Poboljšanje performansi detektora objekata dodavanjem podataka o dubini slike koristeći oblak točaka

Zmeškal, Ivan

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:786921

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

POBOLJŠANJE PERFORMANSI DETEKTORA OBJEKATA DODAVANJEM PODATAKA O DUBINI SLIKE KORISTEĆI OBLAK TOČAKA

Diplomski rad

Ivan Zmeškal

Osijek, 2022.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 03.12.2022.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime Pristupnika:	Ivan Zmeškal
Studij, smjer:	Diplomski sveučilišni studij Automobilsko računarstvo i komunikacije
Mat. br. Pristupnika, godina upisa:	D-56ARK, 11.10.2020.
OIB studenta:	87430122580
Mentor:	Izv.prof.dr.sc. Ratko Grbić
Sumentor:	,
Sumentor iz tvrtke:	Borna Jelić
Predsjednik Povjerenstva:	Prof. dr. sc. Marijan Herceg
Član Povjerenstva 1:	Izv.prof.dr.sc. Ratko Grbić
Član Povjerenstva 2:	Izv. prof. dr. sc. Mario Vranješ
Naslov diplomskog rada:	Poboljšanje performansi detektora objekata dodavanjem podataka o dubini slike koristeći oblak točaka
Znanstvena grana diplomskog rada:	**Umjetna inteligencija (zn. polje računarstvo)**
Zadatak diplomskog rada:	U okviru diplomskog rada potrebno je istražiti načine koliko informacija o dubini može poboljšati standardnu 2D detekciju objekata u prometu u RGB slikama dobivenim s kamere montirane na prednjoj strani vozila. Najprije je potrebno proučiti senzore koji daju informaciju o dubini, a prikladni su za automotiv primjenu (LiDAR, stereo kamera), a zatim pronaći i prikladne skupove podataka za izradu detektora objekata. Potrebno je implementirati ukupno tri detektora objekata - jedan koji detektira objekte na temelju RGB slike, drugi koji detektira objekte na temelju oblaka točaka te treći koji koristi RGB-D slike koje su dobivene fuzijom podataka RGB slika i oblaka točaka te ih evaluirati na testnom skupu podataka. Tema rezervirana za: Ivan Zmeškal Sumentor iz
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Dobar (3)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 1 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 2 razina
Datum prijedloga ocjene od strane mentora:	03.12.2022.

Potvrda mentora o predaji konačne verzije rada:	*Mentor elektronički potpisao predaju konačne verzije.*
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 21.12.2022.

Ime i prezime studenta:	Ivan Zmeškal
Studij:	Diplomski sveučilišni studij Automobilsko računarstvo i komunikacije
Mat. br. studenta, godina upisa:	D-56ARK, 11.10.2020.
Turnitin podudaranje [%]:	3

Ovom izjavom izjavljujem da je rad pod nazivom**: Poboljšanje performansi detektora objekata dodavanjem podataka o dubini slike koristeći oblak točaka**

izrađen pod vodstvom mentora Izv.prof.dr.sc. Ratko Grbić

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD

Automobilska industrija je kroz nekoliko prethodnih desetljeća znatno napredovala. Uz neprekidan razvoj tehnologije, neizbježan je bio i razvoj područja umjetne inteligencije. Umjetna inteligencija postaje sve manja nepoznanica te u skladu s tim postaje i sve više korištena u gotovo svim granama industrije, pa tako i u automobilskoj. Primjenom umjetne inteligencije u vozilima je omogućen porast razine autonomije, dok je konačan cilj potpuno autonomno vozilo. Takvo vozilo bi, osim što bi uklonilo potrebu za vozačem, omogućilo i povećanje sigurnosti za sudionike u vožnji, budući da uklanja ljudski faktor i u prometu općenito. U slučaju autonomnog vozila potreban je skladan rad različitih sustava koji koriste informacije s različitih senzora pomoću kojih bi takvo vozilo bilo sposobno i pouzdano za sudjelovanje u prometu.

U današnjim automobilima često je prisutan jedan ili više sustava koji pomažu vozaču u vožnji i čine je ugodnijom. Takvi sustavi nazivaju se napredni sustavi za pomoć vozaču u vožnji (engl. *Advanced Driver Assistance System* – ADAS). Ovi sustavi temelje svoj rad na informacijama koje dobivaju sa senzora montiranih na vozilo poput kamera, RADAR-a (engl. *Radio detection and ranging*) i LiDAR-a (engl. *Light Detection and Ranging*). Najvažniji senzori su kamere na prednjoj strani vozila pomoću kojih se vrši detekcija objekata, prometnih znakova, pješaka i uzdužnih kolničkih oznaka ispred vozila na cesti te su osnova mnogih ADAS sustava poput sustava za automatsko kočenje, sustava za upozoravanje na napuštanje vozne trake i sl.

ADAS sustavi čine važnu ulogu u povećanju sigurnosti tijekom vožnje. Mnogi se temelje na radu dubokih neuronskih mreža koje za ulaznu sliku daju lokacije objekata od interesa u obliku graničnih pravokutnika. Za potrebe izrade detektora objekata mogu biti korišteni YOLO detektori čiji se rad zasniva na konvolucijskim neuronskim mrežama. Konvolucijske neuronske mreže su prikladne za izradu detektora objekata jer su se pokazale kao dobar izbor u različitim problemima u području računalnog vida. Nad takvim detektorima objekata je potrebno izvršiti proces treniranja koristeći podatkovne skupove koji sadrže RGB slike gdje su objekti od interesa (vozila, pješaci, prometni znakovi, i sl.) označeni graničnim pravokutnicima.

U okviru ovog diplomskog rada cilj je istražiti može li dodavanje informacije o dubini slike poboljšati performanse standardnih detektora objekata temeljenih na RGB slikama. Korišteni detektori su implementacije treće i četvrte verzije YOLO algoritma. Informacija o dubini slike je izvornim ulaznim slikama dodana kao još jedan kanal, pri čemu su korišteni

LIDAR podatci prikazani kao dubinske mape. Navedeni detektori su ispitani korištenjem KITTI i A2D2 skupova podataka koji osim RGB slika sadrže i LiDAR podatke.

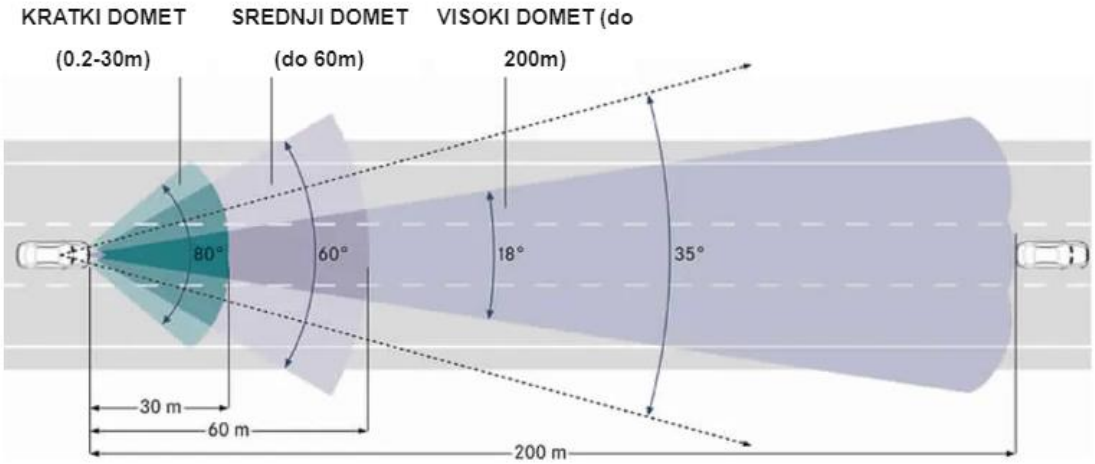
Detaljniji opis izvornog YOLO algoritma, ali i YOLOv3 i YOLOv4 algoritama je dan u drugom poglavlju. Uz osnovna obilježja, navedene su i najznačajnije razlike između njih. U trećem poglavlju je opisano radno okruženje, detaljan opis pripreme skupova podataka i konačne postavke korištenih detektora objekata. Četvrto poglavlje sadrži postignute rezultate i na kraju rada je dan zaključak.

2. DETEKCIJA OBJEKATA I DOSTUPNI SKUPOVI PODATAKA

U ovom poglavlju detaljnije su opisani načini prikupljanja informacija na temelju kamere i LiDAR senzora u području autonomne vožnje. Zatim je dan pregled često korištenih detektora objekata koji koriste podatke dobivene kamerom i LiDAR senzorom. Nakon pregleda detektora, dan je pregled KITTI i A2D2 skupova podataka koji sadrže obje vrste podataka te su korišteni za izradu ovog rada.

2.1. Kamere za ADAS

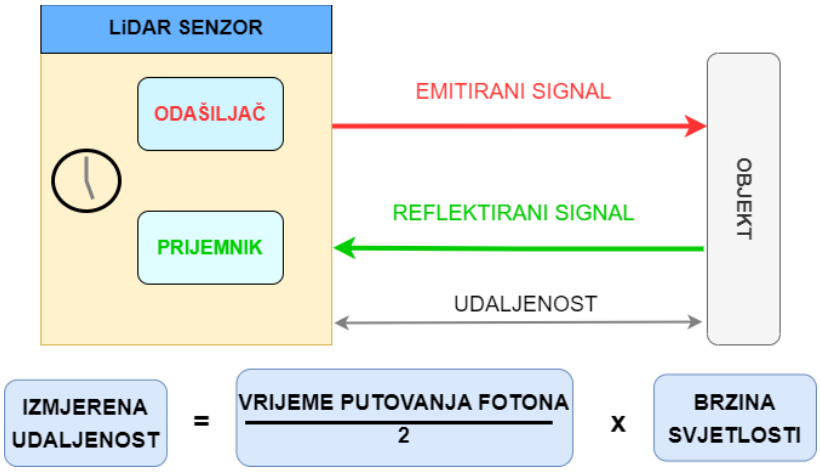
Upotreba kamera u automobilskoj industriji već godinama nije novost, samo se svrha i način na koji su korištene mijenjao tijekom godina. Osim što podaci dobiveni kamerama pružaju informacije ljudskom vozaču o njegovoj okolini tijekom vožnje, također mogu biti izvor informacija i za ADAS sustave. Od najvećeg značaja za autonomnu vožnju su prednje kamere srednjeg i visokog dometa, do 200 metara. Kamere srednjeg dometa su korištene u algoritmima za upozoravanje vozača o nailasku na križanje, pješake, u sustavima za detekciju voznih traka i sl. Kamere visokog dometa koriste se s algoritmima za automatsku detekciju i klasifikaciju objekata te određivanje udaljenosti do njih. Glavna razlika između kamera srednjeg i visokog dometa je u kutu otvora objektiva, odnosno vidnom polju (prikazano na slici 2.1.). Za sustave srednjeg dometa korištene su kamere s horizontalnim vidnim poljem postavljenim u rasponu od 70° do 120°, dok se vidno polje kamere korištene u sustavima visokog dometa postavlja na horizontalne kuteve od približno 35° [1]. U ovom radu su korišteni skupovi podataka snimljeni kamerama visokog dometa.



Sl. 2.1. *Prikaz vidnog polja ovisno o dometu prednje kamere* [2]

2.2. LiDAR

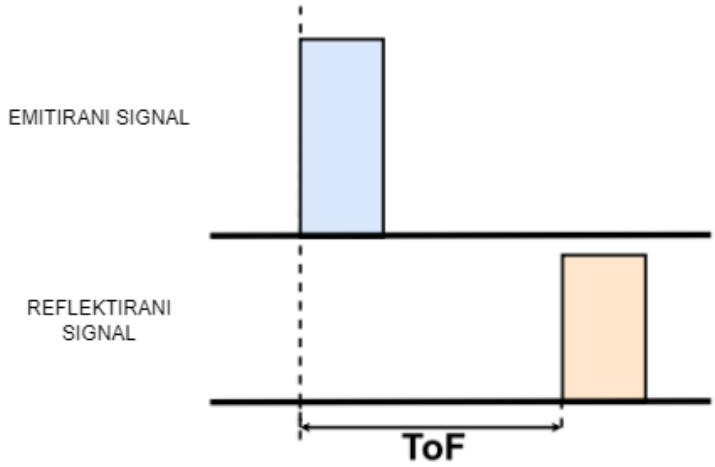
Kao što je prethodno spomenuto, LiDAR je sustav za mjerenje udaljenosti koji se već godinama koristi u industriji u svrhu određivanja udaljenosti na nekoliko metara. Međutim, za primjenu u automobilskom sektoru mora ispuniti zahtjev za sveobuhvatnom vidljivošću od 360° kako bi uspješno snimao prostorne slike svih objekata u okolini vozila u svim mogućim vremenskim uvjetima. Dvije ključne komponente LiDAR sustava su odašiljač i visokoosjetljivi prijemnik. Odašiljač se sastoji od lasera koji emitira svjetlost valnih duljina između 250 nanometara (nm) i 1600 nm, valna duljina lasera se odabire u ovisnosti o radnom okruženju i o objektima prema kojima je svjetlost emitirana, dok je prijemnik odgovoran za prikupljanje, obradu i izvršavanje proračuna na temelju odbijene svjetlosne zrake. Dijelovi prijemnika koji su potrebni za njegovo ispravno djelovanje su teleskop – zadužen za prikupljanje fotona, optički analizator koji omogućuje filtriranje svjetlosti određene valne duljine te pretvara optički signal u električnu veličinu, i modul za prikupljanje podataka koji izračunava vrijeme potrebno svjetlosnom impulsu za povratak do prijemnika te prikuplja podatke i pohranjuje informacije [3]. Na slici 2.2. je prikazan pojednostavljen prikaz principa rada LiDAR senzora.



