

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6870

**Prepoznavanje i klasifikacija
motornih vozila na snimkama iz
zraka**

Dominik Vrbanić

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ODBOR ZA ZAVRŠNI RAD MODULA

Zagreb, 13. ožujka 2020.

Grana: 2.09.04 umjetna inteligencija

ZAVRŠNI ZADATAK br. 6870

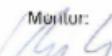
Pristupnik: Dominik Urbanić (0036507932)
Studij: Računarstvo
Modul: Računarska znanost

Zadatak: Prepoznavanje i klasifikacija motornih vozila na snimkama iz zraka

Opis zadatka:

Radom na zadatku potrebno je istražiti postojeće metode za prepoznavanje i klasifikaciju motornih vozila na snimkama iz zraka zasnovane na strojnom učenju. Potrebno je odabratи metodu pogodnu za primjenu u realnom vremenu za analizu fotografija i video zapisa vozila dobivenih snimanjem iz zraka pomoću bespilotne letjelice. Osim prepoznavanja i klasifikacije vozila, analizom je potrebno odrediti gustoću prometa te brzinu gibanja vozila.

Zadatak uručen pristupniku: 13. ožujka 2020.
Rok za predaju rada: 12. lipnja 2020.

Mentor:

Prof. dr. sc. Stjepan Bogdan

Dječavoda:

Izv. prof. dr. sc. Tomislav Hrkač

Predsjednik odbora za
završni rad modula:


Doc. dr. sc. Marko Čupić

Zahvaljujem mentoru prof. dr. sc. Stjepanu Bogdanu i mag. ing. Antonelli Barišić na pomoći pri izradi ovog završnog rada.

SADRŽAJ

1. Uvod	1
2. Strojno učenje	2
2.1. Vrste	3
2.2. Modeli	4
2.3. Pristupi rješavanju problema	5
3. Postojeće metode	7
3.1. Metode temeljene na ručnom stvaranju klasifikacijskog sustava	7
3.2. Metode temeljene na dubokom učenju	8
4. Odabrani algoritam	12
4.1. Implementacija modela	13
4.2. Rezultati	14
4.3. Gustoća prometa	21
5. Zaključak	22
Literatura	23

1. Uvod

Sustav za praćenje prometa sastavni je dio Inteligentnih transportnih sustava (ITS). Zadnjih godina u njega se sve više ulaže. Razlog tome je rast broja ljudi na zemlji pa tako i rast broja motornih vozila na cestama.

Prepoznavanje i klasifikacija vozila na snimkama iz zraka važno je za razne primjene. Primjeri koji se koriste u stvarnom svijetu su upravljanje prometom, korištenje parkirališta, planiranje gradova itd. Prikupljanje podataka o prometu korištenjem bespilotnih letjelica znatno je efikasnije od bilo kojeg drugog načina. Prednosti bespilotnih letjelica su niska cijena, mala veličina, sigurnost, ekološka povoljnost, brzina, ali i kvalitetne snimke. Bolje snimke dobivale bi se korištenjem satelita, međutim to bi zahtjevalo znatno više resursa.

Otkrivanje vozila može predstavljati izazovan problem zbog male veličine vozila na slici (automobil može biti samo 30x12 piksela). Još jedan problem predstavlja složena pozadina koju mogu sačinjavati objekti slični vozilima. Promjenjivi vremenski uvjeti (npr. oblaci, kiša, snijeg itd.) i osvjetljenje također predstavljaju velik problem pri detekciji vozila. U sustavu u stvarnom vremenu, vrijeme obrade (i računalne snage) je ograničeno, stoga metoda obrade treba biti što brža. Za rješavanje ovih izazova primjenjuje se računalni vid i strojno učenje.

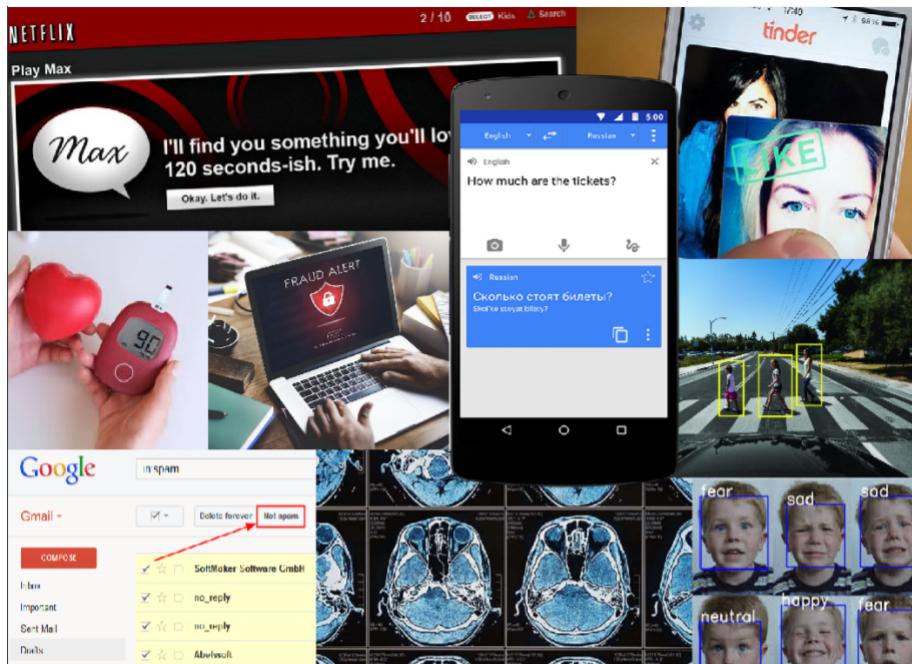
U drugom je poglavlju detaljnije objašnjeno strojno učenje te algoritmi i modeli koji se koriste u praksi. U trećem poglavlju nabrojane su i ukratko objašnjene postojeće metode za prepoznavanje i klasifikaciju vozila na snimkama iz zraka. Zatim je odabran optimalan algoritam za realizaciju zadalog problema te su u četvrtom poglavljju detaljno opisani postupak i rezultati dobiveni odabranom implementacijom.

2. Strojno učenje

Strojno učenje je grana umjetne inteligencije koja se bavi učenjem i samostalnim poboljšavanjem računalnih programa. Posebnim algoritmima grade se matematički modeli na temelju podataka iz skupa za učenje (engl. *training data*). Na tim se podacima model uči kako bi bolje mogao predvidjeti svojstva novih, još neviđenih podataka.

