UNIVERZITET U BEOGRADU
MATEMATIČKI FAKULTET

Nikola Grulović

TEHNIKE AUGMENTACIJE I KREIRANJA PODATAKA ZA DETEKCIJU OBJEKATA

master rad

Beograd, 2022.

Mentor:

dr Milena Vujošević Janičić, vandredni profesor
Univerzitet u Beogradu, Matematički fakultet

Članovi komisije:

dr Mladen Nikolić, vandredni profesor
Univerzitet u Beogradu, Matematički fakultet

Lester de Abreu Faria, Professor at ITA & PHD in Microelectronics
IPFacens, Sorocaba, São Paulo, Brazil

Datum odbrane: Septembar, 2022.

Naslov master rada: Tehnike augmentacije i kreiranja podataka za detekciju objekata

Rezime: Automatska detekcija i prepoznavanje objekata ima veoma važnu ulogu u svakodnevnom životu. Jedan od najznačajnijih primera korišćenja detekcije i prepoznavanja objekata je kod autonomne vožnje. Da bi se omogućilo efikasno prepoznavanje objekata, koriste se metode mašinskog učenja, na primer neuronske mreže, duboko učenje i metode računarskog vida. Međutim, da bi se dobili rezultati u skladu sa željenim kvalitetom, obično je neophodna velika količina podataka, velika procesorska snaga i obeležavanje podataka koje može da bude veoma skupo ili u velikoj meri nedostupno. Ovaj rad se bavi metodama za proširivanje skupa podataka, kao i metodama za smanjenje vremena potrebnog za obučavanje modela, u slučaju da velika količina podataka ili velika procesorska snaga nisu na raspolaganju. U radu se podaci proširuju standardnim transformacijama za slike i metodama kreiranja novih sintetičkih podataka. Korišćenjem obuke sa prenosom znanja smanjuje se vreme koje je potrebno neuronskim mrežama da nauče glavne karakteristike raspoloživih podataka. Korišćenjem navedenih metoda dobijaju se poboljšani rezultati u odnosu na trenutno najbolji model za detekciju objekata. Poboljšanja modela se ogledaju u boljoj detekciji i u boljem prepoznavanju objekata.

Ključne reči: veštačka inteligencija, mašinsko učenje, računarski vid, duboko učenje, digitalna slika, VGG16, YOLO, StyleGAN

Sadržaj

Glava 1

Uvod

Mašinsko učenje (eng. *machine learning*) se odnosi na sposobnost mašine da donosi zaključke koristeći dostupne podatke umesto strogo definisanih i kodiranih pravila. Zapravo, mašinsko učenje omogućava računarima da sami uče. Rezultati mašinskog učenja posredno se javljaju u svakodnevnom životu, od predviđanja vremenske prognoze do autonomne vožnje.

Tehnike nadgledanog mašinskog učenja podrazumevaju korišćenje velike količine označenih podataka. Međutim, postoji više razloga zbog kojih velika količina označenih podataka nije uvek dostupna. Prikupljanje i obeležavanje podataka je proces koji može biti izuzetno skup ili čak i nemoguć [1]. Na primer, u oblasti medicine, obeležavanje podataka od strane stručnjaka radiologije je skupo i nije izvodljivo u velikim razmerama [2]. U takvim situacijama, mašinsko učenje ima na raspolaganju samo malu količinu podataka i potrebno je primeniti posebne tehnike i pristupe kako bi se problem dostupnosti male količine podataka, ukoliko je to moguće, prevazišao.

U ovom radu se rešava problem dostupnosti male količine podataka, pri čemu su podaci slike sa obeleženim okvirima objekata. Za rešavanje problema male količine podataka koriste se dva pristupa: proširivanje skupa podataka i učenje sa prenosom znanja (eng. *transfer learning*). U radu je opisan postupak proširivanja skupa podataka, pri čemu se podaci proširuju standardnim transformacijama za augmentaciju slika i podacima dobijenim pomoću modela za generisanje sintetičkih slika. Proširivanje skupa podataka navedenim metodama i dalje ne rešava dovoljno dobro problem male količine podataka. U radu je opisana i tehnika učenja sa prenosom znanja [3] na već obučenom modelu za detekciju objekta [4]. Korišćenje učenja sa prenosom znanja za cilj ima da se upotrebi naučeno znanje iz jednog domena kako bi se bolje učilo u drugom [5]. Indirektan doprinos korišćenja učenja sa prenosom

znanja je i smanjivanje procesorkse snage potrebne za obuku modela. Korišćenjem navedenih metoda proširivanja skupa podataka i učenja sa prenosom znanja, dobija se model sa boljim karakteristikama.

Ideje koje su ovde predstavljene zasnivaju se na prethodnim istraživanjima. Kod nekih istraživanja gde su na raspolaganju bile male količine podataka, pomoću standardnih transformacija slika, generisane su nove slike koje su korišćene u obuci modela i koje su pozitivno uticale na krajnje rezultate [6]. Pored toga, predloženo je i generisanje novih slika generativnim suparničkim mrežama za proširenje skupa podataka radi povećanja performansi [6, 7]. Obuka generativnih suparničkih mreža sa malom količinom podataka ne daje zadovoljavajuće rezultate [6], pa se pojavom metode diferencijabilne augmentacije za generativne suparničke mreže [8] stvara mogućnost generisanja novih podataka pomoću malog skupa podataka. Cilj ovog rada je proširivanje skupa podataka standarnim transformacijama slika i generisanje sintetičkih podataka generativnim suparničkim mrežama sa malom količinom podataka, koje ranije nisu bile u fokusu i koje su postale dostupne metodom diferencijabilne augmentacije.

U glavi 2 je dat kratak pregled modela za detekciju objekata koji se koriste u ovom radu. U glavi 3 je dat pregled tehnika i generativne suparničke mreže koje se koriste u rešavanju problema male količine podataka, dok u glavi 4 se opisuju tehnike evaluacije tih modela. U glavi 5 su predstavljeni eksperimenti i rezultati dobijenih modela. U glavi 6 su ukratko prikazani problemi sa kojima se ovaj rad susreće, metode njihovog prevazilaženja i moguća dalja unapređenja rešenja.

Glava 2

Detekcija objekata

Računarski vid (eng. *computer vision*) je oblast mašinskog učenja, koja se bavi izgradnjom i korišćenjem sistema za obradu, analizu i interpretaciju digitalnih slika ili video zapisa [9, 10]. Detekcija objekata je tehnika računarskog vida, koja za zadatak ima da otkrije objekte na slici. Za uspešnu detekciju objekta potrebno je obučiti model na velikoj količini podataka. Kako je nekada nemoguće nabaviti dovoljnu količinu podatka, koristi se augmentacija podataka: podaci se proširuju dodavanjem modifikovanih ili sintetički generisanih podataka [11].

2.1 Razvojni put detekcije objekata

Detekcija objekata se koristi u velikom broju aplikacija, kao što su, na primer, autonomna vožnja, robotski vid, video nadzor, itd [12]. U poslednje dve decenije, napredak u detekciji objekata je prošao kroz dva perioda [12]: „tradicionalni period detekcije objekata" (pre 2014. godine), i „period detekcije objekata zasnovan na dubokom učenju (eng. *deep learning*)" (posle 2014. godine).

Učinak detekcije objekata je značajno poboljšan zahvaljujući dubokoj konvolutivnoj neuronskoj mreži (eng. *convolutional neural networks, CNN*), što je dovelo do izuzetnih otkrića. Razvoj duboke konvolutivne neuronske mreže takođe je omogućio otkrivanje graničnih okvira[1] i značajnih delova objekta na slikama [13, 14, 15, 16].

U eri dubokog učenja, detekcija objekata se može grupisati u dva tipa: *dvostepena detekcija* i *jednostepena detekcija* [12].

[1]Granični okviri predstavljaju koordinate na delu ili celoj slici u okviru kojih se nalazi prepoznati objekat.

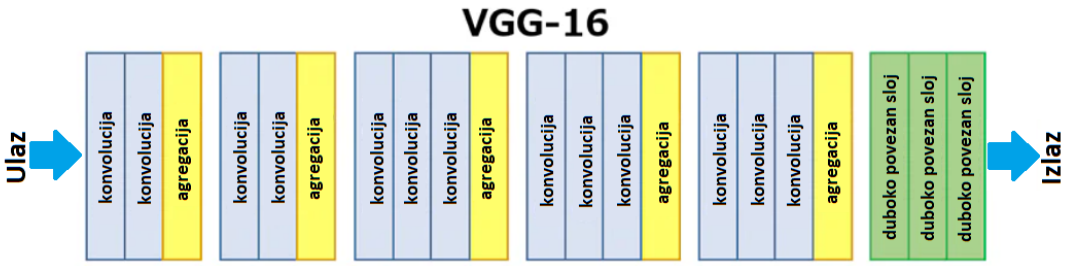
U okviru dvostepene detekcije, prva faza modela se koristi za izdvajanje regiona[2], dok se druga koristi za klasifikaciju i dalje preciziranje objekata. Ovakav metod detekcije objekata je relativno spor, ali daje veoma precizne rezultate.

Modeli jednostepene detekcije objekata odnose se na klasu modela koji preskaču fazu predloga regiona. Ovakvi modeli zahtevaju samo jedan prolaz kroz neuronsku mrežu i predviđaju sve granične okvire u jednom prolazu na kraju. Ovaj tip modela obično ima bržu detekciju, ali manje je precizan pri otkrivanju i prepoznavanju objekata.

2.2 Model za klasifikaciju objekata VGG16

Model VGG16 koristi CNN i smatrao se jednim od najboljih modela računarskog vida 2014. godine nakon pobede na takmičenju ILSVR (Imagenet) [17]. Autori ovog modela su posmatrali druge modele i zapazili da povećanjem dubine i koristeći arhitekturu sa manjim konvolutivnim filterima, može doći do poboljšanja performansi. Promena arhitekture se pokazala kao značajno poboljšanje u odnosu na prethodne modele [17].

Model VGG16 spada u model jednostepene detekcije i dobio je svoj naziv po šesnaest slojeva koji mogu da uče, dok se celokupna mreža sastoji od trinaest konvolutivnih slojeva, pet slojeva agregacije (eng. *max pooling*) i tri duboko povezana sloja koji se sumiraju u ukupno dvadeset i jedan sloj. Arhitektura modela VGG16 je data na slici 2.1.



Slika 2.1: VGG16 arhitektura

U skupu podataka za ILSVR, test skup je sadržao oko četrdeset hiljada slika, što je bilo oko 10% celokupnog skupa podataka [18]. Model je bio u mogućnosti da predvidi objekte na datim slikama sa greškom klasifikacije koja je iznosila 6.8% na skupu za test i time je nadmašio prvi naredni model za 0,9% [17]. Nedostatak

[2]Regioni predstavljaju delove slike koji mogu da sadrže objekte ili delove objekata

modela VGG16 je da nije u mogućnosti da odredi okvire detektovanog objekta na slici.

2.3 Model za detekciju objekata R-CNN

Zbog ponovne upotrebe konvolutivnih neuronskih mreža i veće računarske snage na raspolaganju, 2014. godine pojavljuje se model pod nazivom regioni sa CNN karakteristikama (eng. *regions with CNN features, (R-CNN)*) [13]. Model R-CNN je prvi model dvostepene detekcije i sastoji se od tri modula koji obavljaju različite zadatke [13]:

1. Prvi modul generiše predloge regiona pomoću algoritma selektivne pretrage. Algoritam selektivne pretrage vrši segmentaciju slike na osnovu intenziteta piksela koristeći metod segmentacije zasnovan na grafu [19]. Na osnovu dobijene mape segmentacije stvaraju se predlozi regiona koji se koriste u sledećem modulu [20].

2. Drugi modul je velika konvolutivna neuronska mreža. Svaki predlog se skalira na sliku fiksne veličine i unosi u obučeni CNN model za izdvajanje vektora karakteristika fiksne dužine.

3. Treći modul je skup linearnih metoda potpornih vektora (eng. *support vector machines, (SVM)*) specifičnih za svaku klasu, koji vrše prepoznavanje objekata iz vektora karakteristika.

Problemi koji se javljaju kod R-CNN-a su [21]:

- Zbog moguće klasifikacije velikog broja predloga regiona po slici, predviđanja modela nije moguće koristiti u realnom vremenu.

- Algoritam selektivne pretrage je fiksiran i kao takav se ne prilagođava podacima. To može da ima za posledicu da se u nekim situacijama dobijaju loši predlozi regiona.

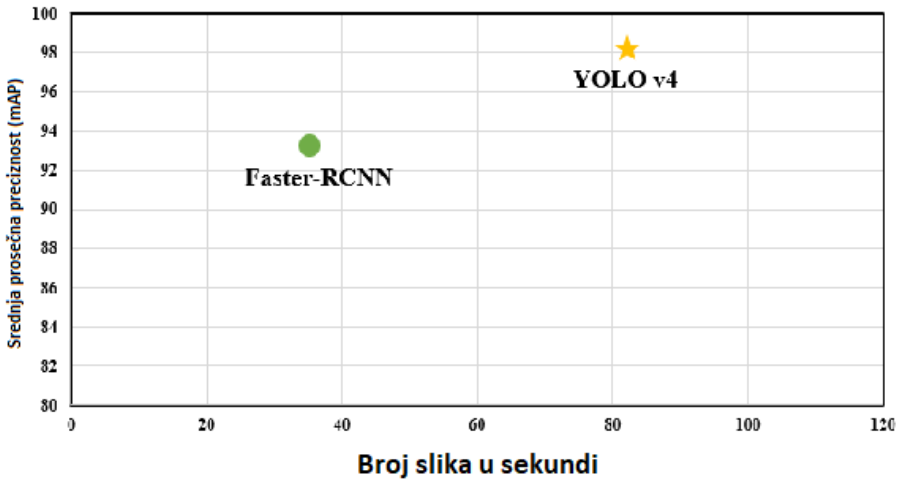
Tokom nekoliko iteracija razvoja R-CNN-a dolazi do znatnog ubrzavanje modela i time nastaju novi modeli brz R-CNN (eng. *fast R-CNN*) [22] i brži R-CNN (eng. *faster R-CNN*) [14]. Model za detekciju objekata brz R-CNN ima sličan pristup kao R-CNN uz nekoliko jednostavnih poboljšanja modela. Model za detekciju objekata brži R-CNN je sastavljena od dva modula. Prvi modul je duboko povezana

konvolutivna mreža koja predlaže regione, dok je drugi modul model R-CNN koji koristi predložene regione za prepoznavanje objekata. Ceo sistem je jedinstvena, ujedinjena mreža za detekciju objekata [14].

