq, H., Horstschaefer, T., & Scaramuzza, D. (2018). Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High-Speed Scenarios. *IEEE Robotics and Automation Letters*, *3*(2), 994–1001. https://doi.org/10.1109/LRA.2018.2793357

von Stumberg, L., Usenko, V., Engel, J., Stückler, J., & Cremers, D. (2017). From monocular SLAM to autonomous drone exploration. *2017 European Conference on Mobile Robots (ECMR)*, 1–8. https://doi.org/10.1109/ECMR.2017.8098709

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2021). Scaled-YOLOv4: Scaling Cross Stage Partial Network. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13024–13033. https://doi.org/10.1109/CVPR46437.2021.01283

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2023). YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors.

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 7464–7475.

Wang, C.-Y., Liao, H., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., & Yeh, I.-H. (2020). *CSPNet: A New Backbone that can Enhance Learning Capability of CNN*. 1571–1580. https://doi.org/10.1109/CVPRW50498.2020.00203

Zhang, L., Deng, F., Chen, J., Bi, Y., Phang, S. K., Chen, X., & Chen, B. M. (2018). Vision-Based Target Three-Dimensional Geolocation Using Unmanned Aerial Vehicles. *IEEE Transactions on Industrial Electronics*, *65*(10), 8052–8061. https://doi.org/10.1109/TIE.2018.2807401

Zhao, X., Pu, F., Wang, Z., Chen, H., & Xu, Z. (2019). Detection, Tracking, and Geolocation of Moving Vehicle From UAV Using Monocular Camera. *IEEE Access*, *7*, 101160–101170. https://doi.org/10.1109/ACCESS.2019.2929760

Zhu, J., & Fang, Y. (2019). Learning Object-Specific Distance From a Monocular Image. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 3838–3847. https://doi.org/10.1109/ICCV.2019.00394