Primjeri strojnog učenja:

- Google koristi strojno učenje za poboljšanje rezultata traženja
- Netflix daje preporuke na temelju dotadašnjih odabira
- Facebook koristi strojno učenje za prikaz objava koje bi se korisniku trebale svidjeti na temelju njegovog dotadašnjeg ponašanja
- Digitalni pomoćnici koriste strojno učenje za unaprjeđenje tehnologije raspoznavanja govora
- Samovozeći automobili prate objekte oko sebe kako bi poboljšali svoje vozačke sposobnosti
- Raspoznavanje rukom pisanih znamenki
- Predviđanje ishoda izbora
- Zdravstvene ustanove koriste strojno učenje za dijagnosticiranje bolesti
- E-mail filtriranje dolazne pošte



Slika 2.1: Primjeri strojnog učenja [2]

2.1. Vrste

Postoje tri vrste strojnog učenja ovisno o ulaznim podacima:

- Nadzirano učenje (engl. *supervised learning*) :**

Računalu se daju primjeri ulaza i očekivanih izlaza. Na temelju zadanih podataka uči ga se pravilima kojima će ono samo ulazima pridruživati izlaze pomoću izgrađenog modela.

- Nenadzirano učenje (engl. *unsupervised learning*) :**

Računalu se daju podaci bez ciljne vrijednosti. Ono samo treba pronaći pravilnosti među podacima. Nenadzirano učenje ima veći opseg problema koje može riješiti od nadziranog učenja jer mu je za rad potrebno manje podataka.

- Podržano/ojačano učenje (engl. *reinforcement learning*) :**

Učenje optimalne strategije na temelju pokušaja s odgođenom nagradom, odnosno ovdje algoritam mora učiti iz svog iskustva.

Postoje dvije vrste strojnog učenja ovisno o izlaznim podacima:

- Regresija** : izlaz prima kontinuirane vrijednosti

- Klasifikacija** : izlaz može poprimiti samo određene diskretne vrijednosti

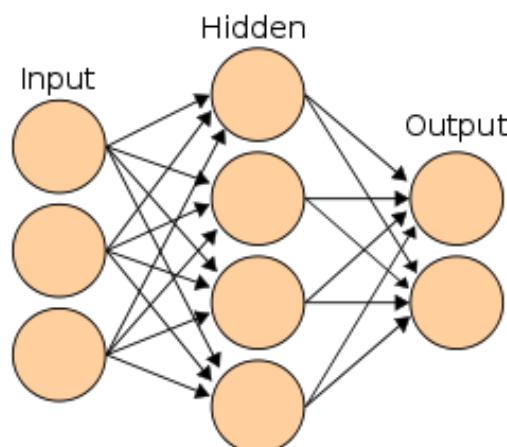
2.2. Modeli

Korištenje strojnog učenja uključuje stvaranje modela. Model se trenira na temelju sakupljenih podataka (zadanih parova ulaza i izlaza), koji mu pomažu da kasnije napravi prepostavke. Postoje mnogi tipovi modela, a za prepoznavanje i klasifikaciju motornih vozila na snimkama iz zraka uglavnom se koriste:

- **Umjetne neuronske mreže (engl. *Artificial neural networks*) :**

Inspiriran je neuronima u ljudskom mozgu. To je skup međusobno povezanih jednostavnih procesnih elemenata (neurona) čija se funkcionalnost temelji na biološkom neuronu i koji služe distribuiranoj paralelnoj obradi podataka. Mreža se sastoji od jednog ulaznog, jednog izlaznog te jednog ili više skrivenih slojeva među njima. Svaki sloj čine neuroni. Neuroni između slojeva međusobno su spojeni vezama kroz koje idu signali tako da se izlaz prethodnog sloja dovodi na ulaz sljedećeg sloja. Te se veze aktiviraju ako je zadovoljen uvjet postavljen aktivacijskom funkcijom.

Postoje unaprijedne neuronske mreže (neuron može biti povezan s više veza u sljedećem sloju i ne postoje ciklusi) i povratne neuronske mreže (veze su zatvorene, tj. postoji barem jedan ciklus). Posebna vrsta specijalizirana za preprocesiranje nestrukturiranih podataka (vezanih uz vizualne i audio signale) zove se konvolucijske neuronske mreže (engl. *CNN*) koje se često koriste pri obradi slikovnih podataka.



Slika 2.2: Neuronska mreža [19]

- **Stabla odluke (engl. *Decision trees*) :**
Stabla odluke su prediktivni modeli koji na temelju podataka izvode njihove veze s ciljem dobivanja izlaznih vrijednosti. Iz grana se postepeno dolazi do listova. Koriste se u statistici, rudarenju podataka i strojnom učenju.
- **Strojevi s potpornim vektorima (engl. *Support vector machines*) :**
Pripadaju vrsti nadziranog strojnog učenja, tj. uče na temelju zadanog skupa podataka za učenje. Skup se sastoji od unaprijed određenih parova vektora svojstava nekog objekta i njegove klasifikacije.
- **Regresijske analize (engl. *Regression analysis*) :**
Koriste se raznim statističkim metodama za procjenu odnosa između ulaznih podataka i izlaza. Najpoznatiji oblik je linearna regresija. Zavisna varijabla je u obliku linearne funkcije. Iz te se poznate funkcije određuju parametri modela, a iz njih se određuju nepoznate vrijednosti atributa novih podataka.
- **Bayesove mreže (engl. *Bayesian networks*) :**
Jednostavna grafička notacija za tvrdnje o uvjetnoj nezavisnosti i za sažetu specifikaciju zajedničke distribucije vjerojatnosti. Temelji se na Bayesovom pristupu koji kaže da je vjerojatnost stupanj uvjerenja da će se događaj s određenim ishodom dogoditi.

2.3. Pristupi rješavanju problema

Problemu prepoznavanja i klasifikacije motornih vozila može se pristupiti na nekoliko načina, a najvažniji su:

1. Metode oduzimanja pozadine (engl. *Background subtraction*) :
Uglavnom se koristi kod problema prepoznavanja pokreta. Kamera snima više sličica (engl. *frameova*) te ih analizira. Algoritam stvara model pozadine te se usporedbom s drugom sličicom dolazi do onoga što je u prvom planu. Ova se metoda koristi kod analize fotografija nepomične kamere. Razlog tome je što se analiza znatno otežava kretanjem kamere jer se samim time mijenja i pozadina pa je teško razlučiti gdje je došlo do promjene.