2.4 Model za detekciju objekata YOLO

Model za detekciju objekata R-CNN koristi regione da otkrije objekte unutar slike. Model ne gleda na kompletnu sliku, već umesto toga, gleda delove slike koji imaju visoku verovatnoću da sadrže objekte. YOLO (eng. *you only look once*) je model za otkrivanje objekata koji se razlikuje od modela zasnovanih na regionima. Model YOLO spada u model zasnovan na jednostepenoj detekciji i nastao je 2015. godine [23]. Kod modela YOLO jedna konvolutivna mreža predviđa granične okvire i otkriva objekte.

Odvojene komponente detekcije objekata objedinjuju se u jednu neuronsku mrežu. Model koristi pronađene karakteristike iz cele slike da predvidi svaki granični okvir. Takođe, predviđaju se svi granični okviri za sve objekte na slici istovremeno. To znači da model globalno razmišlja o celokupnoj slici i svim objektima na njoj [23]. Model YOLO je duplo brži od modela brži R-CNN, i ima 5% bolju preciznost pri detekciji objekata. Modelu YOLO je potrebno 12 milisekundi da uspešno predvidi objekte na slici, dok modelu brži R-CNN je potrebno 25 milisekundi da predvidi objekte. Poređenje modela je prikazano na slici 2.2 [24].



Slika 2.2: Poređenje modela YOLO sa modelom brži R-CNN [24]. Na y osi je prikazana metrika srednje prosečne preciznosti koju modeli ostvaruju, dok x osa predstavlja broj slika koji modeli mogu da obrade u sekundi.

Glava 3

Tehnike proširivanja skupa podataka

Skup podataka je moguće proširiti kreiranjem novih podataka. U tehnike za kreiranje podataka spadaju augmentacija podataka i generativne suparničke mreže. U nekim slučajevima i nakon proširenja skupa za obuku, količina podataka koja je na raspolaganju može da bude nedovoljna za obuku modela. Takav problem se može ublažiti korišćenjem učenja sa prenosom znanja.

3.1 Problem malog skupa podataka

Modeli obučeni na malom skupu podataka obično uočavaju karakteristike koje su svojstvene tom malom skupu, ali ne i celoj populaciji podataka. To rezultira velikom varijansom i velikom greškom na skupu za test. U ovom radu koriste se dve metode kako bi se izbeglo ovakvo ponašanje. Prvi metod je proširivanje skupa podataka. Podaci se proširuju slikama koje su izmenjene standardnim transformacijama i slikama koje su sintetički generisane pomoću generativnih suparničkih mreža. Drugi metod predstavlja korišćenje učenja sa prenosom znanja.

3.2 Standardne transformacije slika koje se koriste za augmentaciju

U standardne transformacije spadaju rotacija (eng. *rotation*), refleksija (eng. *flip*), skaliranje (eng. *scale*), seckanje (eng. *crop*), zamućenje (eng. *blur*) i augmentacija boja (eng. *color augmentation*). Nasumičnim kombinovanjem navedenih metoda se kreiraju novi podaci koji služe kao proširenje skupa podataka za obučavanje.

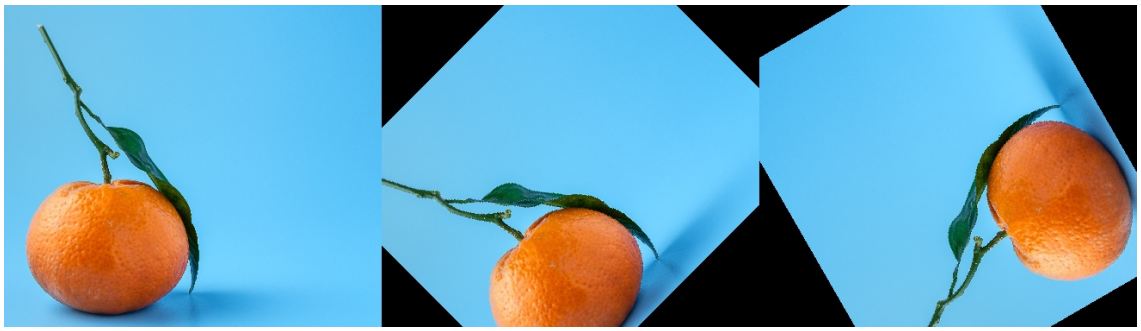
Refleksija

Refleksija slike se odnosi na kreiranje nove slike koja odgovara slici u ogledalu, u odnosu na vertikalnu ili horizontalnu osu. Operaciju refleksije nije moguće primeniti u svakom domenu. Na primer, vertikalna refleksija neće imati mnogo smisla u slučaju saobraćajnih vozila. Primeri refleksije su prikazani na slici 3.1.

Slika 3.1: Primer refleksije: originalna slika (levo), vertikalna refleksija (sredina) i horizontalna refleksija (desno) [25]

Rotacija

Rotacija predstavlja transformaciju koja kreira nove slike koje odgovaraju originalnoj slici okrenutoj za proizvoljan ugao. Primenom operacije rotacije dimenzije slike se često menjaju. Ukoliko je neophodno sačuvati originalne dimenzije, slika se po potrebi seče, dok se prazni delovi slike popunjavaju određenom metodom. Primeri rotacija su dati na slici 3.2.



Slika 3.2: Primer rotacije: originalna slika (levo), rotacija za 45° (sredina) i rotacija za 120° (desno) [25]

Skaliranje

Skaliranje predstavlja transformaciju koja povećava ili smanjuje sliku za zadati faktor koji je veći od nule. Prilikom korišćenja faktora skaliranja većeg od jedan, dimenzije konačne slike su veće od originalne. Da bi se sačuvala originalna dimenzija, nova slika se seče na originalnu dimenziju. U slučaju da je faktor skaliranja između nule i jedinice, dimenzije slike su manje od originalnih. Da bi se sačuvale originalne dimenzije, prazan prostor se popunjava određenom metodom. Primeri skaliranja su dati na slici 3.3.

Slika 3.3: Primer skaliranja: originalna slika (levo), faktor skaliranja veći od 1 (sredina) i faktor skaliranja između 0 i 1 (desno) [25]

Isecanje

Isecanje slike se odnosi na kreiranje nove slike nasumičnim uzorkovanjem dela originalne slike. Isečak se potom skalira na dimenzije originalne slike. Primer isecanja je dat na slici 3.4.

Slika 3.4: Primer isecanja: originalna slika (levo), nasumično isecanje (sredina i desno) [25]

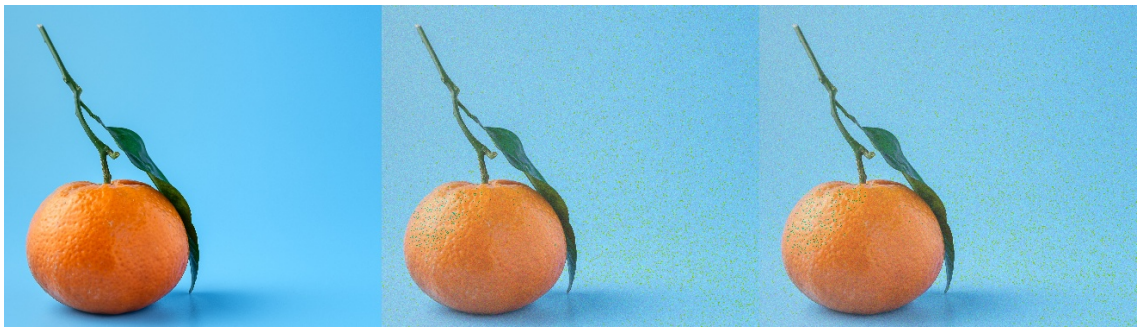
Translacija

Translacija podrazumeva pomeranje slike u horizontalnom ili vertikalnom pravcu. Ovaj metod je veoma koristan, jer se većina objekata može nalaziti na bilo kom mestu na slici. Prazan prostor koji nastaje nakon translacije se boji odgovarajućim metodama. Primer translacije je prikazan na slici 3.5.

Slika 3.5: Primer translacije: originalna slika (levo), nasumična translacija (sredina i desno) [25]

Gausov šum

Gausov šum predstavlja dodavanje nasumičnih vrednosti iz normalne raspodele na piksele originalne slike. Time se dobija nova slika sa šumom. Dodavanjem Gausovog šuma, deformišu se sve karakteristike u podacima. To je bitno jer u skupu podataka za obučavanje mogu postojati podaci koji sadrže slične karakteristike, što može dovesti do preprilagođavanja modela koji se obučava na tom skupu. Primer šuma je dat na slici 3.6.



Slika 3.6: Primer dodavanja šuma: originalna slika (levo), srednja vrednost 1 i standardna devijacija 0.1 (sredina), srednja vrednonst 0 i standardna devijacija 0.5 (desno) [25]

Gausovo zamućenje

Gausovo zamućenje slike predstavlja primenu operacije konvolucije[1] sa Gausovim zvonom [26]. Povećanjem standardne devijacije dobija se jači efekat zamućenosti. Primer zamućenja je prikazan na slici 3.7.

Slika 3.7: Primer zamućenja: originalna slika (levo), standardna devijacija 7 (sredina), standardna devijacija 21 (desno) [25]

Augmentacija boja

Augmentacija boja (eng. *color augmentation*) menja svojstava boje slike izmenom vrednosti njenih piksela. Vrednosti piksela mogu se menjati na različite načine.

Tehnika osvetljenja (eng. *brightness*) podrazumeva množenje svakog piksela slike odgovarajućom faktorom osvetljenja većim od nule. Ako je faktor osvetljenja između nule i jedinice, dobija se tamnija slika, a ako je veći od jedinice dobija se svetlija slika. Primer osvetljenja je prikazan na slici 3.8.

Slika 3.8: Primer menjanja osvetljenja, originalna slika (levo), faktor osvetljenja između 0 i 1 (sredina), faktor osvetljenja veći od 1 (desno) [25]

[1]konvolucija je operacija nad dve funkcije f i g koja proizvodi treću funkciju $f * g$, koja izražava kako se oblik jedne modifikuje od strane druge

Tehnika menjanja kontrasta (eng. *contrast*) predstavlja menjanje stepena razlike između svetlijih i tamnijih delova slike. Povećanjem kontrasta se povećava razlika između svetlijih i tamnijih delova, dok smanjivanjem razlika se umanjuje. Primer promene kontrasta je prikazan na slici 3.9.

Slika 3.9: Primer promene kontrasta, originalna slika (levo), primer pojačanih kontrasta (sredina i desno) [25]

Tehnika menjanja intenziteta (eng. *saturation*) slike predstavlja menjanje jačine boja. Povećavanjem intenziteta se dobija slika sa izraženijim bojama. Primer promene intenziteta je prikazan na slici 3.10.

Slika 3.10: Primer promene intenziteta, originalna slika (levo), slabije povećanje intenziteta (sredina), jače povećanje intenziteta (desno) [25]

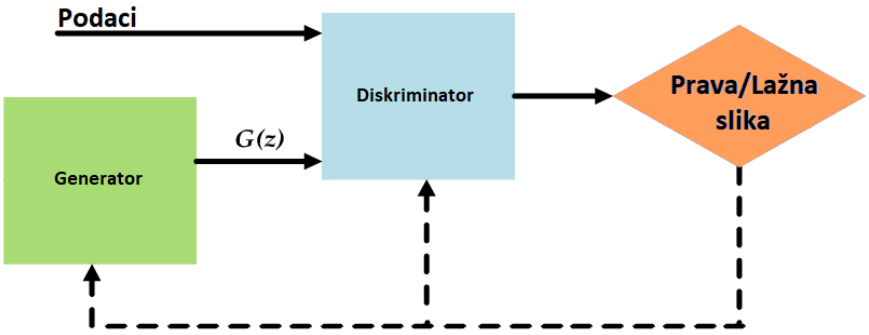
Tehnika menjanja nijanse (eng. *HUE*) pomera piksele na slici u drugu tačku na krugu boja. Nasumično pomeranje generiše nove slike sa drugim bojama. Primer promene nijanse je prikazan na slici 3.11.

3.3 Generativne suparničke mreže

Tokom razvoja mašinskog učenja, razvijen je koncept generativnih suparničkih mreža (eng. *generative adversarial network, GAN*). Kod generativnih suparničkih

Slika 3.11: Primer promene nijanse, originalna slika (levo), pomereni pikseli za nasumične vrednosti (sredina i desno) [25]

mreža postoje dva modela: *generator* i *diskriminator*. Generator za zadatak ima da generiše slike, a diskriminator da razlikuje slike generisane generatorom od pravih slika [27]. Osnovna arhitektura generativnih suparničkih mreža je prikazana na slici 3.12.