Sl. 2.2. *Princip rada LiDAR senzora* [3]

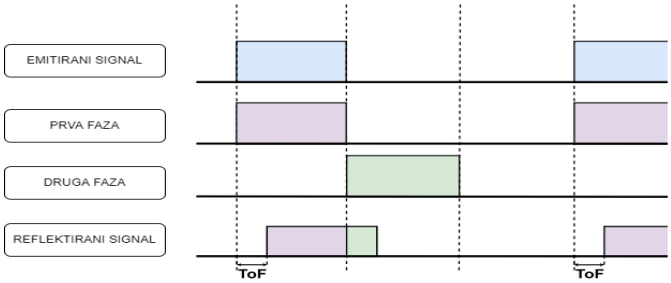
Rad LiDAR senzora temelji se na emitiranju svjetlosnog signala te zabilježavanju reflektiranog signala. Postoje tri glavne metode kojima se određuje udaljenost između senzora i promatranog objekta: mjerenje vremena leta između pulsiranog emitiranog i reflektiranog signala, zatim mjerenje vremena leta uspoređivanjem amplitude te mjerenje vremena leta uspoređivanjem frekvencije emitiranog i reflektiranog signala.

Mjerenje vremena leta između pulsiranog emitiranog i reflektiranog signala, prikazano na slici 2.3., (engl. *Pulsed Time of Flight – ToF*) se odnosi na mjerenje vremenskog intervala potrebnog za povratni put signala između senzora i objekta. Iznos vremena kašnjenja, do kojeg dolazi prilikom povratnog puta signala, je dobiveno korištenjem preciznih vremenskih mjernih uređaja (engl. *time-to-digital converter*) koji se mogu koristiti i kod jednostavnijih sustava odašiljača i prijemnika te su prepoznati kao povoljno rješenje bez dodatnih složenih optičkih sustava (leće ili dodatni senzorski sklopovi).



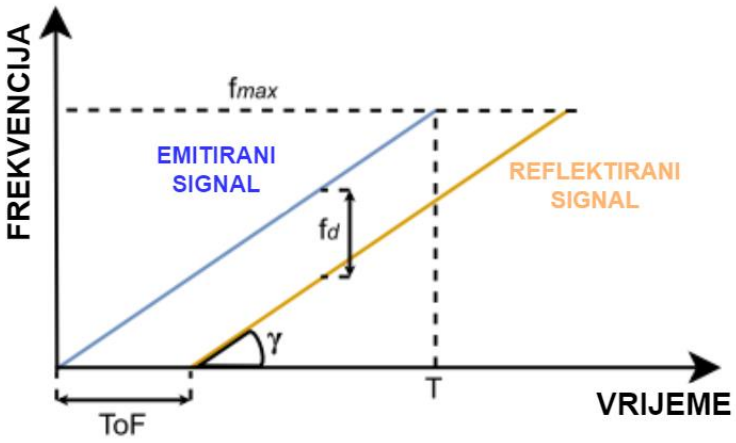
Sl. 2.3. *Mjerenje leta između pulsiranog emitiranog i reflektiranog signala* [3]

Mjerenje vremena leta uspoređivanjem amplitude emitiranog i reflektiranog signala, prikazano na slici 2.4., (engl. *Amplitude Modulated Continous Wave – AMCW*) kontinuirano emitira snop svjetlosnih signala, za razliku od prethodnog načina gdje se signal emitira u impulsima. Prvo senzor emitira signal sinkroniziran s prvim prozorom, koji će se reflektirati izvan faze. Nakon što se signal emitira, omogućen je drugi prozor te se na temelju omjera između oba prozora, koji zadržavaju reflektirani signal, određuje vrijeme leta signala.



Sl. 2.4. *Mjerenje leta uspoređivanjem amplitude emitiranog i reflektiranog signala* [4]

Mjerenje vremena leta uspoređivanjem frekvencije emitiranog i reflektiranog signala, prikazano na slici 2.5., (engl. *Frequency Modulated Continous Wave – FMCW*) određuje se koristeći frekvenciju emitiranog signala. Frekvencija emitiranog signala je linearno modulirana u vremenu sa signalom naviše (engl. *up-chirp signal*), pri čemu su mjerenja ograničena na povratna kašnjenja kraća od perioda signala. Kašnjenje između emitiranog i reflektiranog signala uzrokuje konstantnu razliku u frekvenciji koja je izravno proporcionalna udaljenosti do određenog objekta.



Sl. 2.5. *Mjerenje vremena leta uspoređivanjem frekvencije emitiranog i reflektiranog signala* [3]

2.3. Detekcija objekata pomoću YOLO algoritma

2.3.1. Izvorni YOLO algoritam

YOLO (engl. *You Only Look Once*) je algoritam za detekciju objekata razvijen 2016. godine [5] koji se osim za potrebe autonomne vožnje primjenjuje i u raznim drugim područjima poput sigurnosnih sustava za detekciju i identifikaciju osoba, u slučaju detektiranja i prepoznavanja životinja u šumama prilikom snimanja dokumentarnih filmova o prirodi [6] i sl.

Detekcija YOLO algoritmom se izvodi kao regresijski problem i radi tako da ulaznu sliku dijeli na rešetku sa *SxS* ćelija, gdje je *S* cijeli broj. Za svaku ćeliju predviđa granične pravokutnike (engl. *bounding boxes*) i vjerojatnost da je unutar nekog graničnog pravokutnika pretpostavljeni objekt. Vjerojatnost se određuje kao omjer površine presjeka detektiranog i stvarnog graničnog pravokutnika te unije površina detektiranog i stvarnog graničnog pravokutnika (engl. *Intersection over Union*):

$$IoU = \frac{A \cap B}{A \cup B}, \tag{2-1}$$

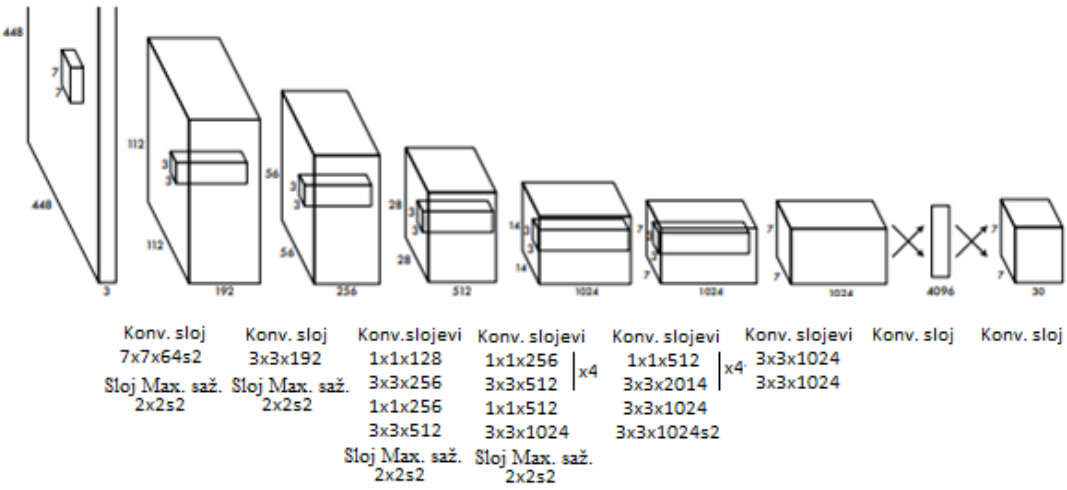
gdje je:

- IoU – *Intersection over Union*,
- A – površina detektiranog graničnog pravokutnika,
- B – površina stvarnog graničnog pravokutnika (engl. *ground truth*).

Općenito govoreći, svaki granični pravokutnik je opisan sa pet vrijednosti – predikcija: *x*,*y*,*w*,*h* i vjerojatnost predikcije. Predikcije *x* i *y* predstavljaju koordinate središta graničnog pravokutnika, a *w* i *h* su širina i visina prikazanog objekta. Vjerojatnost predikcije je predstavljena kao *IoU* između predviđenog graničnog pravokutnika s dimenzijama (*x*, *y*, *w*, *h*) i stvarnog graničnog pravokutnika s dimenzijama (x_g, y_g, w_g, h_g). Primjer stvarnog i predviđenog graničnog pravokutnika je prikazan na slici 2.6.

Sl. 2.6. *Prikaz stvarnog graničnog pravokutnika (zeleno na slici) i predviđenog graničnog pravokutnika (crveno na slici)* [7]

Arhitektura YOLO algoritma sastoji se od 24 sloja konvolucijske neuronske mreže (engl. *convolutional neural network*) koji služe za izdvajanje značajki iz ulazne slike. Osim navedena 24 sloja, YOLO sadrži i dva potpuno povezana sloja koji su zaduženi za predviđanje izlaznih vjerojatnosti i koordinata graničnog pravokutnika. Arhitektura izvornog YOLO modela je prikazana na slici 2.7.



Sl. 2.7. *Prikaz arhitekture YOLO modela* [8]

2.3.2. YOLOv3

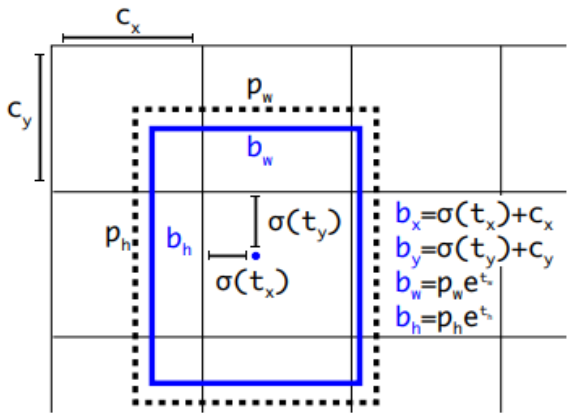
YOLOv3 u odnosu na izvorni YOLO sadrži veći broj konvolucijskih slojeva, 53 prema 24 sloja izvornog algoritma. Za predviđanje graničnih pravokutnika YOLOv3 koristi klastere dimenzija dobivene pomoću algoritma *K-Means Clustering* [9] na trening skupu podataka. Za svaki predviđeni granični pravokutnik, YOLOv3 predviđa četiri vrijednosti: t_x, t_y, t_w, t_h, na slici 2.6. su prikazane kao (*x*, *y*, *w*, *h*). U slučaju kada ćelija ima pomak od gornjeg lijevog kuta slike za (c_x, c_y), tada je predviđeni granični pravokutnik određen širinom i visinom p_w, p_h (slika 2.8.). Koristeći vrijednosti c_x, c_y, p_w, p_h se određuju predikcije (b_x, b_y, b_w, b_h) u odnosu na mjesto primjene filtra, za koje vrijedi:

$$b_x = \sigma(t_x) + c_x \tag{2-2}$$

$$b_y = \sigma(t_y) + c_y \tag{2-3}$$

$$b_w = p_w e^{t_w} \tag{2-4}$$

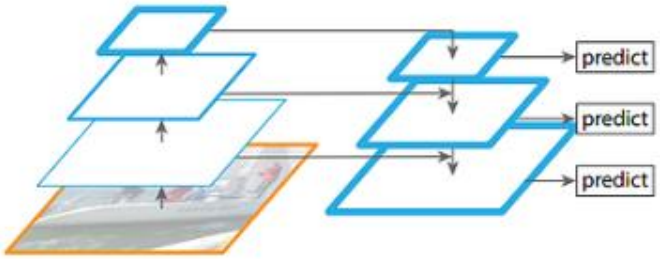
$$b_h = p_h e^{t_h} \tag{2-5}$$



Sl. 2.8. *Prikaz predviđenog graničnog pravokutnika (isprekidana crta) i predikcije (plavo)* [12]

YOLOv3 model za svaki okvir predviđa koje klase može granični pravokutnik sadržavati koristeći klasifikaciju višestrukih oznaka (engl. *multilabel classification*). U prethodnim modelima korišten je *softmax* sloj koji radi na principu da svaki granični pravokutnik sadrži samo jednu klasu, što često može biti netočno: u [10] je kao primjer navedena detekcija čovjeka i žene - jedan granični pravokutnik, ali dvije klase. Iz tog razloga se u YOLOv3 ne koristi *softmax sloj*, već se koriste nezavisni logistički klasifikatori čime se izbjegava problem preklapanja oznaka.