2. Metode temeljene na značajkama (engl. *Feature-based methods*) :

Metoda lokalnih značajki (engl. *local features method*) – sustav na početku pro-
nađe karakteristike jednog objekta ili grupe objekta. Zatim pokušava kategorizi-
rati pronađene karakteristike u različite klase putem klasifikacijskih modela. Na
temelju njih sustav odlučuje kojoj kategoriji objekt pripada.

Prednost : oblik i karakteristike ciljnih objekata poznate su prije detekcije.

Nedostatak : može prepoznati samo već viđene i naučene objekte.

3. Metode temeljene na pokretu i frameovima :

Metoda kliznog prozora (engl. *Sliding windows method*) - skenira ulaznu sliku
s puno prozora različitih veličina. Zatim analizira nalazi li se traženi objekt u
prozoru ili ne. Najpopularniji pristup zadnjih godina za detekciju, prepoznavanje
i lokalizaciju objekata.

3. Postojeće metode

Ulaskom u 21. stoljeće prepoznavanje i brojanje motornih vozila (ponajviše automobila) u prometu postaje aktivna tema istraživanja.

3.1. Metode temeljene na ručnom stvaranju klasifikacijskog sustava

Zhao T. i Nevatia R. 2001. godine razvili su metodu prepoznavanja automobila temeljenu na iskorištavanju sjene, intenzitetu boje i Bayesovoj mreži. Tu su metodu koristili za detekciju automobila na slikama iz zraka niske rezolucije[16].

Moon H., Chellappa R. i Rosenfeld A. predložili su tehniku zasnovanu na operateru kako bi se istaknuli rubovi objekata za otkrivanje vozila[4].

2007. godine Kluckner S., Pacher G. i suradnici koristili su Haarove značajke, lokalne binarne uzorke i orijentacijske histograme za otkrivanje automobila[12].

Dvije godine kasnije, Holt A. C., Seto, Rivard T. i Gong P. koristili su metodu temeljenu na objektima. Navedeni se rad usredotočuje na otkrivanje parkiranih vozila na fotografijama vrlo visoke rezolucije (VHR) pomoću nadziranog algoritma kliznog prozora[1].

Shao W., Yang W., Liu G. i Liu J. 2012. godine osmislili su novu metodu. Autori u navedenoj metodi prikupili su više obilježja, lokalne histograme gradijenta (HOG), lokalne binarne obrasce (LBP) i histograme za otkrivanje automobila[18].

Moranduzzo T. i Melgani F. 2014. godine predložili su metodu koja je koristila pretraživanje kliznog prozora, gdje su se operacije filtriranja u vodoravnom i okomitoj smjeru izvodile kako bi se izvukle karakteristike HOG-a. Automobili su otkriveni nakon izračuna mjere sličnosti koristeći katalog automobila kao referencu[13].

Isti autori sljedeće godine predstavili su metodu za otkrivanje objekata i primijenili je na problem detekcije automobila, ali i detekciju solarnih panela u urbanom okruženju. Metoda se temelji na gradijentima višeg reda i regresiji Gaussovog procesa (GP) za slike iz bespilotnih letjelica[14].

Xu Y., Yu. G, Wang Y., Wu X. i Ma Y. radili su na detekciji automobila u videozapisima sa snimki snimljenih iz bespilotnih letjelica te su predložili kombinaciju algoritma Viola – Jones i HOG + SVM algoritma koristeći strategiju prebacivanja detektora temeljenu na različitim silaznim trendovima brzine otkrivanja oba algoritma kako bi se poboljšala učinkovitost otkrivanja[21].

Navedene se metode temelje na ručnom stvaranju klasifikacijskog sustava. U novije vrijeme te su metode zamijenjene metodama dubokog učenja (npr. CNN).

3.2. Metode temeljene na dubokom učenju

U slučaju detekcije vozila, Chen X. i suradnici predložili su metodu koja se temelji na kliznim prozorima i dubokoj CNN nazvanoj hibridnom dubokom neuronskom mrežom (HDNN). Njihova je ideja kopirati slojeve u različitim mjerilima, omogućujući dubokoj mreži da prepozna vozila. HDNN zahtijeva nekoliko dana za treniranje prepoznavanja automobila pomoću grafičke procesorske jedinice (GPU). Kako bi se pronašli automobili u testnom slučaju, korišteno je modificirano pretraživanje kliznih prozora koje je pokušalo centrirati klizne prozore oko automobila[20].

Sokalski je 2010. koristio boju i detekciju rubova kako bi razdvojio i razlikovao umjetne od prirodnih objekata.

Wang i suradnici 2014. godine koristili su se detekcijom rubova u snimkama iz bespilotnih letjelica. Algoritam detekcije rubova koristi se za pronalazak ravnih linija koje oblikuju vozilo. Prije toga algoritam koristi threshold kako bi smanjio pozadinsku

buku na slici. Rad je bio prilagođen funkcioniranju u stvarnom vremenu te su se uz detekciju vozila, ona i brojala[10].

Obojica koriste boju i teksturu kako bi prepoznali objekte. Prepoznavanje bojom je uglavnom brže, ali tekstura sadrži više informacija pa je točnija. Sva istraživanja koriste nenađiranu klasifikaciju.

Saman G. (Kanistras, 2015) – koristi procjenu debljine. Kutni vektori utvrđeni su u rubnom dijelu povišenih slika. Prate se vektori nagiba i utvrđuje standardno odstupanje. Tako se unaprijednim definiranjem rubova mogu procijeniti prepoznata vozila. Ta se procjena koristila u Turskoj te se ostvarila točnost od 86% u F1-mjeri(4.2). Loša strana ove tehnike je to što se traženi objekt teže prepozna zbog različite okoline.

Peng W. (Wei, 2015) osmislio je metodu koja se rješava uličnih oznaka kako bi se odvojili nepredvidivi temelji. Za ovu metodu potrebna je fiksna kamera.