Slika 3.12: Generativne suparničke mreže: ulazni podaci predstavljaju slike, G(z) je slika generisa od strane generatora [28]

Generativne suparničke mreže se u velikoj meri oslanjaju na velike količine raznovrsnih i visokokvalitetnih podataka za obučavanje, kako bi generisali fotorealistične slike [8]. Generator se obučava da što bolje generiše slike koje diskriminator neće razlikovati od stvarnih, dok diskriminator se obučava da što bolje razlikuje lažne slike od stvarnih. Kako se generator menja, mora i diskriminator i obratno. Tako se generator i diskriminator obučavaju naizmenično [29]. Funkcije grešaka za generativne suparničke mreže mogu se formulisati preko igre nulte sume [30, 31]. Generator

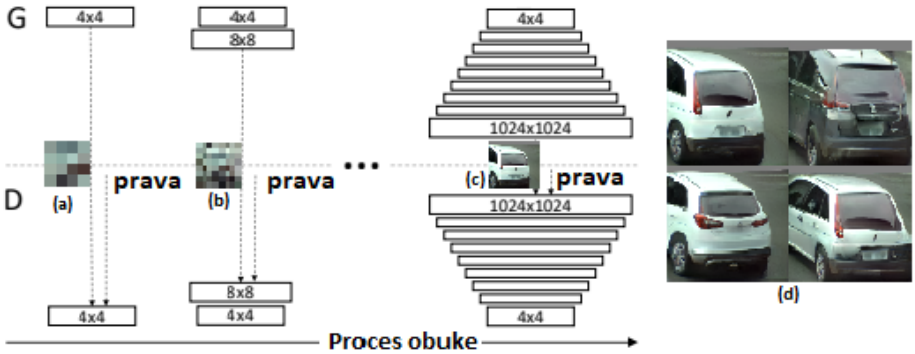
pokušava da minimizuje funkciju dok diskriminator pokušava da maksimizuje. Za dati generator G i diskriminator D problem igre nulte sume se definiše kao [30, 29]:

$$\min_G \max_D \mathbb{E}_x[\log D(x)] + \mathbb{E}_z[1 - \log D(G(z))] \tag{3.1}$$

U datoj formuli:

- $D(x)$ predstavlja procenu diskriminatora da je pravi podatak x stvaran,
- $\mathbb{E}_x$ je očekivanje po raspodeli stvarnih podataka,
- $G(z)$ je izlaz generatora za vrednost z iz latentnog prostora,
- $D(G(z))$ predstavlja procenu diskriminatora da je lažni podatak stvaran,
- $\mathbb{E}_z$ je očekivanje po raspodeli podataka u latentnom prostoru.

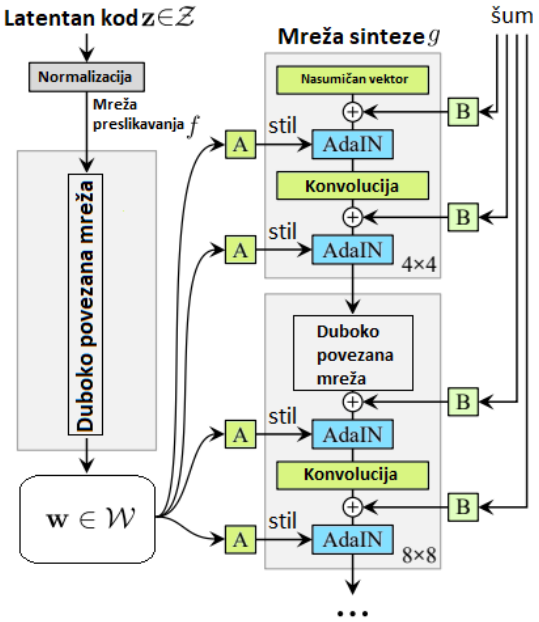
Progresivne generativne suparničke mreže (eng. *progressive generative adversarial network*) modifikuju arhitekturu na taj način da obučavanje generativnih suparničkih mreža počinje sa niskom rezolucijom slike, a zatim progresivno povećava rezoluciju dodavanjem slojeva u mrežu. Proces obuke je prikazan na slici 3.13. Ovakva inkrementalna priroda omogućava da model tokom obuke prvo otkrije vizuelne karakteristike koje se manifestuju na početnim slojevima, od grubih crta pa do sitnih detalja [32].



Slika 3.13: Proces obuke progresivnih generativnih suparničkih mreža počinje od slika niske rezolucije (4×4) i progresivno se dodaju slojevi u mrežu i rezolucija povećava do (1024×1024). Prva slika (slika a) predstavlja prvu iteraciju obuke modela, druga slika (slika b) predstavlja drugu iteraciju obuke, dok treća slika (slika c) predstavlja poslednju iteraciju. Slike automobila (slika d) predstavljaju generisane slike sa najvećom rezolucijom [32].

Tokom razvijanja generativnih suparničkih mreža najviše se radilo na poboljšanju diskriminatora, što je vodilo ka boljim rezultatima. S druge strane, *StyleGAN* (eng. *style generative adversarial network*) je dodatak arhitekturi generativnih suparničkih mreža, koji uvodi značajne modifikacije u model generatora [33].

Generator modela zasnovan na stilu se sastoji iz dva dela i prikazan je na slici 3.14. Prvi deo čini duboko povezanu mrežu sa osam slojeva. Mreža je predstavljena kao funkcija $f : Z \rightarrow W$ i naziva se mreža preslikavanja (eng. *mapping network*). Ulaz mreže predstavlja normalizovani latentni kod $z \in Z$. Latentni kod (eng. *latent code*) ili z-vektor z, je vektor koji sadrži slučajne vrednosti iz Gausove raspodele. Prostor u kome se nalaze svi z-vektori se naziva Z-prostor (eng. *Z-space*) i obeležava se sa Z. Kao izlaz mreže se dobija vektor w koji pripada latentnom prostoru W. Latentni prostor (eng. *latent space*) je apstraktni višedimenzionalni prostor koji sadrži vrednosti karakteristika koje se ne mogu direktno interpretirati. Drugi deo generatora predstavlja mrežu koja uz pomoć vektora sa nasumičnim vrednostima iz Gausove raspodele, slučajnog šuma i vektora w transformisanog afinim transformacijama generiše sliku.



Slika 3.14: Model generatora zasnovanog na stilu, „A" predstavlja afine transformacije dok „B" predstavlja faktor skaliranja za šum [33].

Vektor w transformisan afinim transformacijama služi za prilagođavanje stila sintetički generisanim slikama pomoću adaptivne normalizacije instance (eng. *adaptive instance normalization, AdaIN*). Stil slike predstavlja vektor u latentnom prostoru koji je razdvojen od vektora semantičkog sadržaja slike [34]. Primer prenosa stila sa jedne slike na drugu je dat na slici 3.15.

Slika 3.15: Prenos stilova na željene slike i rezultati operacije AdaIN [34].

Model *StyleGAN* generiše fotorealistične slike. Tokom razvoja, model *StyleGAN* je prošao kroz fazu optimizacije i manje izmene mreže sinteze. Nova verzija *StyleGAN*-a, *StyleGAN2* donosi i do 60% brže obučavanje modela u odnosu na prvu verziju [35]. Zbog ovakvih ubrzanja, direktno se smanjuje procesorska snaga po-

trebna za obučavanje. U radu se koristi ova poboljšana verzija, tačnije arhitektura *StyleGAN2*.

Ukoliko ne postoji na raspolaganju velika količina podataka, rad sa generativnim suparničkim mrežama može da rezultira nekvalitetnim modelima. Da bi se takav problem ublažio, uvedena je metoda diferencijabilne augmentacije (eng. *differentiable augmentation*) za model *StyleGAN2*. Metoda koristi različite tipove augmentacija nad stvarnim i lažnim podacima. Metoda omogućava stabilnije obučavanje i dovodi do bolje konvergencije u odnosu na obučavanje modela bez metode diferencijabilne augmentacije nad istim skupom podataka. Model *StyleGAN2* sa metodom diferencijabilne augmentacija je u stanju da proizvede realistične slike, čak i nad skupom podataka koji sadrži samo sto slika [8].

3.4 Učenje sa prenosom znanja

U nadgledanom učenju, modeli se obučavaju da nauče odnose koji važe između datih ulaza i izlaza. Modeli zatim mogu da vrše predviđanja izlaza nad novim podacima. Modeli će na ovaj način davati dobre rezultate kada rešavaju problem koji se nalazi u istom domenu kao i podaci za obučavanje. Rezultat će se pogoršati ako se domen problema promeni. U najgorem slučaju, može da bude potrebno obučavati novi model čak i kada su domeni problema slični [3].

Učenje sa prenosom znanja predstavlja oblast mašinskog učenja koja se bavi iskorišćavanjem znanja iz postojećeg obučenog modela pri obučavanju novog modela koji treba da rešava sličan zadatak. Tehnika se najčešće koristi kada za obuku novog modela nije na raspolaganju dovoljna količina podataka ili resursa. Korišćenje učenja sa prenosom znanja donosi niz prednosti od kojih su glavne ušteda resursa, smanjivanje greške tokom obuke i generalizacija modela [3].

Glava 4

Tehnike evaluacije modela

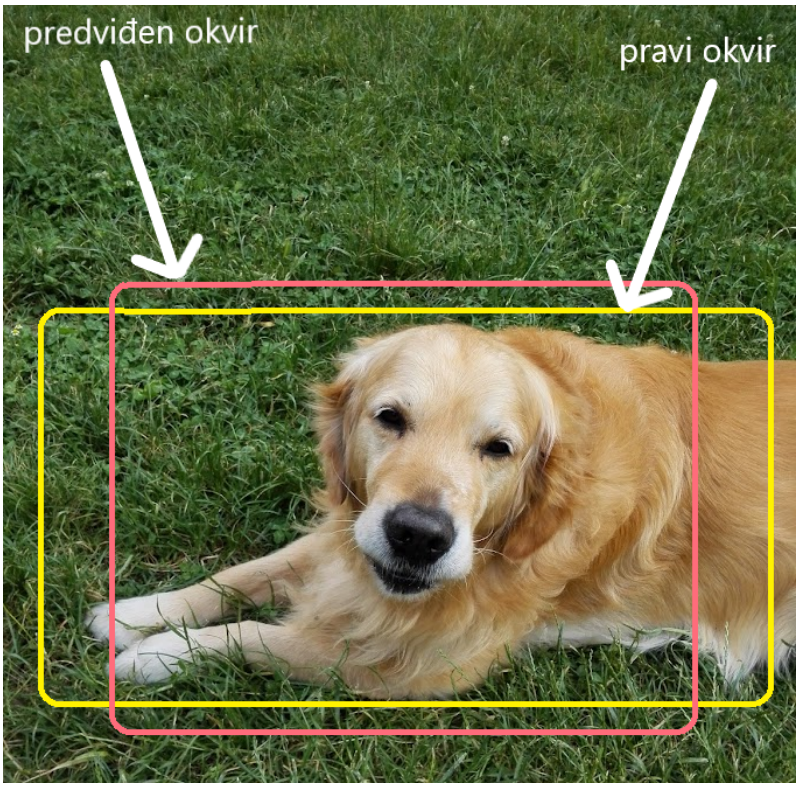
Za evaluaciju modela kod detekcije objekata koristi se mera srednje prosečne preciznosti (eng. *mean average precission, mAP*), koja uzima u obzir preciznost (eng. *precision*) i odziv (eng. *recall*). Takođe, koristi se i matrica konfuzije, kod koje računanje ispravno i pogrešno klasifikovanih objekta zavisi od zadatog praga metrike preseka po uniji (eng. *intersection over union, IoU*) i ocene pouzdanosti (eng. *confidence score*) klasifikovanih objekata. Ocena pouzdanosti predstavlja vrednost verovatnoće objekta koju joj je model dodelio prilikom klasifikacije.

4.1 Presek po uniji

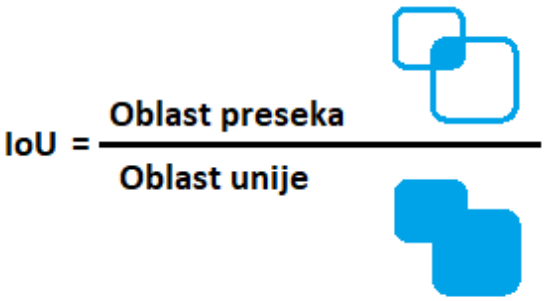
Presek po uniji predstavlja metriku evaluacije koja se koristi za merenje tačnosti detektora objekata. Na slici 4.1 dat je primer okvira, gde je predviđen okvir nacrtan crvenom bojom, dok je pravi okvir nacrtan žutom bojom. Metrika se računa tako što se presek dva okvira podeli njihovom unijom (slika 4.2). Na savršeno predviđene granične okvire metrika IoU će davati vrednost 1.0, dok u slučaju da se granični okviri ne seku vrednost će biti 0. Idealni slučajevi da je metrika 1.0 su retki, i u praksi se smatra dobrim rezultatom kada IoU ima za vrednost 0.8.

4.2 Matrica konfuzije za detekciju objekata

Matrica konfuzije je tabelarni prikaz brojeva ispravno i pogrešno klasifikovanih objekata na osnovu kojih se mogu vršiti ocene modela klasifikacije. Za višeklasnu klasifikaciju detekcije objekata matrica konfuzije se proširuje jednom kolonom i vrstom. Nova kolona služi za zapis broja objekata koji su klasifikovani na slici, a na



Slika 4.1: Predviđen okvir (crveno) i pravi okvir (žuto) nad zadatim objektom



Slika 4.2: Demonstracija računanja IoU metrike

njoj se ne nalaze. Vrsta služi za zapis broja objekata koji se nalaze na slici, ali nisu klasifikovani. Objekat se klasifikuje ispravno kada ispunjava sledeće uslove:

- model je tačno klasifikovao objekat, tj. dodelio mu je pravu klasu,
- vrednost ocene pouzdanosti objekta prelazi zadati prag ocene pouzdanosti (obično se u praksi koristi 0.5) i
- IoU vrednost objekta prelazi zadati IoU prag (obično se u praksi koristi 0.5)

U slučaju da objekat neispunjava jedan od datih uslova, klasifikuje se pogrešno.