Model za predviđanje graničnih pravokutnika koristi tri različite skale. Na prvoj skali model izdvaja mapu značajki, a nakon prve skale ju proširuje za dva puta. Takvom metodom se dobiva bolje semantičko značenje. Potom se dobivena mapa značajki provlači kroz nekoliko konvolucijskih slojeva te se predviđa matrica graničnih pravokutnika. Sličnim postupkom se radi predikcija graničnih pravokutnika i na zadnjoj skali (slika 2.9.), što između ostalog rezultira preciznijom i točnijom detekcijom malih objekata u odnosu na izvorni YOLO.

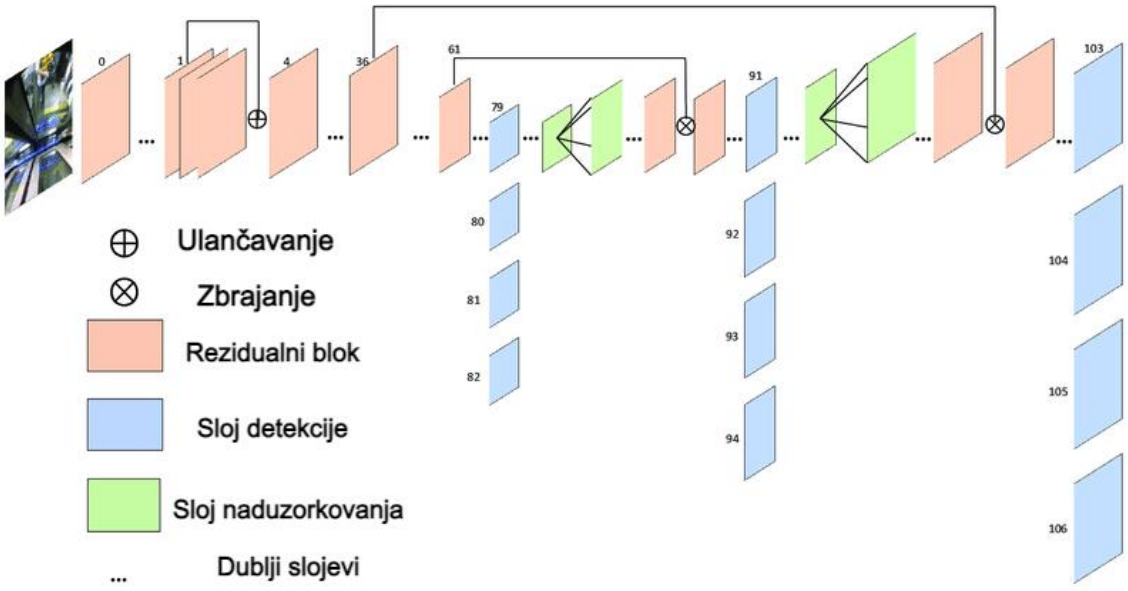


Sl. 2.9. *Piramida značajki na tri različite skale* [11]

YOLOv3 model je nastao kao hibrid mreža YOLOv2, Darknet-19 [12] i rezidualnih slojeva, koji omogućavaju izravan prolaz signala kroz slojeve dubokog modela bez utjecaja nelinearnih funkcija [13]. Mreža se sastoji od 53 konvolucijska sloja, prikazano na slici 2.10., dok je cijela arhitektura YOLOv3 prikazana na slici 2.11.

	Vrsta	Filteri	Veličina	Izlaz
	Konvolucijski	32	3 × 3	256 × 256
	Konvolucijski	64	3 × 3 / 2	128 × 128
1×	Konvolucijski	32	1 × 1	
	Konvolucijski	64	3 × 3	
	Rezidualni			128 × 128
	Konvolucijski	128	3 × 3 / 2	64 × 64
2×	Konvolucijski	64	1 × 1	
	Konvolucijski	128	3 × 3	
	Rezidualni			64 × 64
	Konvolucijski	256	3 × 3 / 2	32 × 32
8×	Konvolucijski	128	1 × 1	
	Konvolucijski	256	3 × 3	
	Rezidualni			32 × 32
	Konvolucijski	512	3 × 3 / 2	16 × 16
8×	Konvolucijski	256	1 × 1	
	Konvolucijski	512	3 × 3	
	Rezidualni			16 × 16
	Konvolucijski	1024	3 × 3 / 2	8 × 8
4×	Konvolucijski	512	1 × 1	
	Konvolucijski	1024	3 × 3	
	Rezidualni			8 × 8
	Sloj sažimanja po prosjeku		Globalno	
	Pot. povezani sloj		1000	
	Softmax			

Sl. 2.10. *Darknet-53 mreža* [8]



Sl. 2.11. *Arhitektura YOLOv3 modela* [8]

2.3.3. YOLOv4

Osim razlike u broju slojeva, YOLOv4 je u odnosu na prethodne verzije (YOLO, YOLOv2, YOLOv3) ostvario bolje rezultate prilikom detekcije malih objekata te je ostvaren

bolji omjer pozitivnih i negativnih uzoraka što u konačnici rezultira ublažavanjem problema neravnoteže između pozitivnih i negativnih uzoraka [14]. U usporedbi s YOLOv3, arhitektura YOLOv4 nije doživjela značajne promjene. Osjetna poboljšanja u preciznosti i točnosti detekcije su ostvarene manipulacijom trening podacima, odnosno augmentacijom postojećih podataka. Skup metoda kojima se to ostvaruje, metode koje samo mijenjaju strategiju treniranja ili samo povećavaju troškove treninga se naziva *„bag of freebies“* (*BoF*). Svrha *BoF*-a je povećanje raznolikosti ulaznih slika, tako da korišteni model ima veću robusnost prema slikama dobivenim iz različitih okruženja. Najčešće vrste augmentacije podataka su fotometrijska i geometrijska izobličenja. Fotometrijsko izobličenje se postiže prilagođavanjem svjetline, kontrasta, nijanse, zasićenosti i šuma slike, dok geometrijsko izobličenje podrazumijeva okretanje, rotiranje i obrezivanje slike.

Osim *BoF*-a postoje i *BoS – „bag of specials“*. *BoS* obuhvaća metode koje služe za poboljšanje određenih atributa u modelu, kao što je povećanje receptivnog polja, jačanje sposobnosti integracije značajki, itd.

Sljedeće poboljšanje koje se uvodi u YOLOv4 je metoda normalizacije serija uzastopnih iteracija (engl. *Cross-Iteration Batch Normalization, CBN*) [15], u kojoj se uzorci iz više nedavnih iteracija zajednički koriste za poboljšanje kvalitete procjene.

Napredak koji je ostvaren u YOLOv4 može se u konačnici svesti na tri stvari [16]:

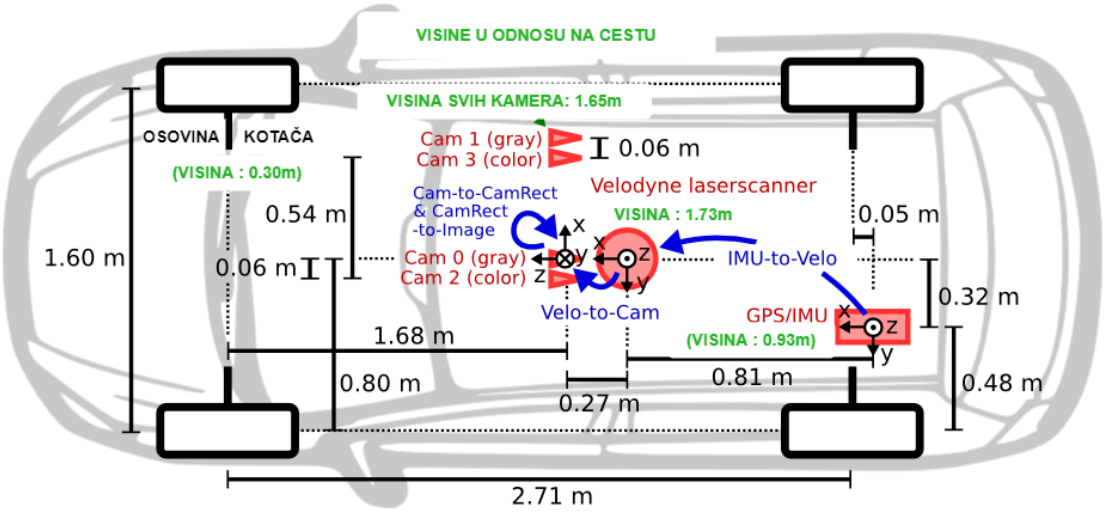
- Omogućeno treniranje brzog i preciznog detektora objekata koji ima mogućnost rada u stvarnom vremenu
- Utjecaj *BoF* i *BoS* metoda tijekom treninga detektora je verificiran
- Učinkovitije i prikladnije korištenje CBN normalizacije podataka prilikom treninga koristeći jednu GPU

2.4. Pregled dostupnih skupova podataka s informacijom o dubini

2.4.1. KITTI podatkovni skup

KITTI podatkovni skup [17] je nastao suradnjom Instituta za tehnologiju Karlsruhe (KIT) i Toyotinog tehnološkog instituta u Chicagu (TTI-C) koji su iskoristili vlastitu platformu za autonomnu vožnju – Annieway, za razvoj baze podataka s prizorima iz svakodnevnog prometa. Cijeli skup je snimljen na cestama unutar i u užoj okolici grada Karlsruhe. Podatci su snimljeni

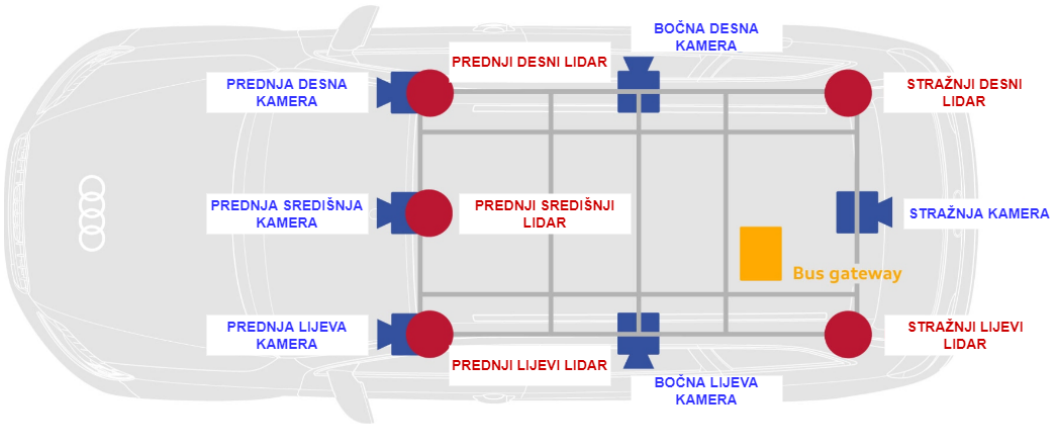
korištenjem standardnog osobnog vozila, Volkswagen Passat 6B, koje je opremljeno s dvije kamere rezolucije 1.4MPx za snimanje u boji i dvije kamere iste rezolucije za snimanje monokromatskih slika. Osim kamera, vozilo je opremljeno LiDAR sustavom dometa do 120m s visoko preciznim GPS sustavom za lokalizaciju. Na slici 2.12. je prikazana ilustracija vozila korištenog za snimanje baze podataka s rasporedom senzora te navedenim visinama i međusobnim razmacima između senzora i ostatka vozila. Razlučivost snimljenih slika iznosi 1242x375 elemenata slike te je ukupan broj slika s oznakama objekata na slici korištenih za izradu ovog rada 8481.



Sl. 2.12. *Raspored senzora na vozilu* [17]

2.4.2. Audi Autonomus Driving Dataset (A2D2) podatkovni skup

Istraživački tim poznatog njemačkog proizvođača automobila [18] je kreirao podatkovni skup s prizorima iz prometa na prostorima nekoliko njemačkih gradova: Gaimersham, Ingolstadt i Munchen. Vozilo kojim su snimljeni podatci je opremljeno skupom senzora koji se sastoji od šest kamera i pet LiDAR senzora. Takvom postavkom kamere i LiDAR senzori imaju pokrivenost 360°, oko cijelog vozila. Korišteni LiDAR senzori imaju domet do 100m i preciznost od +/- 3cm. Korištenim kamerama su snimljene slike rezolucije 1920*1208, brzinom 30 okvira po sekundi. Ukupan broj slika s oznakama objekata na slici korištenih za izradu ovog rada iznosi 10089. Na slici 2.13. je prikazana ilustracija rasporeda senzora na korištenom vozilu.