Susaki, Kubo 2015. – generiraju se dvije pozadinske slike – dugotrajna i kratkotrajna. Prednja slika generira se micanjem pozadine za svaki frame i pretvara se u binarne podatke. Zaustavljena vozila detektiraju se uz pomoć obje pozadinske slike, vozila u pokretu detektiraju se kratkotrajnom pozadinskom slikom i originalnom slikom. Tada se primjenom obrasca podudaranja otkriva regija vozila. Predlošci raznih vozila ručno se generiraju iz slika. Konačno, vozilo se pronalazi spajanjem zajedničkih područja koja su prepoznata između uzastopnih okvira. Ako se vozilo ne pronađe uzimaju se prethodni i sljedeći okviri. Metoda se isprobala u Hanoiju gdje je točnost bila 80%[3].

Tang T. i njegovi suradnici 2017. godine predložili su korištenje strukture YOLOv2 (You Only Look Once) za prepoznavanje vozila na snimkama iz bespilotnih letjelica[15]. Iste godine Radovic M. koristio je YOLO i konvolucijske neuronske mreže kako bi pronašao objekte na snimkama iz zraka[7]. Godinu kasnije Xu Z. i suradnici predložili su dublji YOLO model, DOLO, koji je koristio strukturu YOLOv2 modela[22]

Cao i njegovi suradnici razvili su metodu koja se temelji na zračnoj platformi. Ključni doprinos je poboljšanje pronalaska vozila pomoću nove značajke koja se zove HOG. Zatim se linearni SVM koristi za razvrstavanje. Videosnimke su snimljene u

gradskom prometu radi procjene predloženog sustava. Iako većina prizemnih sustava praćenja prometa postiže gotovo 99% točnosti za otkrivanje sustava (samo otkrivanje, ne i klasifikacija), predloženi sustav postigao je točnost otkrivanja vozila od 90%. Međutim, ovaj je sustav samo za otkrivanje vozila što najčešće nije dovoljno.

Zbog niske razlučivosti slike, mnogi sustavi klasifikacije vozila koji se temelje na zračnoj platformi ciljaju samo ograničen broj vrsta vozila, poput automobila i kamiona. Posebno se Liu i Mattus usredotočuju na poboljšanje brzine računanja za klasifikaciju vozila. Za otkrivanje vozila po zračnoj slici primjenjuje se binarni detektor kliznih prozora. Jednom kada se vozilo otkrije, karakteristike HOG se izvlače pomoću neuronske mreže s jednim skrivenim slojem. Vozila se zatim razvrstavaju u dvije vrste, tj. automobile i kamione. Točnost klasifikacije je bila visoka 98,2% zbog malog broja vrsta vozila za klasifikaciju.

Uz pomoć naprednih tehnika strojnog učenja poboljšava se točnost klasifikacije nekih klasifikacijskih vozila na bazi zračnih platformi. Ipak, rezultati nisu usporedivi s klasifikacijskim sustavima vozila temeljenim na zemaljskim senzorima. Tan i suradnici razvili su metodu klasifikacije vozila u dva koraka pomoću zračnih slika. Shema otkrivanja promjena primjenjuje se za otkrivanje vozila na temelju promjena na razini piksela predstavljenih u obliku toplinske karte. Zatim se za klasifikaciju primjenjuje standardni CNN. Konkretno, oni prihvaćaju potpuno povezani sloj modela AlexNet i završni sloj klasifikacije Inception modela. Eksperimenti su izvedeni sa slikama prikupljenim iz zrakoplova. Vozila su razvrstana u četiri klase: limuzine, kombije, kamioneti i kamioni. Prosječna točnost klasifikacije bila je 80,3%.

Audebert i suradnici također su primijenili standardni CNN na zračne slike za klasifikaciju vozila. Usvojeni su razni CNN modeli poput LeNet, AlexNet i VGG-16 koji su prethodno obučeni s postojećim skupima podataka o treningu. Da bi savladali razlike između skupova podataka o treningu i testiranja skupa podataka, autori koriste tehnike normalizacije i povećavanja podataka temeljene na geometrijskim operacijama, uključujući translaciju, zumiranje i rotaciju slika. Eksperimenti su izvedeni pomoću skupa podataka NZAM / ONERA razvrstavajući vozila u automobile, kombije, kamionete i kamione. Najveća točnost klasifikacije od 80% postignuta je modelom VGG-16.

Car-Rec je algoritam za prepoznavanje automobila u stvarnom vremenu, i u pokretu i dok stoje. Kombinira više algoritama za ekstrakciju poput Speed-Up Robust Features (SURF). Sustav je testiran na velikom broju slika igrački automobila. Jedna od prednosti Car-Rec okvira je modularnost, što znači da se ovaj okvir može unaprjeđivati zamjenom jednog algoritma za prepoznavanje slike drugim. Algoritam je testiran i na snimkama iz zraka u scenarijima s velikim gužvama[9].

Unazad zadnjih nekoliko godina izdvojile su se dvije metode bazirane na konvolucijskim neuronskim mrežama (CNN) koje kvalitetno funkcioniraju u stvarnom vremenu, a to su R-CNN i YOLO.

4. Odabrani algoritam

Sustav YOLO ima drugačiji pristup rješavanju problema od prijašnjih sustava. Na cijelu sliku primjenjuje jednu neuronsku mrežu. Ta mreža dijeli sliku na regije i predviđa okvire i vjerojatnosti za svaku regiju. Novonastali okviri dobivaju težine s obzirom na predviđene vjerojatnosti. Prednost YOLO modela u odnosu na sustave temeljene na klasifikatoru je to što YOLO za vrijeme testiranja gleda cijelu sliku. Njegova su predviđanja tako informirana cjelokupnim kontekstom sa slike. Prednost u odnosu na R-CNN je to što predviđa jedinstvenu mrežnu procjenu za razliku od R-CNN koji ih treba tisuće za svaku sliku. Te ga osobine čine izuzetno brzim, više od 1000 puta bržim od R-CNN i 100 puta bržim od poboljšane verzije originalnog algoritma - Brzog R-CNN (engl. *Faster R-CNN*)[11].

YOLOv3 je treća verzija sustava za detekciju objekata sa slike koja se sastoji od 53 konvolucijska sloja. Sustavi prethodne detekcije zamjenjuju klasifikatore ili lokalizatore za obavljanje detekcije. Oni primjenjuju model na sliku na više različitih mesta i skaliranja. Područja na kojima je dobivena visoka vjerojatnost objekata na slici smatraju se uspješnim pronalaskom objekta.