Matrica konfuzije se računa i za svaku klasu pojedinačno i slična je binarnoj klasifikaciji. Prilikom računanja matrice konfuzije, posmatra se jedna konkretna klasa i ta klasa se smatra pozitivnom, dok skup koji sadrži sve objekte ostalih klasa se smatra negativnom klasom. Objekat se klasifikuje kao pozitivan ako ispunjava sledeće uslove:

- objekat pripada pozitivnoj klasi model mu je dodelio pozitivnu klasu,
- IoU vrednost objekta prelazi zadati IoU prag

Ako objekat neispunjava jedan od datih uslova, onda mu se dodeljuje negativna klasa. Ocena pouzdanosti ne učestvuje u računanju matrice konfuzije za svaku klasu.

Stvarno pozitivni (eng. *True Positive, TP*) je broj objekata koji pripadaju pozitivnoj klasi, model im je dodelio pozitivnu klasu i IoU vrednost svakog objekta prelazi zadati IoU prag.

Stvarno negativni (eng. *True Negative, TN*) je broj objekata koji ne pripadaju pozitivnoj klasi i model im je dodelio negativnu klasu. IoU se ne upoređuje sa pragom.

Lažno pozitivni (eng. *False Positive, FP*) je broj objekata koji pripadaju negativnoj klasi ali im je model dodelio pozitivnu klasu. Takođe, to je i broj objekata koji pripadaju pozitivnoj klasi ali IoU vrednost svakog objekta ne prelazi zadati IoU prag.

Lažno negativni (eng. *False Negative, FN*) je broj objekata koji pripadaju pozitivnoj klasi a model im je dodelio negativnu klasu. IoU se ne upoređuje sa pragom jer je klasifikacija pogrešna.

4.3 Preciznost, odziv i F_1-mera

Preciznost je procenat stvarno pozitivnih objekata među objektima koji su klasifikovani kao pozitivni i definiše se formulom:

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

Veća preciznost takođe znači da postoji manji broj lažno pozitivnih objekata.

Odziv predstavlja procenat pozitivnih objekata koji su ispravno klasifikovani i definiše se formulom:

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

Visok odziv znači da postoji mali broj lažno negativnih objekata.

Preciznost i odziv se mogu sumirati u meru, koja se zove F_1-mera (eng. F_1-*score*) [36] i definiše se formulom:

$$F_1 = 2\frac{Precision \cdot Recall}{Precision + Recall} \tag{4.3}$$

F_1-mera je harmonična sredina preciznosti i odziva i ona daje relativno niske ocene kada su preciznost ili odziv niski.

Srednja F_1-mera ili (eng. *macro-averaged* F_1-*score*) računa se kao aritmetička sredina:

$$F_{1avg} = \frac{1}{n}\sum_{k=0}^{k=n-1} F_{1k} \tag{4.4}$$

gde je F_{1k} vrednost F_1-mere za k-tu klasu. Srednja F_1-mera se izračunava korišćenjem aritmetičke sredine svih F_1-mera za svaku klasu i koristi se za procenu kvaliteta sa više klasa.

Srednja težinska F_1-mera se računa sledećom formulom:

$$F_{1wavg} = \sum_{k=0}^{k=n-1} (F_{1k}S_k) \tag{4.5}$$

gde su F_{1k} i S_k vrednost F_1-mere za k-tu klasu i proporcijalan broj instanci k-te klase u odnosu na ukupn broj instanci u skupu podataka.

4.4 Srednja prosečna preciznost

Prosečna preciznost (eng. *average precision*) je način računanja površine ispod krive preciznosti i odziva za jednu klasu. Vrednosti preciznosti i odziva se računaju za zadate vrednosti IoU pragova iz intervala [0.5 do 1.0), najčešće sa korakom 0.05. Niz izračunatih vrednosti odziva i preciznost se dopunjuje sa **0** i **1**, respektivno.

Prosečna preciznost se računa sledećom formulom:

$$AP = \sum_{k=0}^{k=n-1} (R_k - R_{k+1})P_k \tag{4.6}$$

gde su R_k i P_k odziv i prezicnost za k-ti IoU prag. Srednja prosečna preciznost se računa kao srednja vrednost svih prosečnih preciznosti koja je data formulom:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{4.7}$$

gde N predstavlja broj klasa, a AP_i prosečnu preciznost za klasu i.

4.5 Fréchet-ova udaljenost

Fréchet-ova udaljenost (eng. *Fréchet inception distance, FID*) je metrika koja se koristi za procenu kvaliteta slika koje su generisane od strane generativnih suparničkih mreža [37]. Fréchet-ova udaljenost upoređuje distribuciju generisanih slika sa distribucijom stvarnih slika koje su korišćene za obuku generatora. Fréchet-ova udaljenost za dve iste slike daje vrednost nula, dok za različite daje broj koji je veći od nule.

Fréchet-ova udaljenost se računa za dve zajedničke normalne raspodele $\mathcal{N}(\mu, \Sigma)$ i $\mathcal{N}(\mu', \Sigma')$ sledećom formulom [38, 37]:

$$d_F(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\mu', \Sigma'))^2 = \|\mu - \mu'\|_2^2 + tr\left(\Sigma + \Sigma' - 2\left(\Sigma^{\frac{1}{2}} \cdot \Sigma' \cdot \Sigma^{\frac{1}{2}}\right)^{\frac{1}{2}}\right) \tag{4.8}$$

U praktičnoj upotrebi, distribucije slika se dobijaju računanjem srednjih vrednosti (μ, μ') i matrica kovarijansi (Σ,Σ') pomoću vektora dobijenih modelom *InceptionV3*[1].

[1]InceptionV3 je konvolutivna neuronska mreža koja se koristi za analizu slika i detekciju objekata.

Glava 5

Implementacija i rezultati

U okviru implementacije, dostupan skup podataka se proširuje metodama augmentacije i sintetičkim podacima koje generiše model *StlyeGAN2*. Da bi se ocenilo da li primenjene tehnike omogućavaju poboljšanje, vrši se i obučavanje modela na osnovnom skupu podataka i njegovo poređenje sa modelima koji su obučeni na proširenom skupu podataka. Kod koji je razvijen u okviru rada na tezi je dostupan na adresi *github*-a[1] [39]. Podaci koji su korišćeni u ovom radu, kao i generisani podaci, dostupni su na ovoj adresi[2] [40]. Za obuku modela korišćena je platforma *Google Colab*[3]. Platforma je obezbeđivala grafičku kartu NVIDIA Tesla P100-PCIE sa 16GB RAM memorije.

5.1 Dostupni podaci

Institut BRAIN (eng. *Brazilian Atrifical Inteligence Nucleus*), u okviru fakulteta FACENS[4] (pt. *Centro Universitário FACENS*, eng. *Faculty of Engineering of Sorocaba*) Sao Paulo, Brazil je obezbedio označeni skup podataka, pod nazivom BVS (eng. *Brazilian vehicle set, BVS*), koji je korišćen u ovom radu. Skup podataka BVS sadrži četiri klase: *car*, *motorbike*, *truck* i *bus*. Primeri iz ovih klasa su prikazani na slici 5.1. Ukupna količina podataka iznosi 1872 slike, od kojih većina pripada klasama *car* i *motorbike*. U skupu podataka BVS, rezolucija svih slika iznosi 800×600px.

[1] https://github.com/Grula/vehicle-detection
[2] https://drive.google.com/file/d/1EkrO28-iBCVWyPqy5mJXn8P8EoKQ6xUH/view?usp=sharing
[3] https://research.google.com/colaboratory
[4] (www.facens.br)

(a) *car* (b) *motorbike*

(c) *truck* (d) *bus*

Slika 5.1: Primeri slika iz dostupnih klasa

Slike se dele u dva skupa, skup za obuku i skup za test modela. Nasumično je uzeto 10% slika iz svake klase za test skup. Test skup nije učestvovao ni u jednoj tehnici augmentacije i nije korišćen ni na koji način pri štimovanju modela. Ukupan broj slika svake klase, broj slika koji pripadaju skupu za obuku i broj slika koji pripada test skupu je prikazan u tabeli 5.1. Iz tabele 5.1 se vidi da je neizbalansiranost između klasa izuzetno visoka, čak 1:15 između klasa *bus* i *car*.

Tabela 5.1: Ukupan broj slika za svaku klasu, i broj nakon podele u skup za obuku i u skup za test

klasa	ukupan broj	skup za obuku	test skup
car	889	800	89
motorbike	966	869	97
truck	10	9	1
bus	7	6	1

5.2 Kreiranje sintetičkih podataka

Skup za obučavanje je proširivan na dva načina: slike su generisane standardnim transformacijama za augmentaciju i modelom *StyleGAN2*. U tabeli 5.2 je dat prikaz tačnog broja slika koji je generisan da bi se dobile izbalansirane klase, tačnije da svaka klasa sadrži podjednaki broj slika u skupu za obuku. Broj slika koji je izabran za proširivanje skupa za obuku je izabran na osnovu klase koja je sadržala najveću količinu podataka, tačnije klase *motorbike*. Klasa *motorbike* je proširena sa 869 slika, tačnije sa onoliko slika koliko je prisutno u skupu za obuku. Nakon proširenja skupa za obuku, svaka klasa sadrži po 1738 slika.

Tabela 5.2: Ukupan broj generisanih slika

klasa	*car*	*motorbike*	*truck*	*bus*
broj generisanih slika	938	869	1732	1729

Generisanje podataka standardnim transformacijama slika

Zbog potrebe proširivanja skupa podataka BVS, generišu se novi podaci tehnikama augmentacije (poglavlje 3.2). Prošireni skup se dalje u radu naziva BVSstandard. Skup augmentacija koje se primenjuju su:

- refleksija — vrši se samo u horizontalnom smeru i vodi se računa o promeni koordinata graničnih okvira,
- rotacija — vrši sa nasumičnim uglom između jednog i pet stepeni i prazan prostor se dopunjava bojama graničnih piksela,
- translacija — pomera sliku u nasumičnom smeru za deset piksela i prazan prostor se dopunjava bojama graničnih piksela,
- Gausov šum — pikseli sa slike se preslikavaju u interval -1 do 1 i na njih se dodaje šum sa konstantnom standardnom devijacijom 0.1 i srednjom vrednošću 0. Nakon dodavanja šuma, pikseli se vraćaju u interval 0 do 255.
- Gausovo zamućenje — primenjuje se na sliku sa konstantom standardnom devijacijom koja iznosi 7. Vrednost standardne devijacije je izabrana zbog rezolucije slika u skupu BVS. Korišćenjem vrednosti 7 za standardnu devijaciju, slike se zamućuju dok je i dalje moguće prepoznati objekat na slici.
- augmentacija boja — Parametar osvetljenja se nasumično bira iz uniformne raspodele iz intervala 0.5 do 1.5. Parametri kontrasta, intenziteta i nijanse se nasumično biraju iz uniformne raspodele iz intervala 0.0 do 10.0. Intervali augmentacija boja su izabrani nakon testiranja intervala različitih opsega. Zaključeno je da se najbolji rezultati dobijaju korišćenjem navedenih intervala.

Iz opisanog skupa augmentacija primenjuju se četiri nasumično izabrane augmentacije. Nasumično se bira jedna afina transformacija iz skupa koji čine refleksija, rotacija i translacija. Takođe, bira se jedna augmentacija između Gausovog zamućenja i Gausovog šuma. Iz skupa augmentacija boja koje čine osvetljenje, kontrast, intenzitet i nijansa, nasumično se biraju dve augmentacije. Primeri generisanih slika su prikazani na slici 5.2.

Slika 5.2: Primeri slika generisanih standardnim transformacijama

Generisanje podataka modelom *StyleGAN2*

Skup podataka BVS se proširuje i sintetičkim podacima generisanim od strane modela *StyleGAN2*. Za ovako proširen skup u radu se koristi naziv BVSstyle.

Međutim, problem male količine podataka se takođe javlja za generisanje novih slika modelom *StyleGAN2*. Korišćenjem metode diferencijabilne augmentacije koja je navedena u poglavlju 3.3 prevazilazi se problem male količine podataka i model uspešno generiše slike. Implementirani metod diferencijabilne augmentacije je testiran nad skupom podataka CIFAR-10 [41].

Arhitektura modela *StyleGAN2* je implementirana pomoću biblioteke *Tensorflow* 2.0[5]. Generisane slike su dimenzija 512×512. Odabrana je ta dimenzija zbog ograničenja memorije grafičke karte. Obuka modela je trajala dvanaest do četrdeset osam sati u zavisnosti od same klase. Na svakih šest sati obuke ili, tačnije, na dvadeset pet hiljada epoha, pravljena je tačka provere (eng. *checkpoint*). Nakon svake tačke provere generisane su slike za računanje Fréchet-ove udaljenosti.

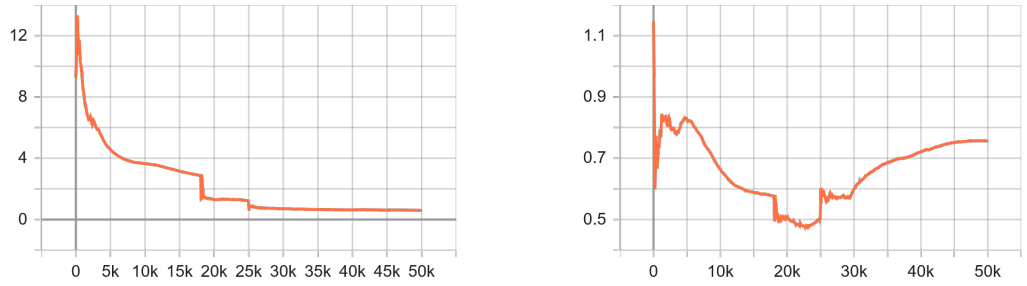
Za evaluaciju modela *StyleGAN2* koristila se metrika Fréchet-ove udaljenosti i funkcije grešaka za generativne suparničke mreže. Nakon svake tačke provere, pomoću Fréchet-ove udaljenosti procenjivana je sličnost između dva skupa slika: generisane slike pomoću modela poređene su sa pravim slikama iz test skupa BVS. Računanje Fréchet-ova udaljenosti odvijalo se na grafičkoj karti NVIDIA 780Ti i trajala je ukupno deset minuta. Kada se na generisanoj slici mogao prepoznati objekat i kada se funkcija greške diskriminatora i generatora nije znatno smanjivala, dalja obuka modela je prekidana. Tačan broj epoha svake klase je dat dalje u tekstu.