Sl. 2.13. *Raspored senzora na vozilu* [18]

2.5. Fuzija podataka i postojeća rješenja

Jedan od najvećih izazova autonomnih vozila je paralelno korištenje podataka sa različitih senzora te osiguravanje njihovog skladnog rada. Postupak spajanja podataka s različitih izvora, odnosno senzora, naziva se fuzija podataka. U ovom radu je opisana fuzija podataka sa samo dva senzora, fuzija slika dobivenih sa prednje kamere na vozilu i podatci dobiveni LiDAR sustavom- oblaci točaka (engl. *point clouds*).

Prema [19] postoji nekoliko načina na koje se odvija fuzija, ovisno o tome jesu li slike monokularne ili binokularne (stereo), odnosno jesu li slike dobivene pomoću jedne ili dvije kamere. U slučaju monokularnih slika, fuzija slike i podataka s informacijama o dubini može biti izvršena na tri načina:

- Fuzija na razini signala (engl. *signal-level fusion*) – u radu [20] je predložena mreža automatskog kodiranja temeljena na ResNet mreži koja koristi slike dobivene kamerom povezane s nepotpunim dubinskim mapama (engl. *sparse depth maps*) što rezultira generiranjem guste dubinske mape (engl. *dense depth map*). Budući da je korištena fuzija na razini signala, mreži je potreban stvarni granični pravokutnik na razini elementa slike za svaku sliku. Taj problem je riješen korištenjem samo-nadziranog modela (engl. *self-supervised*) koji je treniran samo nad nepotpunim dubinskim mapama za odgovarajuće slike.
- Fuzija na razini značajki (engl. *feature-level data fusion*) – prema [21] je predložena mreža temeljena na NASNet-u (engl. *NeuralSearchArchitectureNetwork*) za automatsko kodiranje koja omogućava postizanje guste dubinske mape i semantičke

segmentacije koristeći nepotpune dubinske mape i odgovarajuće slike. Slika i odgovarajuća nepotpuna dubinska mapa su poslane paralelnim koderima, obrađuju se pojedinačno te su zatim spojene u zajedničkom dekoderu.

- Višerazinska fuzija (engl. *multi-level data fusion*) – u radu [22] je kombinirana fuzija na razini signala sa fuzijom na razini značajki u mreži treniranoj samo na slikama dobivenim s kamere. Mreža se sastoji od globalne i lokalne grane (engl. *global and local branch*) pri čemu paralelan rad obje grane osigurava rad u stvarnom vremenu.

U slučaju binokularnih slika, K. Park [23] je 2019. objavio rad u kojem opisuje korištenje konvolucijske neuronske mreže u dvije faze s konačnim ciljem trodimenzionalne rekonstrukcije snimljene scene. U prvoj fazi mreža koristi LiDAR podatke i mape dispariteta stereo slika za generiranje fuzijskog dispariteta. Disparitet stereo slika nastaje kada je isti prizor snimljen dvjema kamerama. Ako su isti objekti istovremeno snimljeni dvjema kamerama, na slikama je postojana određena razlika u položaju objekata koja je ovisna o međusobnoj udaljenosti kamera. Ta razlika u položaju objekata na slikama je predstavljena mapom dispariteta. U drugoj fazi se vrši fuzija mapa dispariteta i RGB slika na razini značajki kako bi se predvidjele konačne mape dispariteta u visokoj rezoluciji. Zatim su dobivene mape dispariteta korištene za konačnu trodimenzionalnu rekonstrukciju snimljene scene. Ograničenje navedene metode je bila potreba za velikim podatkovnim skupovima označenih LiDAR podataka. U LidarStereoNet [24] je taj problem riješen nenadziranim učenje. Osim što je na taj način uspješno kompenziran nedostatak označenih LiDAR podataka, također je smanjen i utjecaj neusklađenosti između LiDAR podataka i RGB slika.

U radu [25] je predstavljena metoda spajanja podataka s LiDAR senzora i slika dobivenih kamerom kojom su prvo oba senzora kalibrirana. Nakon kalibracije, kada su određeni intrinzični i ekstrinzični parametri za oba senzora, je izvršena transformacija LiDAR podataka, odnosno oblaka točaka, u koordinantni sustav kamere te je zatim ostvarena korespondencija između 3D i 2D podataka-oblak točaka i slika s kamere. Uz 3D-2D korespondenciju, oblak točaka može dobiti informacije o boji, dok elementi slike mogu sadržavati točne informacije o dubini.

Wang, Zhan i Tomizuka [26] su predstavili metodu kojom je konstruiran novi sloj u dubokoj konvolucijskoj neuronskoj mreži. Konstruirani sloj se naziva rijetki nehomogeni sloj za sažimanje (engl. *sparse non-homogeneous pooling layer*), za transformaciju značajki između ptičje perspektive i pogleda sprijeda. Za preslikavanje (engl. *mapping*) između ptičje perspektive i pogleda sprijeda korišteni su LiDAR oblaci točaka dok je slojem sažimanja omogućeno

spajanje njihovih značajki u bilo kojoj fazi mreže, a u navedenom radu su upotrijebili predloženu metodu za spajanje podataka prije faze prijedloga (engl. *proposal stage*) tako da se prilikom spajanja podataka mogla iskoristiti cijela mapa značajki.

3. IZGRADNJA VLASTITIH DETEKTORA OBJEKATA

U ovom poglavlju su dane informacije o postavljenom okruženju te je predstavljen način izgradnje detektora objekata koji koriste i informacije o dubini dobivene iz LiDAR podataka. Za izgradnju takvih detektora potrebno je prilagoditi ranije spomenute KITTI i A2D2 skupove. Zatim su dani detalji vezani za treniranje YOLOv3 i YOLOv4 detektora objekata.

3.1. Postavljanje radnog okruženja

Radno okruženje je postavljeno na stolnom računalu s instaliranim Ubuntu 20.04.3.LTS operacijskim sustavom. Računalo sadrži Intelov procesor (engl. *central processing unit – CPU*) sedme generacije, radnog takta 3.40GHz i GeForce GTX 1070Ti 8GB NVIDIA grafičku karticu (engl. *Graphic Processing Unit* – GPU).

Za praktični dio ovog rada korištene su implementacije YOLOv3 i YOLOv4 algoritama unutar *darknet* okruženja koje su preuzete sa GitHub stranice [27]. Korišteni i napisani programski kodovi su pisani u C, C++ i u Python programskim jezicima te su korištene biblioteke kao što su OpenCV, NumPy i PIL.

OpenCV (engl.*Open Source Computer Vision*) je biblioteka otvorenog koda s funkcijama za računalni vid i strojno učenje. Prva verzija biblioteke je objavljena 2000. godine od strane Intela. Napisana je u C i C++ programskim jezicima te se može koristiti i u aplikacijama koje rade u stvarnom vremenu. Sastoji se od preko više od 2500 optimiziranih algoritama te podržava korištenje na više operacijskih sustava – *Microsoft Windows*, *Linux*, *MacOS* i *Android* [28]. Instalacija OpenCV bibiloteke se izvršava naredbom unutar terminala:

sudo pip install opencv-python

Numpy je također biblioteka otvorenog koda koja daje podršku za velike, višedimenzionalne nizove i matrice, uz veliku zbirku matematičkih funkcija. Budući da može poslužiti i kao višedimenzionalni spremnik generičkih podataka, korištenjem NumPy-ja je omogućeno definiranje novih tipova podataka. Iz tog razloga se i može integrirati s velikim brojem podatkovnih skupova [29]. Slično kao i OpenCV, instalacija se izvršava naredbom unutar terminala:

sudo pip install numpy

Biblioteka PIL (engl. *Python Imaging Library*) je biblioteka otvorenog koda, prvi put objavljena 2010. godine od strane švedskog znanstvenika Fredrika Lundha. Svrha PIL biblioteke je dodavanje mogućnosti za obradu slika unutar Python interpretera. Kao i OpenCv i Numpy, i PIL podržava rad na više operacijskih sustava [30]. I instalacija se izvršava na sličan način:

sudo pip install pillow

Nakon instalacije potrebnih biblioteka i preuzimanja implementacije YOLOv4 algoritma, potrebna je instalacija *darknet* okruženja. Taj postupak se pokreće naredbom *make* unutar terminala pozicioniranog na istom mjestu gdje se nalazi i preuzeti direktorij.

3.2. Priprema podatkovnih skupova

Za treniranje i testiranje detektora objekata u ovom radu su korišteni KITTI i AD2D skupovi podataka. Izvorne slike su RGB slike koje su korištene za kreiranje slika s podacima o dubini. Za informacije o dubini iz oblaka točaka su iskorišteni LiDAR podaci koji su naknadno vizualizirani, odnosno prikazani kao slika – dubinska mapa (engl. *depth map*). Zatim umjesto fuzije podataka, koja inače nastupa tijekom treniranja, prije treniranja su kreirane nove slike na način da su spojene informacije od RGB slika i informacije od pripadajućih dubinskih mapa dobivenih iz LiDAR podataka tako da je RGB slici dodan četvrti kanal – dubinska mapa. Dobivene slike se u nastavku rada označavaju kao RGBD slike. Budući da su slike PNG formata, četvrti kanal je prikazan kao prozirnost.

3.2.1. KITTI skup podataka

U slučaju KITTI skupa podataka ukupan broj korištenih RGB slika je 8481. Nakon dobivanja dubinskih mapa i kreiranja RGBD slika, ukupno postoji tri skupa podataka od kojih svaki sadrži 8481 sliku: skup RGB slika, skup dubinskih mapa D i skup RGBD slika. Navedeni skupovi su kasnije podijeljeni na skup za učenje i skup za testiranje. Razlučivost RGB i RGBD slika je 1242x375 elemenata slike, dok su dubinske mape dobivene iz LiDAR podataka imale razlučivost 1440x64 elementa slike, koja je za potrebe kreiranja RGBD slika također prilagođena na 1242x375 elemenata slike. Detaljnije informacije o načinu prikupljanja podataka se mogu vidjeti u potpoglavlju 2.4.1.

Za dobivanje dubinskih mapa iz LiDAR podataka korišten je programski kod s GitHub repozitorija [31]. Nakon preuzimanja repozitorija, izvršava se instalacija alata za kreiranje

dubinskih mapa pokretanjem naredbe *make*. Kada je instalacija završena, alat je spreman za uporabu izvršavanjem sljedeće naredbe unutar terminala:

*./lidar-to-depth-image-converter –o putanja_gdje_se_spremaju_generirane_dubinske_mape –d putanja_sa_LiDAR_podatcima –i *.LiDAR_datoteka*

Na slici 3.1. a i 3.1. b su prikazani primjeri RGB slike i njezine pripadajuće dubinske mape dobivene na temelju LiDAR podataka. Nakon dobivene dubinske mape, kreirane su ranije spomenute RGBD slike, primjer slike prikazan na slici 3.1. c. Programski kod korišten za kreiranje RGBD slika može se pronaći u prilozima pod oznakom P.3.1.

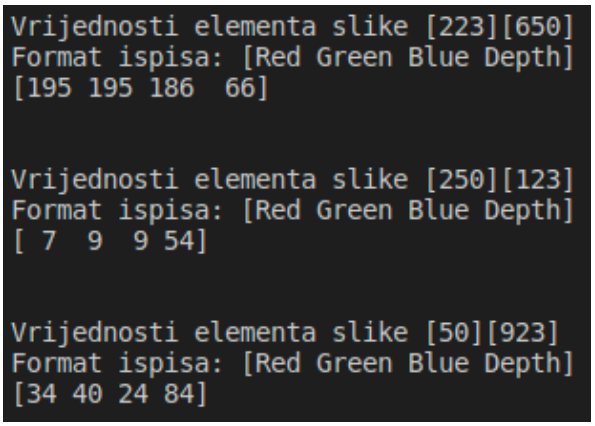
(a)

(b)

(c)

Sl. 3.1. *Primjer RGB slike (a), dubinske mape (b) i RGBD slike (c)*

Na slici 3.2. je prikazan ispis vrijednosti nekoliko proizvoljno odabranih elemenata RGBD slike iz skupa podataka za učenje. Budući da je RGBD slika u pitanju, jedan element slike je određen s četiri vrijednosti: vrijednosti crvene, zelene i plave boje, te vrijednost intenziteta kojim je predstavljena dubina određenog elementa slike.