Jedna od inačica verzije YOLOv3 je YOLOv3-tiny. To je model koji je također stabilan i brz, ali je puno manji od originalne inačice. Sastoji se od 24 sloja, a njihov raspored i svojstva unutar mreže vidljivi su na slici 4.1. Navedeni ga razlozi čine optimalnim za predviđeni zadatak i iz tog je razloga YOLOv3-tiny implementiran u nastavku.

layer	filters	size/strd(dil)	input		output	
0 conv	16	3 x 3 / 1	608 x 608 x	3 ->	608 x 608 x	16 0.319 BF
1 max		2x 2 / 2	608 x 608 x	16 ->	304 x 304 x	16 0.006 BF
2 conv	32	3 x 3 / 1	304 x 304 x	16 ->	304 x 304 x	32 0.852 BF
3 max		2x 2 / 2	304 x 304 x	32 ->	152 x 152 x	32 0.003 BF
4 conv	64	3 x 3 / 1	152 x 152 x	32 ->	152 x 152 x	64 0.852 BF
5 max		2x 2 / 2	152 x 152 x	64 ->	76 x 76 x	64 0.001 BF
6 conv	128	3 x 3 / 1	76 x 76 x	64 ->	76 x 76 x	128 0.852 BF
7 max		2x 2 / 2	76 x 76 x	128 ->	38 x 38 x	128 0.001 BF
8 conv	256	3 x 3 / 1	38 x 38 x	128 ->	38 x 38 x	256 0.852 BF
9 max		2x 2 / 2	38 x 38 x	256 ->	19 x 19 x	256 0.000 BF
10 conv	512	3 x 3 / 1	19 x 19 x	256 ->	19 x 19 x	512 0.852 BF
11 max		2x 2 / 1	19 x 19 x	512 ->	19 x 19 x	512 0.001 BF
12 conv	1024	3 x 3 / 1	19 x 19 x	512 ->	19 x 19 x	1024 3.407 BF
13 conv	256	1 x 1 / 1	19 x 19 x	1024 ->	19 x 19 x	256 0.189 BF
14 conv	512	3 x 3 / 1	19 x 19 x	256 ->	19 x 19 x	512 0.852 BF
15 conv	30	1 x 1 / 1	19 x 19 x	512 ->	19 x 19 x	30 0.011 BF
16 yolo						[yolo] params: iou loss: mse (2), iou_norm: 0.75, cls_norm: 1.00, scale_x_y: 1.00
17 route	13			->	19 x 19 x	256
18 conv	128	1 x 1 / 1	19 x 19 x	256 ->	19 x 19 x	128 0.024 BF
19 upsample		2x	19 x 19 x	128 ->	38 x 38 x	128
20 route	19 8			->	38 x 38 x	384
21 conv	256	3 x 3 / 1	38 x 38 x	384 ->	38 x 38 x	256 2.555 BF
22 conv	30	1 x 1 / 1	38 x 38 x	256 ->	38 x 38 x	30 0.022 BF
23 yolo						

Slika 4.1: Raspored slojeva unutar mreže YOLOv3-tiny

4.1. Implementacija modela

Prvi korak u implementaciji je stvaranje baze fotografija koju čine fotografije vozila iz zraka. Za navedenu su bazu kombinirane tri različite javno dostupne baze fotografija s interneta [5, 8, 6]. Novonastalu bazu koja je korištena za ovaj rad čini 1357 fotografija[17]. Osim fotografija definirane su klase koje će model kasnije znati prepoznati. Skup mogućih klasa s navedenog skupa fotografija preopširan je za potrebe ovoga rada (*car, truck, pickup, tractor, camping car, boat, bus, van, plane, minibus, other*). Iz tog razloga popis klasa smanjen je na: *car, truck, bus, van i other*. *Other* čine klase *camping car* i *minibus* koje su gledano iz zraka sličnog oblika, ali zbog nedovoljnog broja primjera navedenih klasa one se odvojeno ne bi mogle dovoljno kvalitetno istrenirati. Broj oznaka za određene klase nalazi se u tablici 4.1 Skup fotografija podijeljen je na dva skupa. Jedan dio čini skup za treniranje (80%, tj. 1086 fotografija), a drugi čini skup za validaciju (20%, tj. 271 fotografija).

Tablica 4.1: Broj oznaka na skupu podataka za određene klase

Ime klase	Broj oznaka
Car	6489
Truck	320
Bus	107
Van	111
Other	518

Sukladno odabiru YOLOv3-tiny konvolucijske mreže, stvorene su i oznake (engl. *labels*) za svaku korištenu fotografiju. Svaka neuronska mreža ima poseban način označavanja objekata. U YOLO konvolucijskoj mreži potreban je par fotografija-oznaka. Oznake čine tekstualne datoteke u kojima se za svaki detektirani objekt s pripadne fotografije nalazi jedan red teksta. Svaki red u tekstualnoj datoteci je oblika:

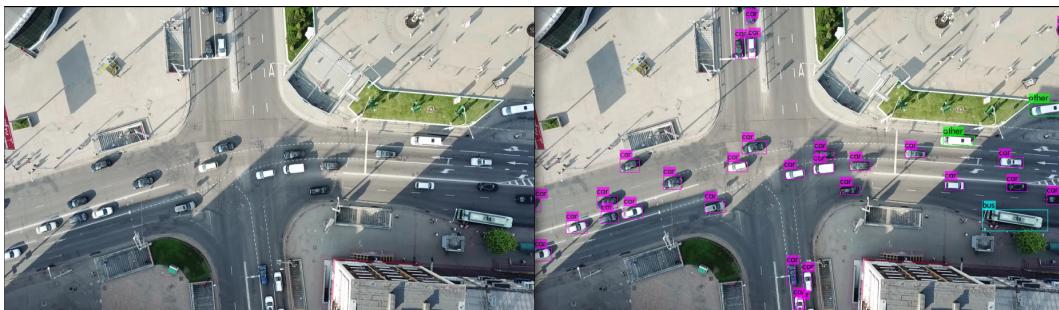
$$< \text{class_name} > < \text{center_x} > < \text{center_y} > < \text{width} > < \text{height} > .$$

Class_name je zadani broj koji predstavlja klasu (npr. car = 0). Center_x i center_y su koordinate sredine zadanog objekta. Width i height su visina i širina pravokutnika koji označava objekt. Sve koordinate gledaju se u odnosu na koordinatni sustav slike kojem u gornja-ljeva točka ima koordinate (0, 0), a donja-desna (1, 1).