Klase *Bus* i *Truck*

Obuka modela *StyleGAN2*-a nad klasama *truck* i *bus* predstavljala je veliki izazov zbog male količine dostupnih podataka. Obuka modela je trajala dvadeset četiri sata. Za obuku modela je generisano dvesta hiljada slika pomoću metode diferencijabilne augmentacije. Originalan skup slika za obuku klase *bus* je sadržao šest slika, a skup za obuku klase *truck* je sadržao devet slika.

[5]Biblioteka *Tensorflow* 2.0 je otvorenog koda i pruža sveobuhvatan ekosistem alata. Objavljena je 2015. godine, a objavio ju je tim *Google Brain*. Osnovna namena je pravljenje modela mašinskog učenja. Biblioteka je namenjena za korišćenje u programskom jeziku *Python*.

Za klasu *truck* funkcija greške je data na slici 5.3 i rezultat Fréchet-ove udaljenosti za dve tačke provere je dat u tabeli 5.3. Iz funkcije greške za diskriminator može se zapaziti da daljom obukom je moglo doći do poboljšanja diskriminatora. Povećanje funkcije greške diskriminatora znači da je generator počeo da generiše slike koje je diskriminatoru bilo teže da razlikuje od stvarnih. Sa slika generisanih poslednjim težinama modela, moguće je uočiti glavne karakteristike objekata. Kako se objekti na slici mogu prepoznati i zbog uštede resursa obuka se prekinula. Primeri generisanih slika klase *truck* su prikazani na slici 5.4.



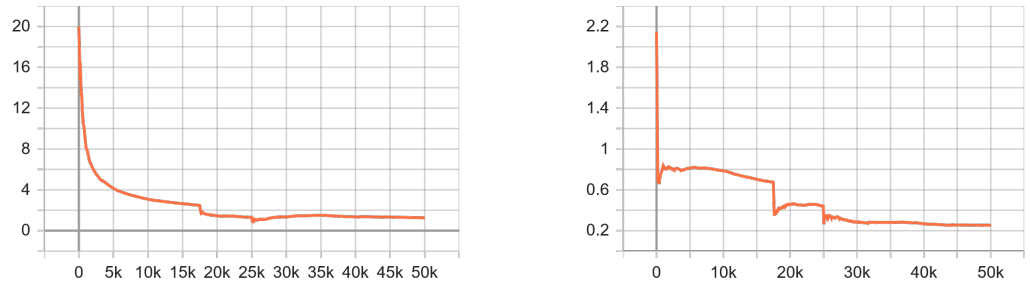
Slika 5.3: Funkcije grešaka za klasu *truck*, generator (levo) i diskriminator (desno). U oba slučaja, x osa predstavlja epohe, dok y osa predstavlja vrednosti odgovarajuće funkcije greške.

Tabela 5.3: Klasa *truck*

broj epohe	Fréchet-ova udaljenost
25 hiljada	432.8
50 hiljada	188.5

Slika 5.4: Primeri generisanih slika klase *truck*

Za klasu *bus* iz funkcije greške (slika 5.5) i rezultata Fréchet-ove udaljenosti (tabela 5.4) može se zaključiti da se daljom obukom za narednih dvadeset pet hiljada epoha ne dobijaju znatno bolji rezultati. Primeri generisanih slika klase *bus* su prikazani na slici 5.6.



Slika 5.5: Funkcije grešaka za klasu *bus*, generator (levo) i diskriminator (desno). U oba slučaja, x osa predstavlja epohe, dok y osa predstavlja vrednosti odgovarajuće funkcije greške.

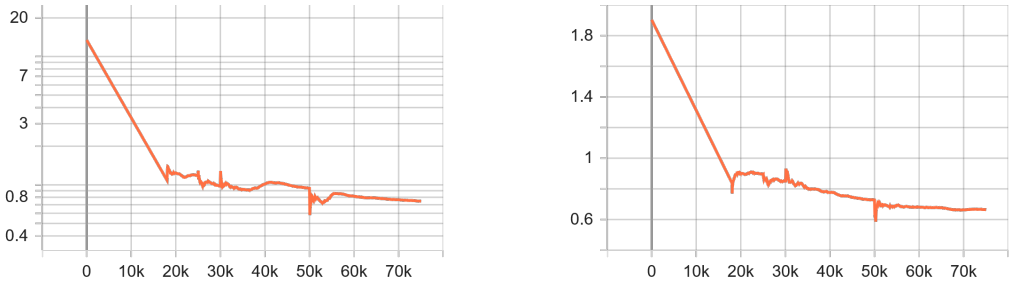
Tabela 5.4: Klasa *bus*

broj epohe	Fréchet-ova udaljenost
25 hiljada	369.2
50 hiljada	220.5

Slika 5.6: Primeri generisanih slika klase *bus*

Klase *car* i *motorbike*

Obuka modela nad klasom *car* je trajala oko trideset šest sati i metodom diferencijabilne augmentacije je generisano trista hiljada slika. Za klasu *car*, funkcija greške je prikazana na slici 5.7. Rezultat Fréchet-ove udaljenosti je prikazan u tabeli 5.5. Daljom obukom klase *car* ne dobijaju se značajna poboljšanja modela. Primeri generisanih slika klase *car* su prikazani na slici 5.8.



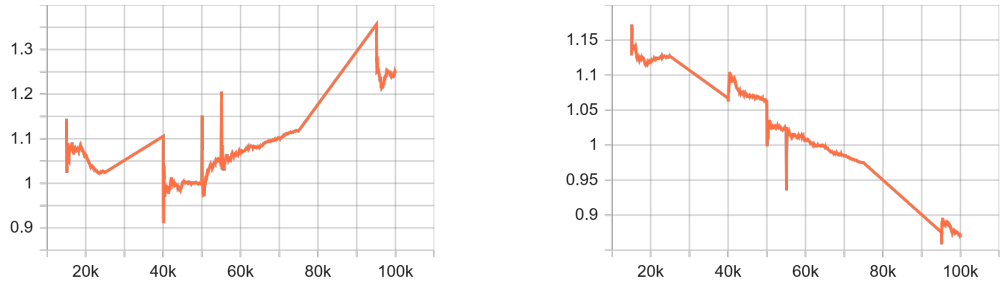
Slika 5.7: Funkcije grešaka za klasu *car*, generator (levo) i diskriminator (desno). U oba slučaja, x osa predstavlja epohe, dok y osa predstavlja vrednosti odgovarajuće funkcije greške.

Tabela 5.5: Klasa *car*

broj epohe	Fréchet-ova udaljenost
25 hiljada	107.6
75 hiljada	87.5

Slika 5.8: Primeri generisanih slika klase *car*

Obuka za klasu *motorbike* je trajala četrdeset osam sati i metodom diferencijabilne augmentacije je generisano četristo hiljada slika. Klasa *motorbike* ima malo lošije rezultate funkcije greške koja je prikazana na slici 5.9. Funkcija greške generatora raste što ukazuje da diskriminator postaje dosta bolji u prepoznavanju stvarnih od lažnih slika. Slike generisane generatorom nisu savršene, ali moguće je prepoznati glavne karakteristike objekata koji su od značaja za obuku modela detekcije objekata. Rezultat Fréchet-ove udaljenosti (tabela 5.6) ukazuje da tokom obuke dolazi do poboljšanja generisanih slika. Primeri generisanih slika klase *motorbike* su prikazani na slici 5.10.



Slika 5.9: Funkcije grešaka za klasu *motorbike*, generator (levo) i diskriminator (desno). U oba slučaja, x osa predstavlja epohe, dok y osa predstavlja vrednosti odgovarajuće funkcije greške.

Tabela 5.6: Klasa *motorbike*

broj epohe	Fréchet-ova udaljenost
25 hiljada	386.8
100 hiljada	151.1

Generisane slike nisu savršene, ali je moguće prepoznati glavne karakteristike objekata. Neke generisane slike su sklone da sadrže artifakte i zamućenosti.

Slika 5.10: Primeri generisanih slika klase *motoribke*

5.3 Model za detekciju i predviđanje objekata VGG16

Osnovni model VGG16 je ograničen na klasifikaciju slika bez predviđanja graničnih okvira i u okviru ovog rada osnovni model je proširen odgovarajućim mrežama da se omogući predviđanje graničnih okvira. Osnovni model je proširen na sledeći način:

- Na poslednji sloj dodato je račvanje na dve nove duboko povezane mreže dubine četiri.
- Na izlazu prve mreže dodata je aktivaciona funkcija *softmax* [29] sa četiri neurona, koja se tumači kao dodeljivanje verovatnoće da objekat pripada jednoj od četiri klase.
- Na izlazu druge mreže dodata je sigmoidna aktivaciona funkcija sa četiri neurona, koja se tumači kao dodeljivanje relativnih graničnih okvira predviđenim objektima.

Prošireni osnovni model je obučen na skupu podataka BVS u trajanju od oko deset minuta.

Zbog problema neizbalansiranosti klasa koji se javlja u skupu podataka BVS, dobija se rezultat modela koji ima visoku pristrasnost ka klasi *car*, kao što se vidi iz tabele 5.7. Model pogrešno daje izlaze graničnih okvira, kao što je prikazano na slici 5.11.

Tabela 5.7: Matrica konfuzije modela VGG16 za IoU prag 0.5 i prag ocene pouzdanosti za prepoznatu klasu 0.5

	car	motorbike	bus	truck	bez detekcije
car	85	26	0	1	0
motorbike	0	0	0	0	0
bus	0	0	0	0	0
truck	0	0	0	0	0
bez detekcije	4	71	1	0	0

(a) Stvarni okviri (b) Predviđeni okviri

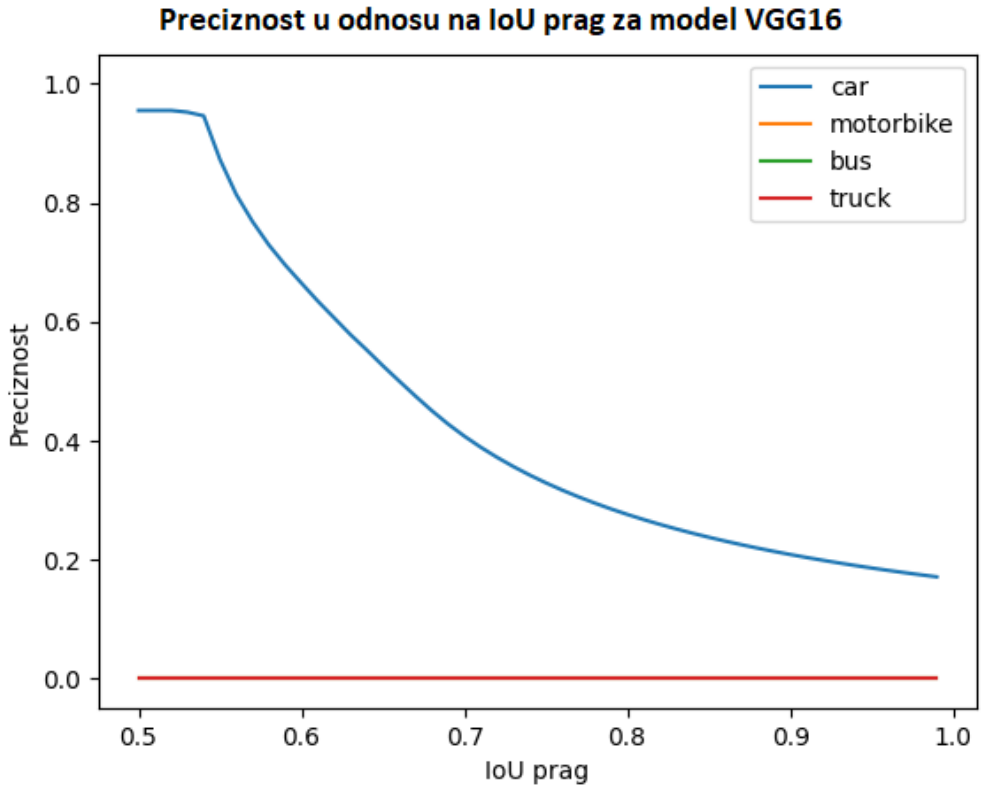
Slika 5.11: Primeri rezultata modela VGG16 nad test skupom podataka

Rezultati evaluacije modela su dati u tabeli 5.8. Iz rezultata se vidi da se model se ne ponaša dobro nad celim skupom podataka. Neki od razloga koji utiču da model daje nezadovoljavajuće rezultate je visoka neizbalansiranost između klasa i mala količina dostupnih podataka. Preciznost klasa u odnosu na IoU prag je prikazana na slici 5.12.

Kako je osnovni model VGG16 proširen, ne postoje slobodno dostupne odgovarajuće težine drugog modela koje mogu da se koriste za učenje sa prenosom znanja. Korišćenjem proširenog skupa podataka BVSstandard ili BVSstyle, dostupna količina podataka bi i dalje bila nedovoljna. Za obuku modela potrebno je desetine hiljada slika koje ni u proširenim skupovima BVS nisu dostupne.