Sl. 3.2. *Prikaz vrijednosti proizvoljno odabranih elemenata RGBD slike*

Nakon modifikacija nad slikama, izvršene su modifikacije nad oznakama (engl. *labels*) svih slika. Izvorne oznake su bile odgovarajućeg formata, tekstualne datoteke, što je prikladno za *darknet* okruženje. Jedina potrebna prilagodba je bila zamjena imena klasa određenom znamenkom zato što su u *darknet* okruženju imena klasa predstavljena brojem. U skupu podataka je označeno šest različitih klasa objekata: *Car, Van, Truck, Pedestrian, Cyclist* i *Tram,* a u oznakama je svaka klasa predstavljena brojem kako je prikazano u tablici 3.1. Navedena promjena je ostvarena korištenjem koda koji se može pronaći u prilozima pod oznakom P.3.2.

Tab. 3.1. *Imena klasa KITTI skupa podataka predstavljena brojem*

IME KLASE	BROJ KLASE
Car	*0*
Van	*1*
Truck	*2*
Pedestrian	*3*
Cyclist	*4*
Tram	*5*

3.2.2. A2D2 skup podataka

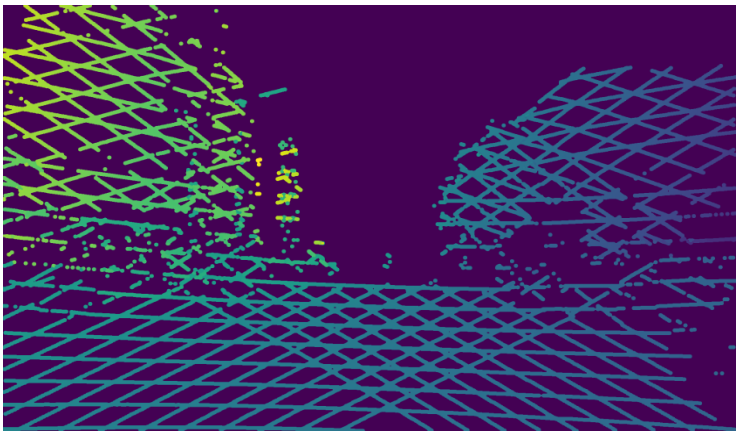
U slučaju A2D2 skupa podataka ukupan broj korištenih RGB slika je 10089. Nakon dobivanja dubinskih mapa i kreiranja RGBD slika, ukupno postoji tri skupa podataka od kojih

svaki sadrži 10089 sliku: skup RGB slika, skup dubinskih mapa D i skup RGBD slika. Navedeni skupovi su kasnije podijeljeni na skup za učenje i skup za testiranje. Razlučivost slika iz sva tri skupa je 1920x1208 elemenata slike. Detaljnije informacije o načinu prikupljanja podataka se mogu vidjeti u potpoglavlju 2.4.2.

Analogno KITTI skupu podataka, i kod A2D2 skupa podataka je također potrebno dobiti dubinske mape iz LiDAR podataka. U ovom slučaju korištena su upustva dostupna uz skup podataka [18], ali uz naknadne preinake, cijeli algoritam se može pronaći u prilozima pod oznakom P.3.3.

Nakon dobivenih dubinskih mapa, na isti način kao i s KITTI skupom podataka su kreirane RGBD slike, prikazane na slici 3.3., zajedno s RGB slikom i dubinskom mapom, dok su na slici 3.4. prikazane vrijednosti proizvoljno odabranih elemenata slike.

(a)



(b)

(c)

Sl. 3.3. *Primjer RGB slike (a), dubinske mape (b) i RGBD slike (c)*

```
Vrijednosti elementa slike [223][650]
Format ispisa: [Red Green Blue Depth]
[166 147 136  84]


Vrijednosti elementa slike [250][123]
Format ispisa: [Red Green Blue Depth]
[133 136 135 140]


Vrijednosti elementa slike [50][923]
Format ispisa: [Red Green Blue Depth]
[227 234 235  84]
```

Sl. 3.4. *Prikaz vrijednosti proizvoljno odabranih elemenata RGBD slike*

Kao i za KITTI skup podataka, bilo je potrebno izvršiti prilagodbu oznaka. Izvorne oznake za A2D2 skup podataka su bile u JSON (engl. *JavaScript Object Notation*) formatu te je nad njima izvršena konverzija iz JSON formata u tekstualni oblik koristeći programski kod u prilozima pod oznakom P.3.4. Zatim su nakon konverzije, oznake u obliku tekstualnih datoteka, prilagođene za rad u *darknet* okruženju. U A2D2 skupu je kao i u KITTI skupu podataka, označeno šest različitih klasa objekata koje su prikazane u tablici 3.2.

Tab. 3.2. *Imena klasa A2D2 skupa podataka predstavljena brojem*

IME KLASE	**BROJ KLASE**
Car_VanSUV	*0*
Truck_UtilityVehicle_Trailer_CaravanTransporter_EmergencyVehicle	*1*
Cyclist_Bicycle_MotorBiker_Motorcycle	*2*
Pedestrian	*3*
Bus	*4*
Animal	*5*

3.3. Proces treniranja detektora objekata

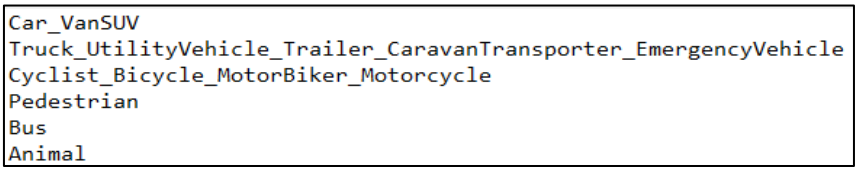
Nakon prilagodbe KITTI i A2D2 skupova podataka, izvršeno je treniranje detektora objekata zasnovanih na YOLOv3 i YOLOv4. KITTI i A2D2 skupovi podataka su podijeljeni na dvije skupine podataka, skup za učenje i skup za testiranje detektora. Budući da se skupovi RGB slika i skupovi dubinskih mapa koriste za kreiranje skupa RGBD slika, sva tri skupa imaju identičan broj slika i u skupu za učenje i u skupu za testiranje. U tablici 3.3. je prikazan broj slika u skupu za učenje i u skupu za testiranje za A2D2 i KITTI skup podataka.

Tab. 3.3. *Broj slika u pojedinim skupovima podataka*

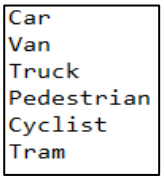
	A2D2	**KITTI**
Skup za učenje	8715	7481
Skup za testiranje	1374	1000

Cijeli proces treniranja je podijeljen na ukupno 12 ciklusa treniranja. Za svaki skup podataka: skup sa RGB slikama, skup sa dubinskim mapama D i skup sa RGBD slikama, je bilo potrebno istrenirati dva tipa detektora. U slučaju KITTI skupa podataka, prvo je na mreži YOLOv3 treniran samo skup sa RGB slikama, zatim skup sa dubinskim mapama D i onda skup sa RGBD slikama. Kada su ta tri ciklusa završena, isti postupak je ponovljen koristeći YOLOv4. Nakon treniranja nad KITTI skupom podataka na YOLOv3 i YOLOv4, isti postupak je izvršen i nad A2D2 skupom podataka.

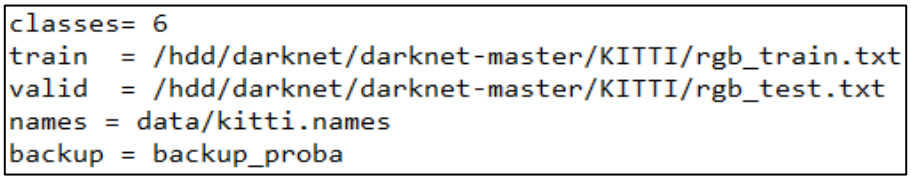
Prije treniranja je bilo potrebno za svaku mrežu i za svaki skup podataka kreirati tri datoteke. Jedna datoteka koja sadrži informacije o broju i imenima klasa, gdje je svaka klasa zapisana u novom redu – *naziv_datoteke.names* (primjeri na slici 3.5. a i slici 3.5. b). Druga datoteka je *naziv_datoteke.data* – datoteka koja sadrži informaciju o broju klasa, putanji do tekstualne datoteke koja sadrži putanje do slika za treniranje, odnosno validaciju mreže, zatim sadrži putanju do datoteke *naziv_datoteke.names* i putanju do direktorija za spremanje datoteke s težinama (primjer na slikama 3.5. c i d). Parametar *valid* koji sadrži putanju do validacijskog skupa podataka nije korišten tijekom procesa treniranja, već je nakon treniranja kao vrijednost tog parametra postavljena putanja do slika iz testnog skupa te je nad tim skupom slika izvršena analiza rezultata u četvrtom poglavlju.



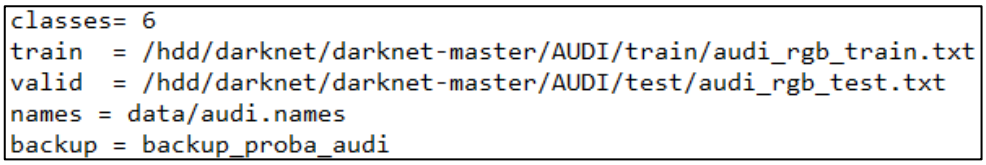
(a)



(b)



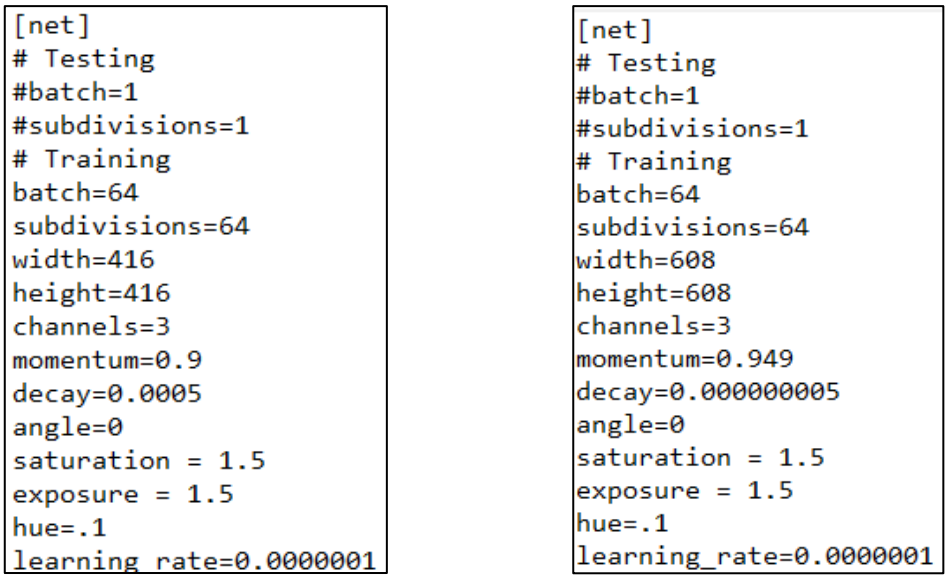
(c)



(d)

Sl. 3.5. *Prikaz datoteka audi.names za A2D2 (a) i kitti.names za KITTI (b) skup podataka, te prikaz datoteke kitti.data (c) za slučaj KITTI RGB skupa podataka i audi.data za A2D2 RGB skup podataka*

Osim navedenih datoteka, još je potrebna konfiguracijska datoteka *naziv_datoteke.cfg* koja sadrži sve potrebne parametre za te treniranje i testiranje mreže strukturu neuronske mreže. Na slici 3.6. (a) prikazan je isječak kofiguracijske datoteke s osnovnim parametrima potrebnim za rad YOLOv3 mreže, s vrijednostima postavljenim za treniranje: širina (engl. *width*) i visna (engl. *height*) ulazne slike, stopa učenja (engl. *learning rate*), moment (engl. *momentum*), i ključan parametar broj kanala slike (engl. *channels*). Ovaj parametar je ključan zato što se on jedini mijenjao prilikom treniranja različitih skupova. U slučaju treniranja s RGB slikama postavljen je na vrijednost 3, u slučaju dubinskih mapa je postavljen na 1, a prilikom treniranja s RGBD slikama, postavljen je na 4. Na slici 3.6 (b) je prikazana ista datoteka za YOLOv4 mrežu. Spomenute tekstualne datoteke se mogu pronaći u prilozima pod oznakom P.3.5 i P.3.6.



(a) (b)

Sl. 3.6. *Prikaz isječaka konfugiracijskih datoteka za YOLOv3 (a) i za YOLOv4 (b)*

Navedene konfiguracijske datoteke sadrže parametre mreže koje je potrebno prilagoditi prije početka treniranja detektora. Prva stvar koju je potrebno promijeniti je vrijednost broja filtera, prije svakog sloja detekcije. Broj filtera se postavlja prema formuli:

$$broj\_filtera = (broj\_klasa + broj\_izlaza)\ x\ broj\_skala \tag{3-1}$$

gdje su:

- *broj_filtera* – broj filtera u konvolucijskom sloju,
- *broj_klasa* – ukupan broj klasa za korišteni skup podataka, definiran u *.data* datoteci,
- *broj_izlaza* – broj veličina u izlaznom vektoru (vjerojatnost predikcije, *x* i *y* koordinate gornjeg lijevog vrha graničnog pravokutnika, *x* i *y* koordinate donjeg desnog vrha graničnog pravokutnika),
- *broj_skala* – postavljeno na 3 jer i YOLOv3 i YOLOv4 rade predikciju na 3 skale

Budući da se razlučivost ulaznih slika definira unutar konfiguracijske datoteke, u slučaju YOLOv3 iznosi 416x416 elemenata slike i 608x608 u slučaju YOLOv4, razlučivost slika prije početka procesa treniranja nije modificirana.