Mreža je trenirana na Nvidia Jetson TX2 koji se nalazi u laboratoriju Fakulteta elektrotehnike i računarstva u Zagrebu. Za početak treniranja korišteni su unaprijed trenirane težine (engl. *weights*) i konfiguracijska datoteka (konfiguracije za treniranje mreže, npr. broj klasa, broj slojeva, rezolucija...) koja je prilagođena za detekciju zadanih vozila.

4.2. Rezultati

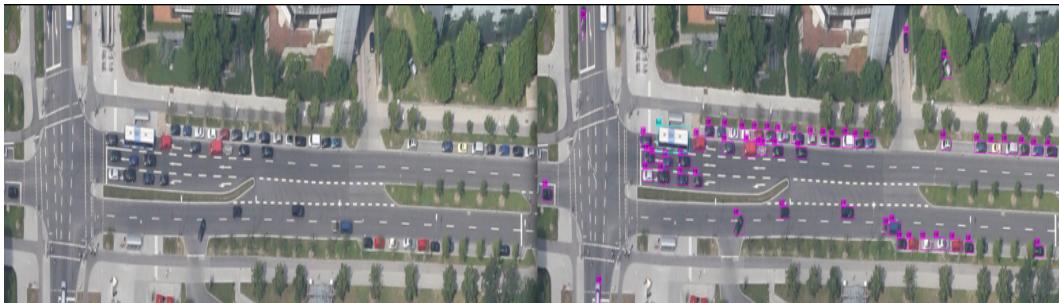
Mreža je trenirana otprilike 2 tjedna, a u tom je razdoblju validacija napravljena nekoliko puta na trenutnim težinama kako bi se napravila usporedba rezultata. Prvi stupac tablice 4.8 čine mjere, a ostali stupci čine dobiveni rezultati validacija za pojedine težine (npr. prvi stupac su mjere nakon 1000 iteracija, drugi nakon 14000 itd.)



Slika 4.2: Primjer 1. originalne slike i slike nakon testiranja



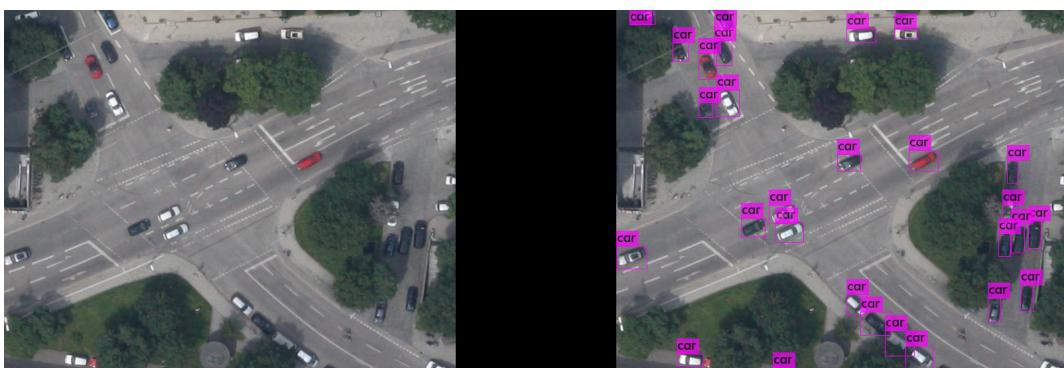
Slika 4.3: Primjer 2. originalne slike i slike nakon testiranja



Slika 4.4: Primjer 3. originalne slike i slike nakon testiranja



Slika 4.5: Primjer 4. originalne slike i slike nakon testiranja



Slika 4.6: Primjer 5. originalne slike i slike nakon testiranja



Slika 4.7: Primjer 6. originalne slike i slike nakon testiranja

Vrednovanje modela:

- **Detection count :**
broj pronađenih objekata za vrijeme testiranja na skupu podataka za validaciju
- **Unique truth count :**
stvarni broj objekata na skupu podataka za validaciju
- **True positives (TP) :**
broj objekata koji su pronađeni za vrijeme testiranja na skupu za validaciju i stvarno postoje
- **False positives (FP) :**
broj objekata koji su pronađeni za vrijeme testiranja na skupu za validaciju, a zapravo ne postoje
- **False negatives (FN) :**
broj objekata koji nisu pronađeni za vrijeme testiranja na skupu za validaciju, a zapravo postoje
- **Preciznost (engl. Precision) :**
udio pozitivnih primjera u skupu pozitivno klasificiranih primjera

$$P = \frac{TP}{TP + FP}$$

- **Odziv (engl. Recall) :**
udio pozitivno klasificiranih primjera u skupu svih pozitivnih primjera:

$$R = \frac{TP}{TP + FN}$$

– **F1-score :**

harmonijska sredina preciznosti i odziva:

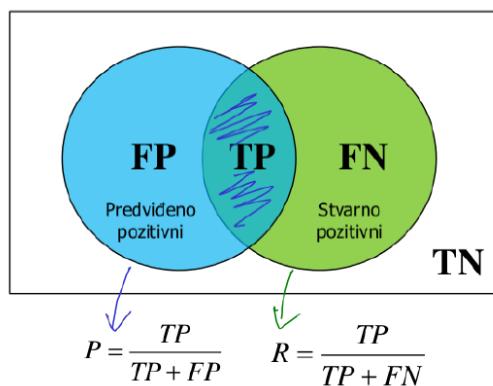
$$F_1 = \frac{2 * P * R}{P + R}$$

– **Mean average precision (mAP) :**

prosječna preciznost svih preciznosti za određenu klasu:

$$mAP = \frac{\sum_{i=1}^K AP_i}{K}$$

gdje je K broj klasa, a AP_i prosječna preciznost pojedine klase.