Tabela 5.8: Rezultati modela VGG16

klasa	prosečna preciznost (jedn. 4.6)	srednja F_1-mera (jedn. 4.4)
car	0.95	0.57
motorbike	0.0	0.0
bus	0.0	0.0
truck	0.0	0.0
Srednja prosečna preciznost (jedn. 4.7)	0.24	
Srednja težinska F_1-mera (jedn. 4.5)	0.27	



Slika 5.12: Preciznost u odnosu na IoU prag za model VGG16. Za klase *motorbike*, *bus* i *truck* preciznost u svim vrednostima IoU iznosi 0.

5.4 Model za detekciju i predviđanje objekata YOLO

Kako model VGG16 ne može da dâ zadovoljavajuće rezultate, u okviru rada razmatra se i model za detekciju i predviđanje objekata YOLO. Koriste se dve konfiguracije modela YOLO. Prva konfiguracija modela YOLO služi za poređenje rezultata, i u mogućnosti je da predvidi do osamdeset klasa. Prva konfiguracija je prilagođena za skup podataka COCO (eng. *common objects in context*). Druga konfiguracija modela YOLO je u stanju da predvidi do četiri klase i prilagođena je za skup podataka BVS.

Predikcija modela YOLO koji je obučen na skupu podataka COCO

Koristi se model YOLO koji je obučen na skupu podataka COCO koji sadrži osamdeset klasa i 328 hiljada slika. Dalje u radu model YOLO obučen na skupu podataka COCO se označava sa YOLO_{coco}. Evaluacija modela se vrši nad test skupom podataka BVS. Matrica konfuzije je izračunata za vrednost IoU praga 0.5 i za ocenu pouzdanosti veću od 0.5. Tokom računanja prosečne preciznosti i srednje F_1-mere nije se uzimala u obzir ocena pouzdanosti klasa.

Iz rezultata matrice konfuzije (tabela 5.9) i metoda evaluacije modela (tabela 5.10) zaključuje se sledeće:

- Instance klase *car* model YOLO_{coco} predviđa sa visokom ocenom preciznosti. Predviđena je osamdeset i jedna instanca od ukupno osamdeset i devet. Iz matrice konfuzije može da se vidi da je model YOLO_{coco} za tri instance imao ocenu pouzdanosti manju od 0.5. Takođe, pogrešno se klasifikuje pet instanci kao instance klase *truck*. Prosečna preciznost za klasu *car* iznosi 0.88, dok srednja F_1-mera iznosi 0.91.

- Instance klase *motorbike* se predviđaju lošije od instanci klase *car*. Model YOLO_{coco} predviđa trideset sedam od ukupno devedeset sedam instanci. Većina instanci, tačnije četrdeset devet, za ocenu pouzdanosti ima vrednost manju od 0.5. Ocene pouzdanosti klase ukazuju da model nije siguran pri predviđanju instanci iz klase *motorbike*. Prosečna preciznost iznosi 0.77 i srednja F_1-mera iznosi 0.48.

- Klasa *bus* sadrži jednu sliku za test i ona se ne predviđa modelom YOLO_{coco}. Prosečna preciznost i srednja F_1-mera iznose 0.
- Instanca klase *truck* se uspešno predviđa modelom YOLO_{coco}. Prosečna preciznost iznosi 1.0 i srednja F_1-mera iznosi 0.99.

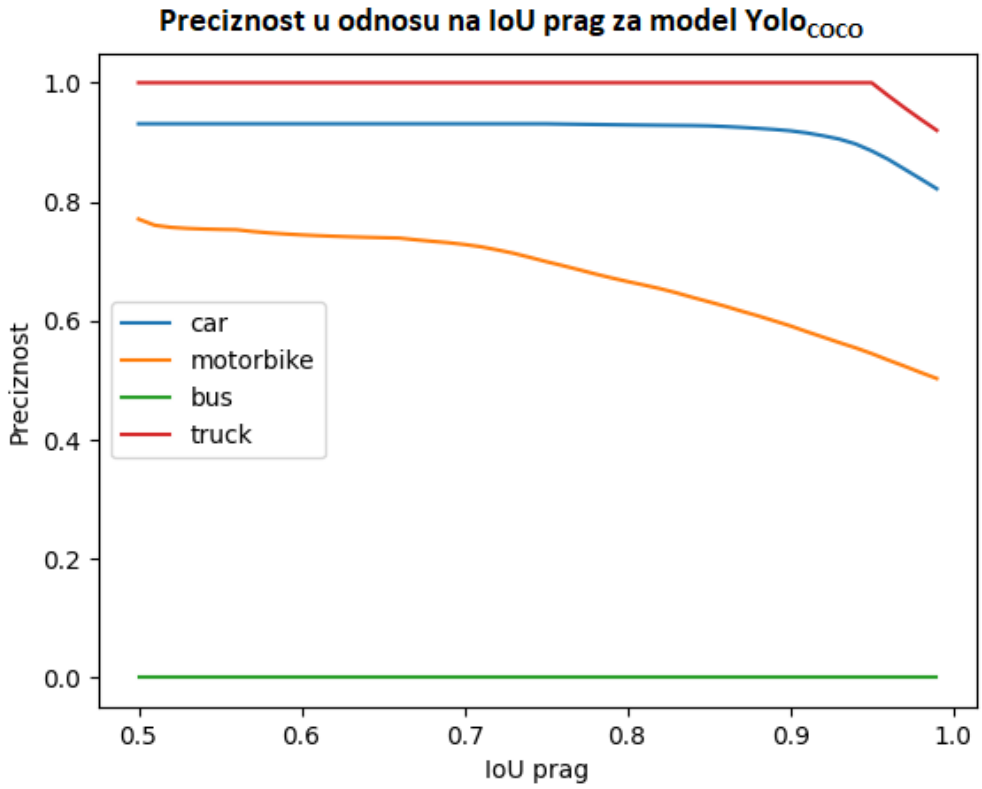
Tabela 5.9: Matrica konfuzije modela YOLO_{coco} za IoU prag 0.5 i prag ocene pouzdanosti za prepoznatu klasu veći od 0.5

	car	motorbike	bus	truck	bez detekcije
car	81	0	0	0	0
motorbike	0	37	0	0	0
bus	0	0	0	0	0
truck	5	0	0	1	0
bez detekcije	3	52	1	0	0

Tabela 5.10: Rezultati evaluacije modela YOLO_{coco}

klasa	prosečna preciznost (jedn. 4.6)	srednja F_1-mera (jedn. 4.4)
car	0.88	0.91
motorbike	0.77	0.48
bus	0.0	0.0
truck	1.0	0.99
Srednja prosečna preciznost (jedn. 4.7)	0.66	
Srednja težinska F_1-mera (jedn. 4.5)	0.69	

Preciznost klasa u odnosu na IoU prag je prikazana na slici 5.13. Model YOLO_{coco} ima srednju prosečnu preciznost (mAP) koja iznosi 0.66 i srednju težinsku F_1-meru koja iznosi 0.69. Kod predviđanja modela YOLO_{coco} možemo da zaključimo da dobro prepoznaje objekte klase *car* i *truck*. Problem koji se uočava kod predviđanja tih klasa je da postoje objekti koji imaju slične karakteristike. Takav problem se može uočiti na slici 5.14. Model YOLO_{coco} je uspešno klasifikovao 40% instanci klase *motorbike* iz test skupa BVS. Instanca klase *bus* se ne klasifikuje ispravno modelom YOLO_{coco}.



Slika 5.13: Preciznost u odnosu na IoU prag [0.5,1) modela YOLO_{coco}

Na slici 5.15 su prikazana neka predviđanja modela YOLO_{coco}. Slika prikazuje da su se uspešno detektovali neki od objekata koji pripadaju klasama *car* (gore desno), *motorbike* (sredina desno) i *truck* (dole levo). Na slici koja pripada klasi *motorbike* (sredina levo), uočavamo da je model uspešno detektovao čoveka, ali nije uspeo da detektuje objekat klase *motorbike*, odnosno ocena pouzdanosti je bila veoma mala, tj. iznosila je 0.2.

Slika 5.14: Slične karakteristike dveju različitih klasa, klasa *car* (levo) i klasa *truck* (desno)

Slika 5.15: Slike iz test skupa i predviđanja nad njima modelom YOLO_{coco}, klasa *car* (gore), klasa *motorbike* (sredina), klasa *truck* (dole levo) i klasa *bus* (dole desno). Objekti na slikama bez graničnih okvira nisu uspešno detektovani.

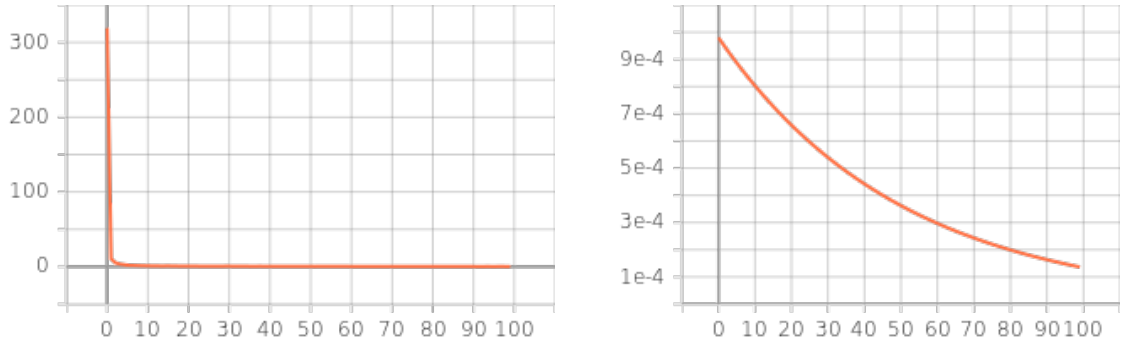
Obuka modela pomoću prenosa znanja

Radi poboljšanja rezultata, naredni modeli su obučeni koristeći težine modela obučene na skupu podataka COCO. Kako postoji razlika u broju klasa između skupa podataka BVS i COCO skupa, neki delovi težina su odbačeni.

Modeli su obučeni na slikama rezolucije do 512×512. Slikama je nasumično menjana veličina iz intervala 224×224 do 512×512, sa korakom 32×32. Obuka modela je bila ograničena na sto pedeset epoha. Za obuku modela korišćen je optimizator ADAM sa početnom stopom učenja $1e^{-3}$. Tokom obuke modela stopa obuke je množena sa faktorom 0.9. Takođe, postavljena je zastavica za rano zaustavljanje, ukoliko se u deset uzastopnih epoha funkcija greške nije umanjila. Iz skupa za obuku odvojeno je 10% podataka za validaciju.

Obuka modela pomoću prenosa znanja na skupu podataka BVS

Za detekciju objekata koristi se model YOLO koji je obučen na skupu podataka BVS koji sadrži četiri klase i 1872 slike. Dalje u radu, model YOLO obučen na skupu podataka BVS se označava sa YOLO_{BVS}. Na slici 5.16 je prikazana funkcija greške i stopa obuke modela YOLO_{BVS}. Iz funkcije greške se vidi da je došlo do ranog zaustavljanja obuke modela nakon devedeset devet epoha. Obuka modela je trajala oko dva sata.



Slika 5.16: Funkcija greške (levo) i stopa obuke (desno) modela YOLO_{BVS}

Iz rezultata matrice konfuzije (tabela 5.11) i metoda evaluacije modela (tabela 5.12) zaključuje se sledeće:

- instance klase *car* model YOLO_{BVS} uspešno predviđa u 100% slučajeva. Predviđeno je osamdeset devet od osamdeset devet instanci. Prosečna preciznost za klasu *car* iznosi 1.00 i srednja F_1-mera iznosi 0.99.

- instance klase *motorbike* model takođe uspešno predviđa u 100% slučajeva. Prosečna preciznost iznosi 1.00 i srednja F_1-mera iznosi 0.99, kao i kod klase *car*.
- instanca klase *bus* se predviđa kao instanca klase *motorbike*.
- instanca klase *truck* se predviđa kao instanca klase *car*.