Nakon prilagodbe broja filtera, potrebno je odrediti vrijednosti predviđenih graničnih pravokutnika u svakom sloju detekcije. Unutar *darknet* okruženja se to postiže naredbom:

./darknet detector calc_anchors putanja_do_data_datoteke -num_of_clusters [broj_klastera] -width [širina] -height [visina] -show

Računski i grafički prikaz rezultata dobivenih pokretanjem navedene naredbe su prikazani na slici 3.7. i slici 3.8.

```
(base) bencevic@rtrkos026-lin:/hdd/darknet/darknet-master$ ./darknet detector calc_anchors cfg/kitti.data -num_of_clusters 9 -width 416 -height 416 -show
 CUDA-version: 10010 (11040), GPU count: 1
 OpenCV version: 4.2.0

 num_of_clusters = 9, width = 416, height = 416
 read labels from 7481 images
 loaded          image: 7481     box: 39375
 all loaded.

 calculating k-means++ ...

 iterations = 89


counters_per_class = 28742, 2914, 1094, 4487, 1627, 511

 avg IoU = 70.01 %

Saving anchors to the file: anchors.txt
anchors =  10, 31,  19, 42,  11, 97,  32, 60,  48, 89,  27,183,  73,123,  82,197, 125,237
^C
(base) bencevic@rtrkos026-lin:/hdd/darknet/darknet-master$
```

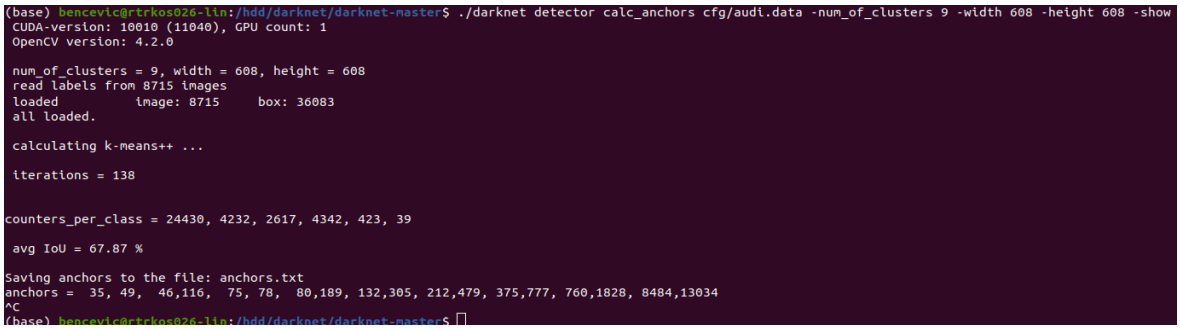
(a)

```
(base) bencevic@rtrkos026-lin:/hdd/darknet/darknet-master$ ./darknet detector calc_anchors cfg/kitti.data -num_of_clusters 9 -width 608 -height 608 -show
 CUDA-version: 10010 (11040), GPU count: 1
 OpenCV version: 4.2.0

 num_of_clusters = 9, width = 608, height = 608
 read labels from 7481 images
 loaded          image: 7481     box: 39375
 all loaded.

 calculating k-means++ ...

 iterations = 71


counters_per_class = 28742, 2914, 1094, 4487, 1627, 511

 avg IoU = 69.96 %

Saving anchors to the file: anchors.txt
anchors =  14, 45,  28, 61,  17,143,  47, 88,  71,130,  40,269, 107,179, 120,287, 182,346
^C
(base) bencevic@rtrkos026-lin:/hdd/darknet/darknet-master$
```

(b)

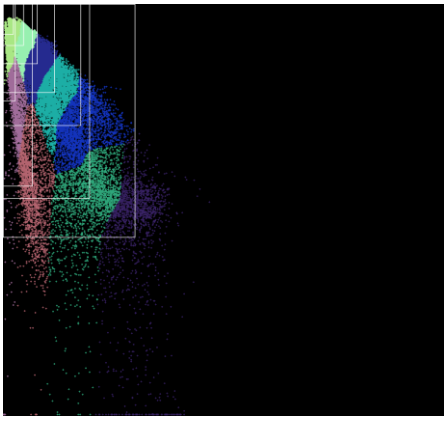
```
(base) bencevic@rtrkos026-lin:/hdd/darknet/darknet-master$ ./darknet detector calc_anchors cfg/audi.data -num_of_clusters 9 -width 416 -height 416 -show
 CUDA-version: 10010 (11040), GPU count: 1
 OpenCV version: 4.2.0

 num_of_clusters = 9, width = 416, height = 416
 read labels from 8715 images
 loaded          image: 8715     box: 36083
 all loaded.

 calculating k-means++ ...

 iterations = 151


counters_per_class = 24430, 4232, 2617, 4342, 423, 39

 avg IoU = 67.87 %

Saving anchors to the file: anchors.txt
anchors =  24, 33,  32, 80,  51, 53,  54,129,  90,209, 145,327, 256,532, 520,1251, 5805,8918
```

(c)

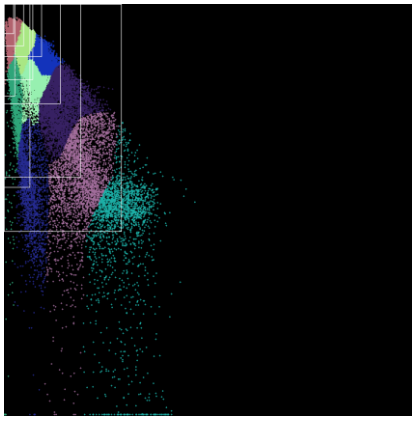


(d)

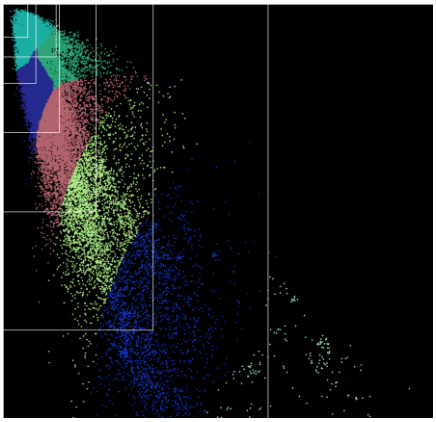
Sl. 3.7. *Računski prikaz klastera predviđenih za KITTI skup podataka za algoritme YOLOv3 (a) i YOLOv4 (b), i za A2D2 skup podataka za algoritme YOLOv3 (c) i YOLOv4 (d)*



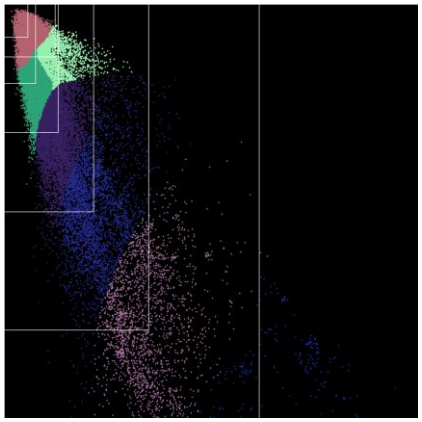
(a)



(b)



(c)



(d)

Sl. 3.8. *Grafički prikaz klastera predviđenih za KITTI skup podataka za algoritme YOLOv3 (a) i YOLOv4 (b), i za A2D2 skup podataka za algoritme YOLOv3 (c) i YOLOV4 (d)*

Na slikama 3.7. i 3.8. su računski i grafički prikaz predviđenih graničnih pravokutnika za KITTI i za A2D2 skupove podataka. To su vrijednosti koje pomažu pri detekciji ako je u pitanju preklapanje klasa objekta. Nakon postavljanja svih konfiguracijskih datoteka, pokrenuto je treniranje na računalu s programskom podrškom i sklopovljem opisanim u potpoglavlju 3.1. Treniranje se može pokrenuti naredbom:

./darknet detector train putanja_do_data_datoteke putanja_do_kofiguracijske_datoteke putanja_do_datoteke_sa_težinama

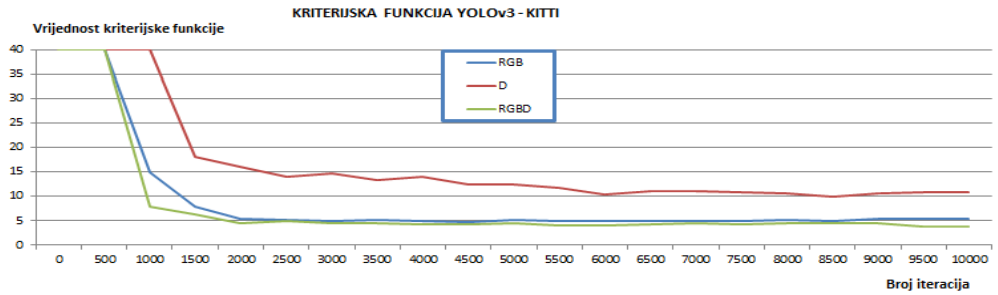
Okvirna vremena, prikazana brojem sati, koja su potrošena na treniranje su navedena u tablici 3.4.

Tab. 3.4. *Okvirna vremena treniranja za svaki skup podataka prikazano u satima*

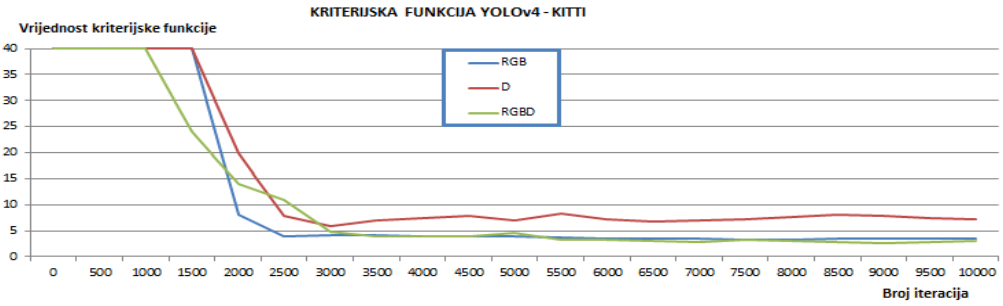
	Vrijeme treniranja [h]			
	KITTI		**A2D2**	
	YOLOv3	**YOLOv4**	**YOLOv3**	**YOLOv4**
RGB	22	23	26	29
D	25	27	28	32
RGBD	26	30	28	31

Svrha treniranja je smanjiti vrijednost kriterijske funkcije i zaustaviti treniranje kada se postigne zadovoljavajuća vrijednost, odnosno kad se ispuni kriterij zaustavljanja. Budući da se tijekom procesa treniranja pokazalo stabiliziranje kriterijske funkcije u nekim slučajevima već nakon manje od 1000 iteracija, za svaki skup podataka kriterij zaustavljanja je postavljen na 10000 iteracija. Jedna iteracija predstavlja treniranje nad jednom serijom slika, unutar konfiguracijske datoteke je serija postavljena na 64 slike. Na slici 3.9. je grafički prikaz kretanja vrijednosti kriterijske funkcije tijekom treniranja, za svaki skup podataka. Na slici 3.9.a je prikazana vrijednost kriterijske funkcije prilikom treniranja nad KITTI skupom podataka s YOLOv3 mrežom, dok slika 3.9.b prikazuje treniranje s YOLOv4 mrežom. Analogno tome, na slikama 3.9.c i 3.9.d je prikazana kriterijska funkcija tijekom treniranja nad A2D2 skupom podataka. U svim slučajevima je treniranje uspješno izvršeno – tijekom iteracija se vrijednost kriterijske funkcije smanjivala dok kriterij zaustavljanja nije bio ispunjen. Na grafovima se može primjetiti kako je vrijednost kriterijske funkcije u slučaju detektora treniranim s RGB i RGBD slikama manjeg iznosa u odnosu na detektore koji su trenirani s dubinskim mapama. To znači da

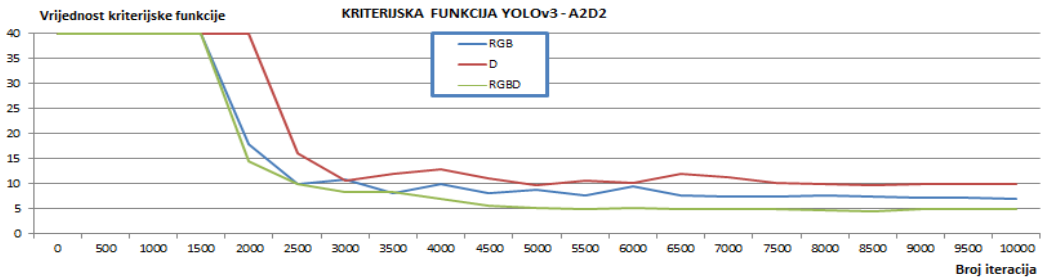
će detektori trenirani s dubinskim mapama ostvariti rezultate s većom pogreškom na skupu za učenje, postići slabije performanse nego detektori koji su trenirani s RGB i RGBD slikama.