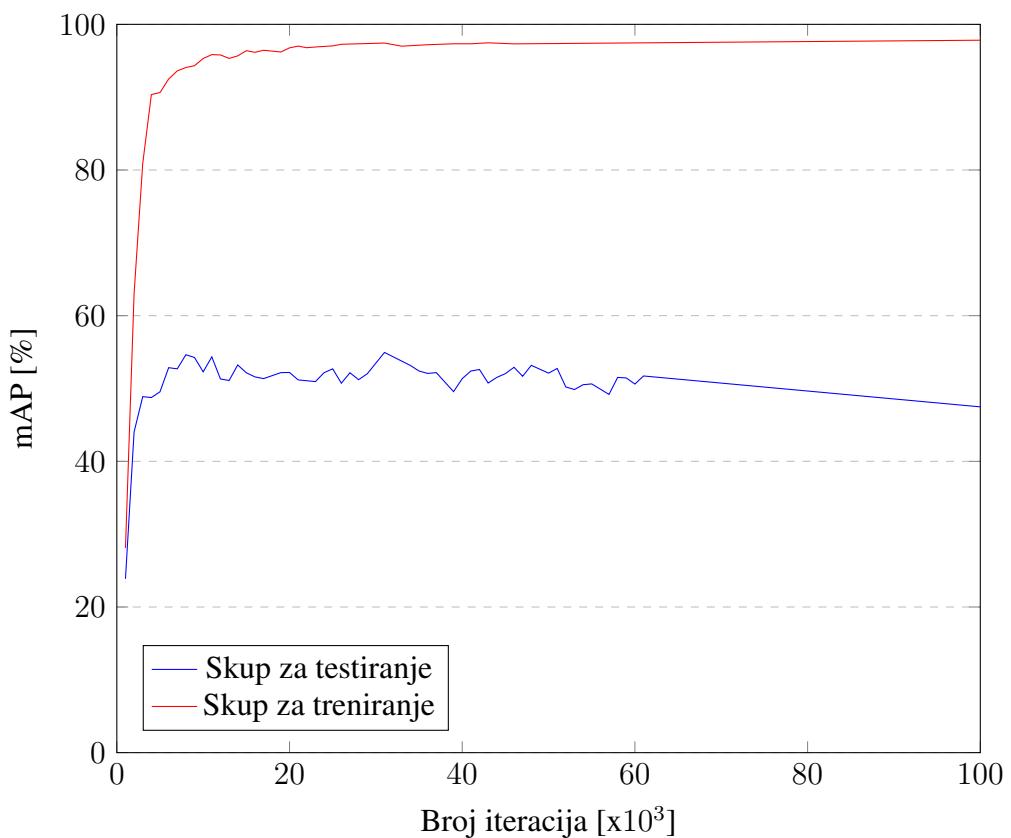
Slika 4.8: Osnovne metrike - preciznost i odziv [23]

Tablica 4.2: Metrike nakon određenog broja iteracija

Mjera	1000	14000	31000	53000	100000
<i>Detection count</i>	9173	2921	2657	2147	1914
<i>Unique truth count</i>	1315	1315	1315	1315	1315
<i>TP</i>	702	1018	1031	1020	1021
<i>FP</i>	491	236	218	188	178
<i>FN</i>	613	297	284	295	294
<i>Precision</i>	59%	81%	83%	84%	85%
<i>AP – car</i>	55.82%	83.94%	86.08%	86.3%	86.43%
<i>AP – truck</i>	12.72%	34.69%	36.57%	33.97%	27.01%
<i>AP – bus</i>	22.39%	85.24%	84.35%	80.95%	80.95%
<i>AP – van</i>	1.22%	20.46%	22.37%	9.8%	6.70%
<i>AP – other</i>	27.18%	42.73%	45.38%	38.3%	36.3%
<i>Recall</i>	53%	77%	78%	78%	78%
<i>F1 – score</i>	56%	79%	80%	81%	81%
<i>mAP</i>	23.87%	53.41%	54.95%	49.86%	47.48%

Povećanjem broja iteracija na slici 4.9 primjećuje se očekivano ponašanje prosječne preciznosti u odnosu na broj iteracija. Na početku treniranja prosječna preciznost je mala i na skupu za treniranje i na skupu za testiranje. U nastavku treniranja obje preciznosti rastu do određenog trenutka. Taj trenutak nastupa kada model dođe do optimalnih težina i tada on najbolje predviđa na skupu još neviđenih podataka. Preciznost skupa podataka nakon te točke svejedno nastavlja rasti jer se model zapravo testira na podacima kojima je treniran pa on zapravo "uči napamet". Iz istog razloga preciznost skupa podataka za testiranje koji još nisu viđeni kreće opadati. Model se do točke optimalne složenosti nalazi u fazi podnaučenosti, a nakon nje u fazi prenaučenosti, tj. kaže se da model loše generalizira.

Na grafu ovisnosti mAP-a o iteracijama vidi se da je mAP većinu vremena oko 50%. Prosječna preciznost je najveća nakon 31000 iteracija kada iznosi 54.95%. Veći postotak nije dobiven iz razloga što postoji 5 klase. Klase *car* i *bus* su istrenirane s preciznošću od otprilike 85%, a ostale 3 klase s puno manjom (*van* 22%, *truck* 37%, *other* 45%). Razlog manjeg postotka navedene 3 klase je manji broj primjera istih u skupu podataka za treniranje.



Slika 4.9: Ovisnosti mAP-a o iteraciji na skupu za treniranje i testiranje

4.3. Gustoća prometa

Sljedeći je korak određivanje gustoće prometa analizom dobivenih slika i detekcije. Gustoća prometa računata je kao omjer piksela koji na slici čine vozila i piksela koji čine cestu. Broj piksela ceste dobiven je ručno i nije predmet ovoga rada pa se taj postupak neće detaljnije objašnjavati. Broj piksela vozila dobiven je zbrajanjem svih dobivenih oznaka vozila na toj slici. U YOLO formatu označavanja nalaze se širina i visina svakog označenog objekta. Površina pojedinačnog vozila je umnožak njegove širine i visine skaliranih ovisno o rezoluciji slike. Formula za gustoću prometa glasi

$$g = \frac{\sum_{i=1}^K P_i}{P_c},$$

gdje je K broj označenih vozila na slici, P_i površina oznake pojedinog vozila, a P_c ukupna površina ceste.