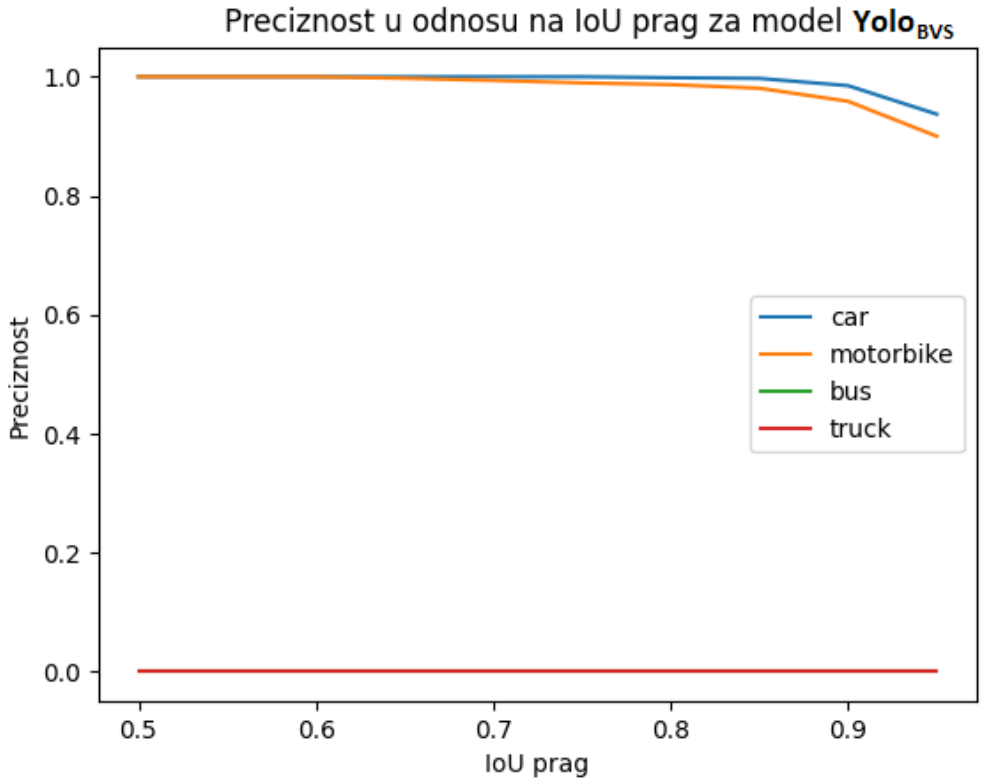
Tabela 5.11: Matrica konfuzije modela YOLO_{bvs} za IoU prag 0.5 i prag ocene pouzdanosti za prepoznatu klasu 0.5

	car	motorbike	bus	truck	bez detekcije
car	89	0	0	1	0
motorbike	0	97	1	0	0
bus	0	0	0	0	0
truck	0	0	0	0	0
bez detekcije	0	0	0	0	0

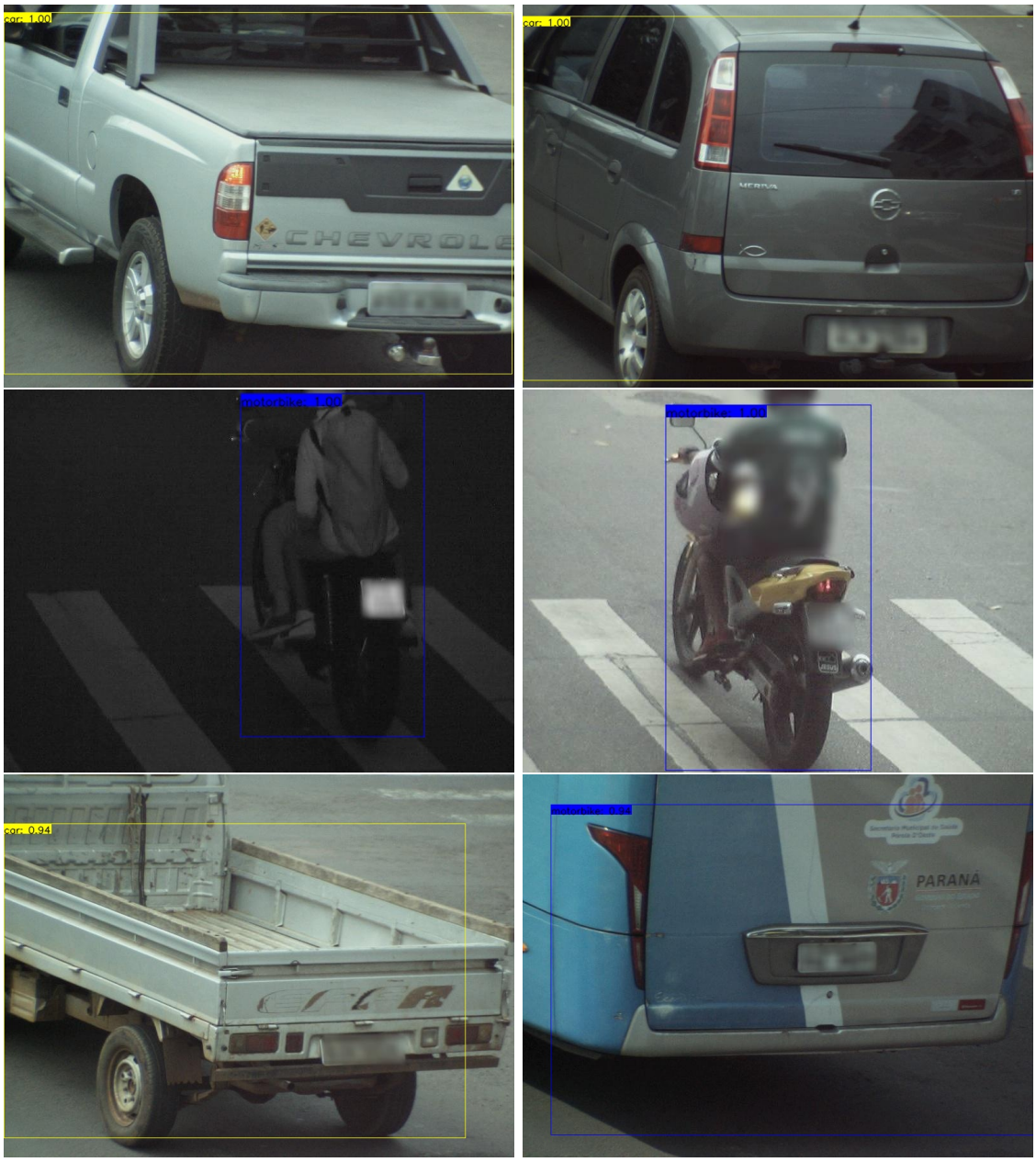
Tabela 5.12: Rezultati evaluacije modela YOLO_{bvs}

klasa	prosečna preciznost (jedn. 4.6)	srednja F_1-mera (jedn. 4.4)
car	1.00	0.99
motorbike	1.00	0.99
bus	0.00	0.00
truck	0.00	0.00
Srednja prosečna preciznost (jedn. 4.7)	0.5	
Srednja težinska F_1-mera (jedn. 4.5)	0.97	

Preciznost klasa u odnosu na IoU prag je prikazana na slici 5.17. Model YOLO_{bvs} ima srednju prosečnu preciznost (mAP), koja iznosi 0.5 i srednju težinsku F_1-meru koja iznosi 0.97. Srednja prosečna preciznost je manja u odnosu na model YOLO_{coco}. Srednja težinska F_1-mera je veća nego kod modela YOLO_{coco} što predstavlja izuzetno poboljšanje. Može se zaključiti da se model ponaša odlično pri detekciji instanci klasa *car* i *motorbike*, dok je za instancu klase *bus* predviđanje pogrešno. Može se primetiti da se problem, pomenut kod modela YOLO_{coco}, za neke instance klase *car* i *truck* i dalje javlja. Primer rezultata detekcije je prikazan na slici 5.18. Može se zaključiti da model YOLO_{BVS} daje bolje rezultate od modela YOLO_{coco}.



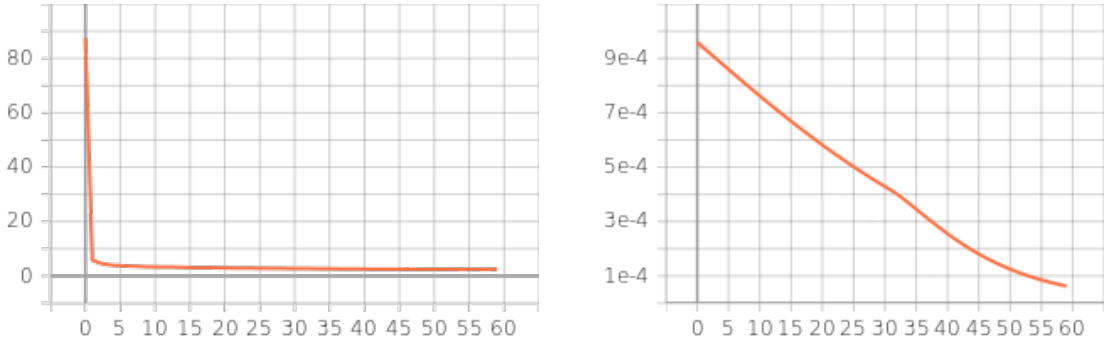
Slika 5.17: Preciznost u odnosu na IoU prag [0.5,1) modela YOLO_{BVS}, za klase *bus* i *truck* preciznost u svim vrednostima IoU iznosi 0



Slika 5.18: Slike iz test skupa i predviđanja nad njima modelom YOLO_{BVS}, klasa *car* (gore), klasa *motorbike* (sredina), klasa *truck* (dole levo) i klasa *bus* (dole desno)

Obuka modela pomoću prenosa znanja na skupu podataka BVSstandard

Za detekciju objekata koristi se model YOLO koji je obučen na skupu podataka BVSstandard koji sadrži četiri klase i 6952 slike. Dalje u radu model YOLO obučen na skupu podataka BVSstandard se označava sa $\text{YOLO}_{\text{BVS}standard}$. Funkcija greške modela i stopa obuke su prikazane na slici 5.19. Iz funkcije greške se vidi da je došlo do ranog zaustavljanja obuke modela nakon pedeset devet epoha. Obuka modela je trajala deset sati.



Slika 5.19: Funkcija greške (levo) i stopa obuke (desno) modela $\text{YOLO}_{\text{BVS}standard}$

Iz rezultata matrice konfuzije (tabela 5.13) i metoda evaluacije modela (tabela 5.14) zaključuje se sledeće:

- instance klase *car* model $\text{YOLO}_{\text{BVS}standard}$ predviđa uspešno u 100% slučajeva (tj. predviđeno je osamdeset devet od osamdeset devet instanci). Prosečna preciznost za klasu *car* iznosi 1.00, dok srednja F_1-mera iznosi 0.99. Ovi rezultati ukazuju na veoma dobro predviđanje modela $\text{YOLO}_{\text{BVS}standard}$ nad klasom *car*.

- instance klase *motorbike* model predviđa uspešno kao i instance klase *car*. Model $\text{YOLO}_{\text{BVS}standard}$ predviđa devedeset sedam od devedeset sedam instanci. Prosečna preciznost iznosi 1.00 i srednja F_1-mera iznosi 0.99.

- instanca klase *bus* nije predviđena modelom $\text{YOLO}_{\text{BVS}standard}$, tačnije ocena pouzdanosti je manja od 0.5.

- instanca klasa *truck* se pogrešno predviđa kao instanca klasa *car*.

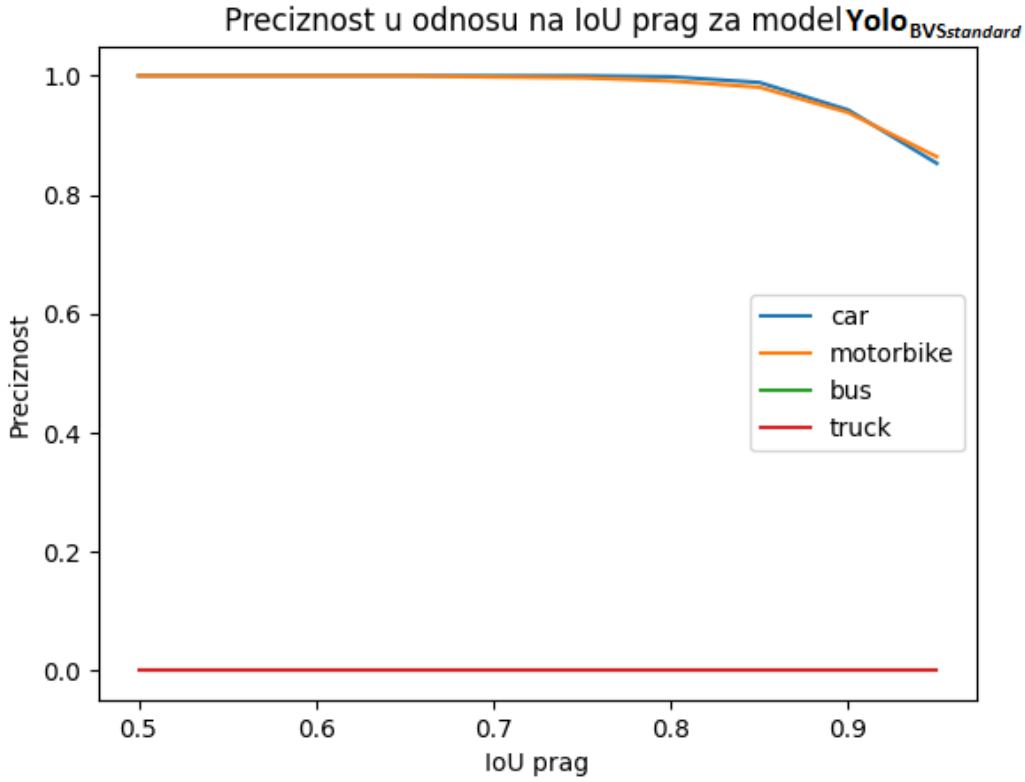
Tabela 5.13: Matrica konfuzije modela $\text{YOLO}_{\text{BVS}standard}$ za IoU prag 0.5 i prag pozdanosti za prepoznatu klasu 0.5

	car	motorbike	bus	truck	bez detekcije
car	89	0	0	1	0
motorbike	0	97	0	0	0
bus	0	0	0	0	0
truck	0	0	0	0	0
bez detekcije	0	0	1	0	0

Tabela 5.14: Rezultati modela $\text{YOLO}_{\text{BVS}standard}$

klasa	prosečna preciznost (jedn. 4.6)	srednja F_1-mera (jedn. 4.4)
car	1.00	0.99
motorbike	1.00	0.99
bus	0.00	0.00
truck	0.00	0.00
Srednja prosečna preciznost (jedn. 4.7)	0.50	
Srednja težinska F_1-mera (jedn. 4.5)	0.97	

Preciznost klasa u odnosu na IoU prag je prikazana na slici 5.20. Model YOLO $_{\text{BVS}standard}$ ima srednju prosečnu preciznost (mAP), koja iznosi 0.50 i srednju težinsku F_1-meru koja iznosi 0.97. Kod predviđanja modela YOLO $_{\text{BVS}standard}$ možemo da zaključimo da dobro prepoznaje objekte klase *car* i *motorbike*. Srednja prosečna preciznost je manja u odnosu na model YOLO$_{coco}$. Rezultati dobijeni modelom YOLO $_{\text{BVS}standard}$ su veoma slični sa rezultatima modela YOLO $_{\text{BVS}}$. Primer rezultata detekcije je prikazan na slici 5.21. Može se zaključiti da model YOLO $_{\text{BVS}standard}$ daje bolje rezultate od modela YOLO$_{coco}$.