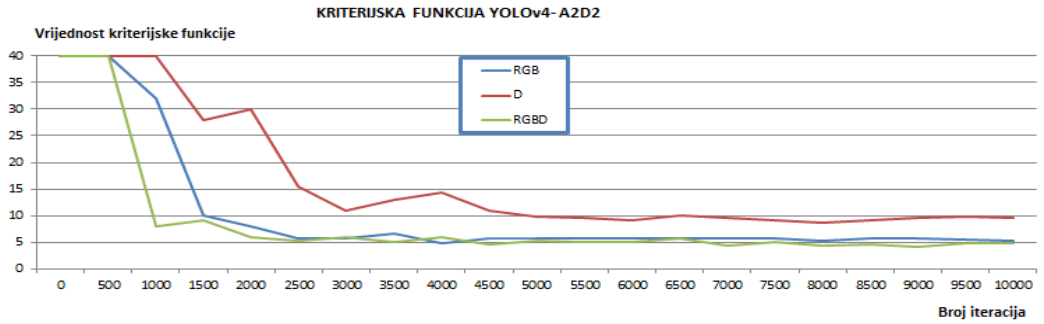
a) kriterijska funkcija YOLOv3 pri treniranju sa KITTI skupom podataka



b) kriterijska funkcija YOLOv4 pri treniranju sa KITTI skupom podataka



c) kriterijska funkcija YOLOv3 pri treniranju sa A2D2 skupom podataka



d) kriterijska funkcija YOLOv4 pri treniranju sa A2D2 skupom podataka

Sl. 3.9. *Grafički prikaz kretanja vrijednosti kriterijske funkcije tijekom treniranja*

4. EVALUACIJA DETEKTORA OBJEKATA

U ovom poglavlju opisan je proces evaluacije detektora koristeći testne skupove podataka. Radno okruženje za proces testiranja je isto kao i prilikom treniranja (vidi potpoglavlje 3.1.).

Za evaluaciju detektora korišteno je pet metrika: preciznost po klasama (engl. *average precision, AP*) i prosječna preciznost po klasama (engl. *mean average precision, mAP*), preciznost (engl. *precision*), odziv (engl. *recall*) i F1-mjera (engl. *F1-score*). Preciznost se računa kao omjer broja točno detektiranih (engl. *true positive, TP*) i ukupnog zbroja točno i netočno detektiranih (engl. *false positive, FP*) objekata (4-1). Točno detektiranim objektom (*TP*) se smatra svaki objekt za kojeg je *IoU* između detektiranog graničnog pravokutnika i stvarnog graničnog pravokutnika barem 50%.

$$preciznost = \frac{TP}{TP+FP} \tag{4-1}$$

gdje je:

- TP – *true positive*, broj točno detektiranih objekata,
- FP – *false positive*, broj pogrešno detektiranih objekata.

Odziv je omjer broja točno detektiranih objekata u odnosu na broj svih objekata koji su trebali biti detektirani:

$$odziv = \frac{TP}{TP+FN} \tag{4-2}$$

gdje je:

- FN – *false negative*, broj objekata koji nisu detektirani.

F1-mjera se određuje na temelju *preciznosti* i *odziva*, a predstavlja njihovu harmonijsku sredinu:

$$F1 = 2 * \frac{preciznost*odziv}{preciznost+odziv} \tag{4-3}$$

AP za svaku klasu se određuje u ovisnosti o *preciznosti* i *odzivu*. Mreža na temelju njih određuje krivulju *preciznost-odziv* (engl. *precision-recall curve*) te površinu ispod te krivulje

(engl. *Area Under the Curve – AUC*) predstavlja kao *AP* za određenu klasu. *mAP* (4-4) se računa kao srednja vrijednost dobivenih *AP* vrijednosti po klasama:

$$mAP = \frac{1}{N}\sum_{i=0}^{N} APi \tag{4-4}$$

gdje je:

- *APi* – prosječna preciznost za *i*-tu klasu
- *N* – broj klasa

Unutar korištenog *darknet* okruženja određivanje *preciznosti* i prosječne preciznosti, *odziva* te *F1* mjere, se postiže pokretanjem sljedeće naredbe u terminalu:

./darknet detector map putanja_do_data_datoteke putanja_do_konfiguracijske datoteke putanja_do_datoteke_sa_težinama

Osim navedenih metrika, tom naredbom se dobije i broj točno detektriranih (*TP*), pogrešno detektiranih (*FP*) objekata za svaku klasu, te ukupan broj objekata koji nisu detektirani (*FN*) uz zadani *IoU* od 50% i uz prag vjerojatnosti postavljen na 25%. Izvršavanjem navedene naredbe nad testnim skupom slika, dobivene su vrijednosti prosječne preciznosti po klasama – *mAP* za svaki od testnih skupova: skup sa RGB slikama, skup sa dubinskim mapama i skup sa RGBD slikama. U nastavku su predstavljeni dobiveni rezultati podijeljeni po skupovima podataka, prvo za KITTI, a zatim za A2D2 skup podataka. Kod oba skupa podataka, za evaluaciju su odabrani modeli dobiveni nakon zadnje iteracije tijekom treniranja.

4.1. Rezultati nad KITTI skupom podataka

mAP vrijednosti za RGB, D i RGBD skupove podataka su prikazane u tablici 4.1., dok su u tablici 4.2. prikazane prosječne preciznosti za svaku pojedinu klasu.

Tab. 4.1. *Prikaz mAP vrijednosti za RGB, D i RGBD skupove podataka*

	mAP	
KITTI	**YOLOv3**	**YOLOv4**
RGB	52,31	60,53
D	25,18	29,68
RGBD	58,06	65,56

Tab. 4.2. *Prikaz prosječne preciznosti po klasama*

	AP_i	
KITTI	**YOLOv3**	**YOLOv4**
RGB	0 = 80,11 1 = 54,41 2 = 63,37 3 = 31,55 4 = 23,49 5 = 60,91	0 = 79,77 1 = 56,84 2 = 68,77 3 = 34,64 4 = 54,40 5 = 68,81
D	0 = 41,66 1 = 28,29 2 = 23,95 3 = 16,41 4 = 12,21 5 = 28,56	0 = 38,99 1 = 27,79 2 = 33,62 3 = 27,08 4 = 16,94 5 = 33,64
RGBD	0 = 88,92 1 = 60,40 2 = 70,34 3 = 35,02 4 = 26,07 5 = 67,61	0 = 86,15 1 = 61,39 2 = 74,27 3 = 59,84 4 = 37,41 5 = 74,31

U tablici 4.3. su prikazane vrijednosti *preciznosti*, *odziva* i *F1* mjere za sve skupove podataka. Za modele testirane na obje mreže, i YOLOv3 i YOLOv4.

Tab. 4.3. *Prikaz preciznosti, odziva i F1 mjere*

KITTI	**YOLOv3**			**YOLOv4**		
	RGB	**D**	**RGBD**	**RGB**	**D**	**RGBD**
PRECIZNOST	0,88	0,42	0,92	0,91	0,44	0,93
ODZIV	0,54	0,26	0,57	0,56	0,27	0,58
F1	0,67	0,32	0,70	0,70	0,34	0,71

U tablici 4.4. su prikazane vrijednosti točno detektiranih (TP), pogrešno detektiranih (FP) i nedetektiranih (FN) objekata. Osim ukupnog broja, vrijednosti TP i FP su prikazane i za svaku klasu posebno, dok je za broj nedetektiranih objekata prikazana samo ukupna vrijednost.

Tab. 4.4. *Prikaz TP, FP i FN vrijednosti*

KITTI	YOLOv3						YOLOv4					
	RGB		D		RGBD		RGB		D		RGBD	
KLASE	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
0	2445	289	1282	1737	2608	209	2521	221	1307	1581	2643	191
1	112	33	67	113	163	21	124	31	64	107	141	23
2	68	8	54	45	97	10	75	5	47	40	89	12
3	62	23	39	84	52	17	68	24	18	84	86	11
4	26	15	26	47	40	18	34	11	11	15	67	4
5	15	2	14	12	30	4	20	6	13	18	68	9
UKUPNO	2728	370	1482	2038	2990	279	2842	298	1460	1845	3094	250
FN	2333		4218		2217		2233		3877		2240	

4.2. Rezultati nad A2D2 skupom podataka

mAP vrijednosti za RGB, D i RGBD skupove podataka su prikazane u tablici 4.5., dok su u tablici 4.6. prikazane prosječne preciznosti za svaku pojedinu klasu. Zatim su u tablici 4.7. su prikazane vrijednosti *preciznosti*, *odziva* i *F1* mjere za sve skupove podataka. Za modele testirane na obje mreže, i YOLOv3 i YOLOv4.

Tab. 4.5. *Prikaz mAP vrijednosti za RGB, D i RGBD skupove podataka*

	mAP	
A2D2	**YOLOv3**	**YOLOv4**
RGB	38,16	41,61
D	18,01	18,10
RGBD	41,71	42,74

Tab. 4.6. *Prikaz prosječne preciznosti po klasama*

	APi	
A2D2	**YOLOv3**	**YOLOv4**
RGB	0 = 57,43 1 = 49,51 2 = 40,93 3 = 24,94 4 = 39,89 5 = 16,24	0 = 61,43 1 = 52,96 2 = 42,66 3 = 35,67 4 = 43,77 5 = 13,16
D	0 = 28,17 1 = 22,24 2 = 19,10 3 = 11,07 4 = 19,29 5 = 8,24	0 = 37,16 1 = 28,53 2 = 14,34 3 = 8,83 4 = 14,43 5 = 5,32
RGBD	0 = 51,50 1 = 46,17 2 = 38,12 3 = 35,16 4 = 44,75 5 = 34,53	0 = 64,15 1 = 57,07 2 = 45,58 3 = 32,27 4 = 41,31 5 = 16,06

Tab. 4.7. *Prikaz preciznosti, odziva i F1 mjere*

A2D2	**YOLOv3**			**YOLOv4**		
	RGB	**D**	**RGBD**	**RGB**	**D**	**RGBD**
PRECIZNOST	0,64	0,29	0,68	0,67	0,29	0,70
ODZIV	0,39	0,18	0,42	0,40	0,21	0,42
F1	0,49	0,23	0,52	0,50	0,24	0,52

U tablici 4.8. su prikazane vrijednosti točno detektiranih (*TP*), pogrešno detektiranih (*FP*) i nedetektiranih (*FN*) objekata. Osim ukupnog broja, vrijednosti točno detektiranih i pogrešno detektiranih su prikazane i za svaku klasu posebno, dok je za broj nedetektiranih objekata prikazana samo ukupna vrijednost.

Tab. 4.8. *Prikaz TP, FP i FN vrijednosti*

A2D2	YOLOv3						YOLOv4					
	RGB		D		RGBD		RGB		D		RGBD	
KLASE	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
0	2261	1250	1323	2564	2376	1072	2387	1141	1328	2492	2901	1093
1	1132	597	361	766	1051	468	1344	553	382	954	1473	630
2	135	113	18	367	164	86	181	104	37	421	213	119
3	97	78	27	186	86	71	76	84	61	296	69	77
4	132	131	56	393	124	117	145	109	58	332	154	101
5	4	7	1	4	3	3	12	7	1	6	10	9
UKUPNO	3761	2176	1786	4280	3804	1817	4145	1998	1867	4501	4820	2029
FN	5882		8136		5706		6108		7023		6656	

4.3. Analiza rezultata

Istrenirani detektori su testirani na testnom skupu slika iz oba skupa podataka – KITTI i A2D2. Dobiveni rezultati sugeriraju da detektori nisu dobri kao detektori iz literature. Uspoređujući samo *mAP* detektora u ovom radu i detektora iz literature, razlika je očita, i do 20% u korist drugih detektora. Međutim, ako se u obzir uzima smjer u kojem se performanse testiranih detektora kreću, prema tablici 4.1. za KITTI skup podataka i prema tablici 4.5. za A2D2 skup podataka, može se uočiti da su detektori koji su tijekom treniranja učili na skupovima sa RGBD slikama ipak ostvarili bolje rezultate od detektora koji su trenirani samo s RGB slikama, ili samo s dubinskim mapama. Prema istim tablicama se vidi i poboljšanje u odnosu na korišteni algoritam. U slučajevima detektora koji su zasnivani na YOLOv4, također su ostvareni bolji rezultati. U tablicama 4.4. i 4.8. su prikazani brojevi detektiranih objekata za svaku klasu, 4.4. za KITTI i 4.8. za A2D2 skup podataka. Prema tim tablicama se može vidjeti da su YOLOv4 detektori za iste skupove podataka ostvarili više točnih detekcija u odnosu na YOLOv3 detektore. U istim tablicama se može vidjeti da su oba detektora točnije detektirali veće objekte, primjerice automobile u odnosu na pješake. Budući da su skupovi podataka prikupljani tijekom vožnje, očita je razlika u broju instanci pojedinih klasa. Na primjer, broj snimljenih automobila u odnosu na broj pješaka je veći, što se može vidjeti po broju TP i FP vrijednosti, te je detektor bolje naučio detektirati automobile. Osim razlike dobivene dodavanjem informacija o dubini, vidljive su razlike i između skupova podataka, detektori koji su trenirani i testirani nad KITTI skupom podataka su ostvarili bolje rezultate.