Slika 4.2 ima rezoluciju $1920x1080$ piksela pa je tako ukupan broj piksela na slici 2 073 600 piksela. Analizom navedene slike dobivene su vrijednosti:

$$\sum_{i=1}^K P_i = 94\,474,08 \quad P_c = 831\,106,76 .$$

Gustoća prometa na toj slici iznosi

$$g = \frac{94\,474,08}{831\,106,76} = 11.37\% .$$

5. Zaključak

U ovome je radu prikazana teorijska podloga područja strojnog učenja te su ukratko objašnjene neuronske mreže. Predstavljen je detaljan pregled pristupa za rješavanje problema koji su u zadnjih dvadeset godina bili korišteni za prepoznavanje i klasifikaciju vozila na snimkama iz zraka. Na temelju proučenih podataka iz dosadašnjih istraživanja, za izvedbu ovoga rada odabran je optimalan algoritam, YOLOv3-tiny. Ukratko je objašnjen, a zatim i implementiran.

Za treniranje modela korišteni su kombinirani, javno dostupni skupovi podataka. Cilj mreže bio je prepoznati i klasificirati odabrane klase vozila, prvenstveno za korištenje na autocestama. Model je kasnije testiran na skupu podataka za testiranje, a završne mjere dale su prosječnu preciznost (engl. *mean Average Precision*) od 54.95%.

U eventualnom dalnjem radu preporučuje se napraviti skup podataka za testiranje s većim i približno podjednakim brojem oznaka svih klasa.

LITERATURA

- [1] T. Rivard P. Gong A.C. Holt, E.Y.W. Seto. *Object-based Detection and Classification of Vehicles from High-resolution Aerial Photography*. American Society for Photogrammetry and Remote Sensing, 2009. doi: 10.14358/PERS.75.7.871.
- [2] J. Šnajder D. Bašić. Umjetna inteligencija: Strojno učenje. URL https://www.fer.unizg.hr/_download/repository/UI-2020-10-StrojnoUcenje.pdf.
- [3] J. Susaki H. Kubo. Automatic detection and tracking of motorcycles and cars from mixed traffic flow. 2012.
- [4] A. Rosenfeld H. Moon, R. Chellappa. Performance analysis of a simple vehicle detection algorithm. *Image and Vision Computing*, 20:1–13, 2002.
- [5] Yauhen Kharuzhy. aerial-cars-dataset. URL <https://github.com/jekhor/aerial-cars-dataset>.
- [6] Anis Koubaa. car-detection-yolo-faster-rcnn-uvsc2019. URL https://github.com/aniskoubaa/car_detection_yolo_faster_rcnn_uvsc2019.
- [7] Q. Wang M. Radovic, O. Adarkwa. Object recognition in aerial images using convolutional neural networks. *Journal of Imaging*, 3(21), 2017. doi: 10.3390/jimaging3020021.
- [8] Nikita Popov. vedai. URL <https://github.com/nikitalpopov/vedai>.
- [9] L. Edwar George R. A. Hadi, G. Sulong. Vehicle detection and tracking techniques: A concise review. *CoRR*, abs/1410.5894, 2014. URL <http://arxiv.org/abs/1410.5894>.

- [10] K. Xiao R. Sun L. Cui R. Wang, L. Zhang. Easisee: Real-time vehicle classification and counting via low-cost collaborative sensing. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):414–424, 2014.
- [11] Joseph Redmon i Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.
- [12] H. Grabner H. Bischof J. Bauer S. Kluckner, G. Pacher. A 3d teacher for car detection in aerial images. *2007 IEEE 11th International Conference on Computer Vision*, stranice 1–8, 2007.
- [13] F. Melgani T. Moranduzzo. Detecting cars in uav images with a catalog-based approach. *IEEE Transactions on Geoscience and Remote Sensing*, 52(10):6356–6367, 2014.
- [14] Y. Bazi N. Alajlan T. Moranduzzo, F. Melgani. A fast object detector based on high-order gradients and gaussian process regression for uav images. *International Journal of Remote Sensing*, 36(10):2713–2733, 2015. doi: 10.1080/01431161.2015.1043760.
- [15] S. Zhou L. Lei H. Zou T. Tang, Z. Deng. Fast vehicle detection in uav images. In *2017 International Workshop on Remote Sensing with Intelligent Processing (RSIP)*, stranice 1–5, 2017.
- [16] R. Nevatia T. Zhao. Car detection in low resolution aerial image. 2001.
- [17] Dominik Vrbanić. github-repository. URL <https://github.com/dvrbanic/zavrski-rad>.
- [18] G. Liu J. Liu W. Shao, W. Yang. *Car detection from high-resolution aerial imagery using multiple feature*. 2012.
- [19] wikipedia. Neural networks. URL https://en.wikipedia.org/wiki/Artificial_neural_network.
- [20] C. Liu C. Pan X. Chen, S. Xiang. Vehicle detection in satellite images by hybrid deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 11(10):1797–1801, 2014.
- [21] Y. Wang X. Wu Y. Ma Y. Xu, G. Yu. A hybrid vehicle detection method based on viola-jones and hog + svm from uav images. *Sensors*, 16(8):1325, 2016. doi: 10.3390/s16081325.

- [22] N. Li C. Xiang H. Zhou Z. Xu, H. Shi. Vehicle detection under uav based on optimal dense yolo method. U *2018 5th International Conference on Systems and Informatics (ICSAI)*, stranice 407–411, 2018.
- [23] Jan Šnajder. Strojno učenje: Vrednovanje modela. 2019.

Prepoznavanje i klasifikacija motornih vozila na snimkama iz zraka

Sažetak

U ovome su radu nabrojane i ukratko objašnjene postojeće metode za prepoznavanje i klasifikaciju motornih vozila na snimkama iz zraka zasnovane na strojnom učenju. Odabrana je konvolucijska neuronska mreža YOLOv3-tiny te su za implementaciju iste napravljeni skup fotografija i njihovih oznaka koje se nalaze u GitHub repozitoriju. Trenirani model testiran je na neviđenim podacima i ostvareni su zadovoljavajući rezultati.

Ključne riječi: strojno učenje, stvarno vrijeme, bespilotna letjelica, konvolucijske neuronske mreže, YOLOv3-tiny

Recognition and classification of motorized vehicles on aerial footages

Abstract

This work enumerates and briefly explains the existing methods for recognition and classification of motorized vehicles on aerial footages based on machine learning. The convolutional neural network YOLOv3-tiny was selected and a set of images and their labels in the GitHub repository were made for its implementation. The model was trained and the test results were presented.

Keywords: machine learning, real-time, unmanned aerial vehicle, convolutional neural network, YOLOv3-tiny