5. ZAKLJUČAK

Cilj ovog rada je bio poboljšanje performansi detektora dodavanjem informacija o dubini slike. Navedeno je ostvareno koristeći detektore temeljene na YOLOv3 i YOLOv4 algoritmima koji su trenirani i testirani nad KITTI i A2D2 skupovima podataka. Prvo je bilo potrebno kreirati nove skupove podataka, RGB, D i RGBD. Zatim su procesi treniranja i testiranja mogli biti izvršeni. Sama razina detekcije nije visoka, najveći ostvareni *mAP* iznosi tek 65,56% u slučaju RGBD skupa podataka s detektorom s YOLOv4 algoritmom za KITTI skup podataka. Uzrok takvih rezultata bi mogao biti poprilično velik broj nedetektiranih objekata, osobito za D skup u slučaju A2D2 skupa podataka, broj nedetektiranih objekata iznosi 8136. Međutim, u usporedbi rezultata dobivenih s istim detektorom nad RGB i RGBD skupovima ipak postoji poboljšanje performansi. Najlošiji rezultati su ostvareni prilikom treniranja i testiranja nad D skupom slika, ali budući da takve slike sadrže najmanje informacija takav rezultat je bio očekivan. U konačnici, nakon treniranja i testiranja rezultati su pokazali da dodana informacija o dubini zaista poboljšava performanse detektora. Bolji rezultati bi se mogli ostvariti modificiranjem korištenih mreža, korištenjem jače grafičke kartice, ali i korištenjem kvalitetnijih skupova podataka. Kvalitetnijih u smislu da je informacija o dubini već prisutna u izvornoj slici, bez potrebe za dodatnim prilagodbama. Takvim pristupom bi se utjecalo na razinu preciznosti i točnosti detektora.

LITERATURA

[1] „‚Alexa: Grill the steaks!‘ A Cape Cod smart home vacation tale | Fierce Electronics“. https://www.fierceelectronics.com/sensors/alexa-grill-steaks-cape-cod-smart-home-vacation-tale (pristupljeno 02. kolovoz 2022.).

[2] „ADAS Tutorial - PiEmbSysTech“. https://piembsystech.com/adas-tutorial/ (pristupljeno 02. kolovoz 2022.).

[3] R. Roriz, J. Cabral, i T. Gomes, „Automotive LiDAR Technology: A Survey“, *IEEE Trans. Intell. Transp. Syst.*, sv. 23, izd. 7, str. 6282–6297, srp. 2022, doi: 10.1109/TITS.2021.3086804.

[4] A. Süss, V. Rochus, M. Rosmeulen, i X. Rottenberg, „Benchmarking time-of-flight based depth measurement techniques“, predstavljeno na SPIE OPTO, San Francisco, California, United States, ožu. 2016, str. 975118. doi: 10.1117/12.2212478.

[5] J. Redmon, S. Divvala, R. Girshick, i A. Farhadi, „You Only Look Once: Unified, Real-Time Object Detection“. arXiv, 09. svibanj 2016. Pristupljeno: 10. kolovoz 2022. [Na internetu]. Dostupno na: http://arxiv.org/abs/1506.02640

[6] „Introduction to YOLO Algorithm for Object Detection | Engineering Education (EngEd) Program | Section“. https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/ (pristupljeno 10. kolovoz 2022.).

[7] „Intersection over Union (IoU) for object detection“, *PyImageSearch*, 07. studeni 2016. https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/ (pristupljeno 11. kolovoz 2022.).

[8] B. Jelić, „Identifikacija objekata u slici dobivenoj s kamere u vozilu uz korištenje ROS-a“, str. 62.

[9] „K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks | by Imad Dabbura | Towards Data Science“. https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a (pristupljeno 02. rujan 2022.).

[10] J. Redmon i A. Farhadi, „YOLOv3: An Incremental Improvement“, str. 6.

[11] „A Closer Look at YOLOv3 – DMP BLOG“. https://blog.dmprof.com/post/a-closer-look-at-yolov3/ (pristupljeno 02. rujan 2022.).

[12] „ImageNet Classification“. https://pjreddie.com/darknet/imagenet/#darknet19 (pristupljeno 02. rujan 2022.).

[13] „What is Residual Connection?. A technique for training very deep… | by Wanshun Wong | Towards Data Science“. https://towardsdatascience.com/what-is-residual-connection-efb07cab0d55 (pristupljeno 06. rujan 2022.).

[14] P. Jiang, D. Ergu, F. Liu, Y. Cai, i B. Ma, „A Review of Yolo Algorithm Developments“, *Procedia Comput. Sci.*, sv. 199, str. 1066–1073, 2022, doi: 10.1016/j.procs.2022.01.135.

[15] Z. Yao, Y. Cao, S. Zheng, G. Huang, i S. Lin, „Cross-Iteration Batch Normalization“, str. 10.

[16] „A Gentle Introduction to YOLO v4 for Object detection in Ubuntu 20.04“. https://robocademy.com/2020/05/01/a-gentle-introduction-to-yolo-v4-for-object-detection-in-ubuntu-20-04/#What_are_the_different_versions_of_YOLO (pristupljeno 05. rujan 2022.).

[17] „The KITTI Vision Benchmark Suite“. http://www.cvlibs.net/datasets/kitti/ (pristupljeno 09. kolovoz 2022.).

[18] „Driving Dataset“, *a2d2.audi*. https://www.a2d2.audi/a2d2/en.html (pristupljeno 09. kolovoz 2022.).

[19] „A survey of LiDAR and camera fusion enhancement - ScienceDirect“. https://www.sciencedirect.com/science/article/pii/S1877050921005767 (pristupljeno 03. kolovoz 2022.).

[20] F. Ma i S. Karaman, „Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image“, u *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, svi. 2018, str. 4796–4803. doi: 10.1109/ICRA.2018.8460184.

[21] M. Jaritz, R. D. Charette, E. Wirbel, X. Perrotton, i F. Nashashibi, „Sparse and Dense Data with CNNs: Depth Completion and Semantic Segmentation“, u *2018 International Conference on 3D Vision (3DV)*, Verona, ruj. 2018, str. 52–60. doi: 10.1109/3DV.2018.00017.

[22] W. Van Gansbeke, D. Neven, B. De Brabandere, i L. Van Gool, „Sparse and Noisy LiDAR Completion with RGB Guidance and Uncertainty“, u *2019 16th International Conference on Machine Vision Applications (MVA)*, Tokyo, Japan, svi. 2019, str. 1–6. doi: 10.23919/MVA.2019.8757939.

[23] K. Park, S. Kim, i K. Sohn, „High-Precision Depth Estimation Using Uncalibrated LiDAR and Stereo Fusion“, *IEEE Trans. Intell. Transp. Syst.*, sv. 21, izd. 1, str. 321–335, sij. 2020, doi: 10.1109/TITS.2019.2891788.

[24] X. Cheng, Y. Zhong, Y. Dai, P. Ji, i H. Li, „Noise-Aware Unsupervised Deep Lidar-Stereo Fusion“, u *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, lip. 2019, str. 6332–6341. doi: 10.1109/CVPR.2019.00650.

[25] Y. Ding, J. Liu, J. Ye, W. Xiang, H.-C. Wu, i C. Busch, „3D LiDAR and Color Camera Data Fusion“, u *2020 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Paris, France, lis. 2020, str. 1–4. doi: 10.1109/BMSB49480.2020.9379430.

[26] Z. Wang, W. Zhan, i M. Tomizuka, „Fusing Bird View LIDAR Point Cloud and Front View Camera Image for Deep Object Detection“. arXiv, 13. veljača 2018. Pristupljeno: 09. kolovoz 2022. [Na internetu]. Dostupno na: http://arxiv.org/abs/1711.06703

[27] „GitHub - AlexeyAB/darknet: YOLOv4 / Scaled-YOLOv4 / YOLO - Neural Networks for Object Detection (Windows and Linux version of Darknet)“. https://github.com/AlexeyAB/darknet (pristupljeno 17. kolovoz 2022.).

[28] „About“, *OpenCV*. https://opencv.org/about/ (pristupljeno 18. kolovoz 2022.).

[29] „NumPy - About Us“. https://numpy.org/about/ (pristupljeno 18. kolovoz 2022.).

[30] „Pillow“. https://pillow.readthedocs.io/en/stable/index.html (pristupljeno 18. kolovoz 2022.).

[31] A. F. Elaraby, „KITTI LiDAR to Depth Image Converter“. 05. veljača 2022. Pristupljeno: 23. kolovoz 2022. [Na internetu]. Dostupno na: https://github.com/ahmedfawzyelaraby/lidar-to-depth-image-converter

SAŽETAK

U okviru ovog diplomskog rada opisane su neke od postojećih metoda detekcije objekata koje se primjenjuju u autonomnim vozilima. Za potrebe izrade rada korišteni su podatci dobiveni sa kamera koje se nalaze na prednjoj strani vozila i podatci dobiveni LiDAR senzorom, dok su korišteni detektori temeljeni na YOLOv3 i YOLOv4 algoritmima. Pomoću RGB slika i LiDAR podataka koji su naknadno vizualizirani, kreiran je novi skup podataka s RGBD slikama. RGBD slike su nastale na način da je RGB slikama kao četvrti kanal dodana informacija o dubini. Testiranjem detektora nad sva tri skupa podataka zaključeno je da informacije o dubini mogu rezultirati poboljšanjem performansi detektora. Također, rezultati testiranja su pokazali i određene nedostatke detektora, kao najveći nedostatak se ističe relativno niska razina detekcije objekata što ostavlja prostor za potencijalna daljnja poboljšanja.

Ključne riječi: detekcija objekata, YOLOv3, YOLOv4, OpenCV, LiDAR

IMPROVING OBJECT DETECTOR PERFORMANCE BY ADDING IMAGE DEPTH INFORMATION USING A POINT CLOUD

ABSTRACT

As part of this thesis, some of the existing object detection methods used in autonomous vehicles are described. For the purposes of creating the work, data obtained from cameras located on the front of the vehicle and data obtained from a LiDAR sensor were used, while detectors based on YOLOv3 and YOLOv4 algorithms were used. Using RGB images and LiDAR data that were subsequently visualized, a new subset of RGBD image data was created. RGBD images were created by adding depth information to RGB images as fourth channel. By testing the detector on all three subsets of data, it was concluded that depth information can result in improved detector performance. Also, the test results showed certain shortcomings of the detector, the biggest drawback being the relatively low level of object detection, which leaves room for potential further improvements.

Keywords: object detection, YOLOv3, YOLOv4, OpenCV, LiDAR

ŽIVOTOPIS

Ivan Zmeškal rođen je 23. travnja 1996. godine u Požegi. Svoje srednjoškolsko obrazovanje započeo je upisom Tehničke škole u Požegi, smjer tehničar za računalstvo, u vremenskom periodu 2011.-2015. godine. Nakon srednje škole, nastavlja s obrazovanjem upisom preddiplomskog studija smjera računarstvo na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek. Godine 2020. stječe akademski naziv prvostupnik (baccalaureus) inženjer računarstva. Iste godine upisuje diplomski sveučilišni studij računarstva, smjer automobilsko računarstvo i komunikacije.

PRILOZI

Prilog P.3.1. Programski kod za kreiranje RGBD slika

Prilog P.3.2. Programski kod za promjenu naziva KITTI klasa u oznakama

Prilog P.3.3. Programski kod za vizualizaciju A2D2 LiDAR podataka

Prilog P.3.4. Programski kod za prilagodbu A2D2 oznaka

Prilog P.3.5. Tekstualne *names* i *data* datoteke za A2D2 skup podataka

Prilog P.3.6. Tekstualne *names* i *data* datoteke za KITTI skup podataka