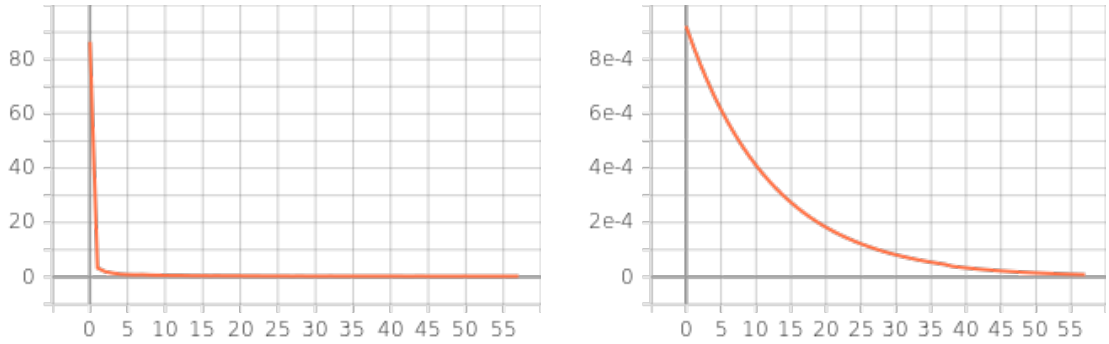


Slika 5.20: Preciznost u odnosu na IoU prag [0.5,1) modela YOLO $_{\text{BVS}standard}$. Za klase *bus* i *truck* preciznost u svim vrednostima IoU iznosi 0.

Slika 5.21: Slike iz test skupa i predviđanja nad njima modelom $\text{YOLO}_{\text{BVS}standard}$, klasa *car* (gore), klasa *motorbike* (sredina), klasa *truck* (dole levo) i klasa *bus* (dole desno). Objekti na slikama bez graničnih okvira nisu uspešno detektovani.

Obuka modela pomoću prenosa znanja na skupu BVSstyle

Za detekciju objekata koristi se model YOLO koji je obučen na skupu podataka BVSstyle koji sadrži četiri klase i 6952 slike. Dalje u radu model YOLO obučen na skupu podataka BVSstyle se označava sa $\text{YOLO}_{\text{BVS}style}$. Funkcija greške modela je prikazana na slici 5.22. Iz funkcije greške se vidi da je došlo do ranog zaustavljanja obuke modela nakon pedeset sedam epoha. Obuka modela je trajala oko pet sati.



Slika 5.22: Funkcija greške (levo) i stopa obuke (desno) modela $\text{YOLO}_{\text{BVS}style}$

Iz rezultata matrice konfuzije (tabela 5.15) i metoda evaluacije modela (tabela 5.16) zaključuje se sledeće:

- instance klase *car* model $\text{YOLO}_{\text{BVS}style}$ predviđa uspešno u 100% slučajeva (tj. predviđeno je osamdeset devet od osamdeset devet instanci). Prosečna preciznost za klasu *car* iznosi 1,0, dok srednja F_1-mera iznosi 0.99. Ovi rezultati ukazuju na savršeno predviđanje modela $\text{YOLO}_{\text{BVS}style}$ nad klasom *car*.
- instance klase *motorbike* se predviđaju podjednako dobro kao i instance klase *car*. Model $\text{YOLO}_{\text{BVS}style}$ predviđa devedeset sedam od devedeset sedam instanci. Prosečna preciznost iznosi 1.0 i srednja F_1-mera iznosi 0.98.
- instancu klase *bus* model $\text{YOLO}_{\text{BVS}style}$ uspešno predviđa. Prosečna preciznost i srednja F_1-mera iznose 1.00.
- instancu klase *truck* model pogrešno predviđa kao instancu klase *car*.

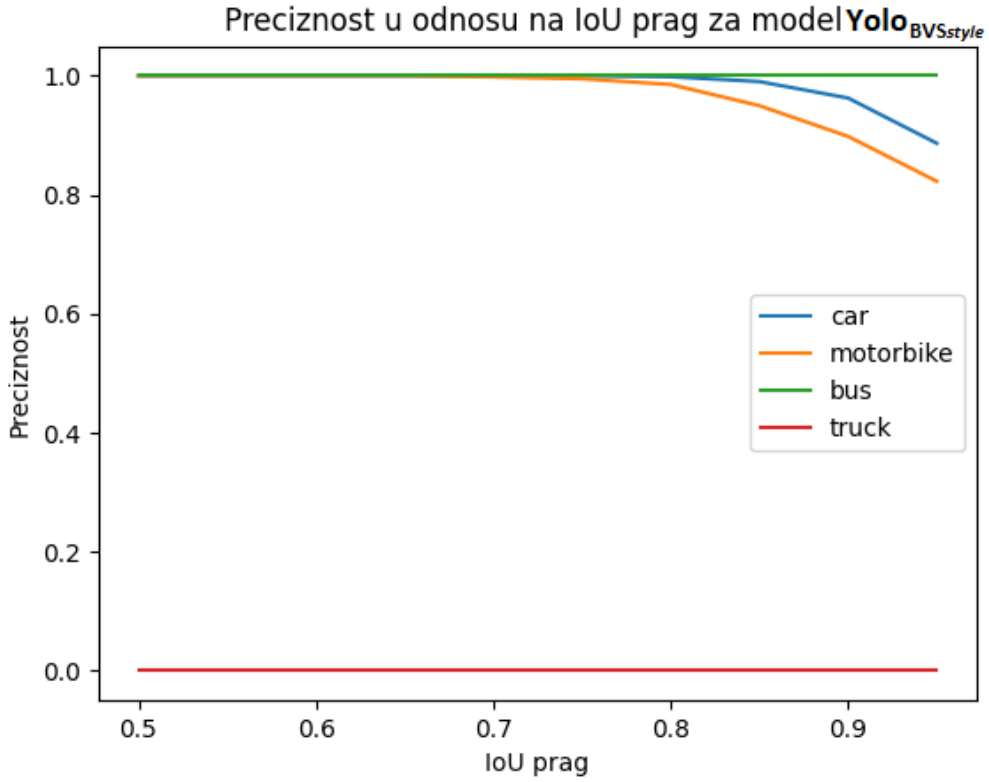
Tabela 5.15: Matrica konfuzije modela YOLO$_{\text{BVS}style}$ za IoU prag 0.5 i prag pouzdanosti za prepoznatu klasu 0.5

	car	motorbike	bus	truck	bez detekcije
car	89	0	0	1	0
motorbike	0	97	0	0	0
bus	0	0	1	0	0
truck	0	0	0	0	0
bez detekcije	0	0	0	0	0

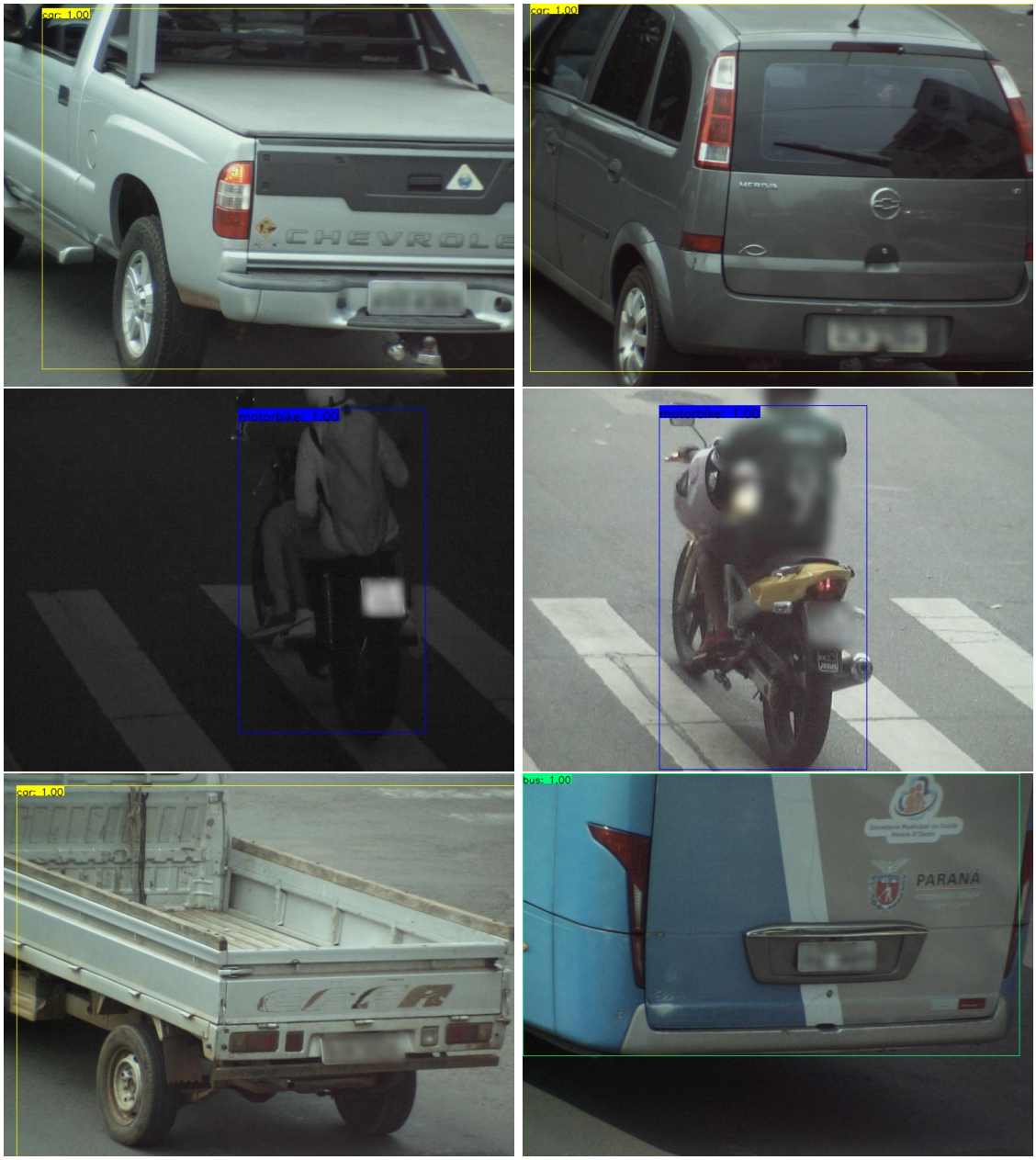
Tabela 5.16: Rezultati modela YOLO$_{\text{BVS}style}$

klasa	prosečna preciznost (jedn. 4.6)	srednja F_1-mera (jedn. 4.4)
car	1.00	0.99
motorbike	1.00	0.98
bus	1.00	1.00
truck	0.00	0.00
Srednja prosečna preciznost (jedn. 4.7)	0.75	
Srednja težinska F_1-mera (jedn. 4.5)	0.98	

Preciznost klasa u odnosu na IoU prag je prikazana na slici 5.23. Model YOLO $_{\text{BVS}style}$ ima srednju prosečnu preciznost (mAP), koja iznosi 0.75. Srednja prosečna preciznost je najveća u poređenju sa svim YOLO modelima. Srednja težinska F_1-mera iznosi 0.98 i ona takođe, predstavlja najbolji dobijeni rezultat. Kod predviđanja modela YOLO $_{\text{BVS}style}$ možemo da zaključimo da dobro predviđa objekte klase *car*, *motorbike* i *bus*. Model YOLO $_{\text{BVS}style}$ daje najbolje rezultate. Gubitak, kao i kod modela YOLO $_{\text{BVS}standard}$ i YOLO $_{\text{BVS}}$, javlja se u predviđanju instanca klase *truck*. Primer rezultata detekcije je prikazan na slici 5.24.



Slika 5.23: Preciznost u odnosu na IoU prag [0.5,1) modela YOLO $_{\text{BVS}style}$



Slika 5.24: Slike iz test skupa i predviđanja nad njima modelom YOLO$_{\text{BVS}style}$, klasa *car* (gore), klasa *motorbike* (sredina), klasa *truck* (dole levo) i klasa *bus* (dole desno).

5.5 Komparativna analiza rezultata

Model VGG16 je obučen na podacima BVS, dok je model YOLO obučen na podacima BVS, BVSstandard i BVSstyle. Dobijeni su rezultati koji su prikazani u tabeli 5.17.

Tabela 5.17: Rezultati obučenih modela za detekciju objekata

	Srednja prosečna preciznost (jedn. 4.7)	Srednja težinska F_1-mera (jedn. 4.5)
VGG16	0.24	0.27
YOLO_{coco}	0.66	0.69
YOLO_{BVS}	0.50	0.97
$\text{YOLO}_{\text{BVS}standard}$	0.50	0.97
$\text{YOLO}_{\text{BVS}style}$	0.75	0.98

Na osnovu srednje prosečne preciznosti, za modele navedene u tabeli 5.17 može da se uoči da najbolje rezultate daje model $\text{YOLO}_{\text{BVS}style}$, dok model VGG16 ne daje zadovoljavajuće rezultate. Neki od razloga koji utiču da model VGG16 daje nezadovoljavajuće rezultate je visoka neizbalansiranost između klasa i mala količina dostupnih podataka. Što se tiče modela YOLO_{BVS} i $\text{YOLO}_{\text{BVS}standard}$, koji daju iste rezultate za metrike evaluacije, došlo je do neznatnog ali, nedovoljnog poboljšanja kod modela $\text{YOLO}_{\text{BVS}standard}$ u odnosu na model YOLO_{BVS}. Naime, došlo je do poboljšanja prepoznavanja instance klase *bus*, ali ocena pouzdanosti nije prešla zadati prag (0.5).

Glava 6

Zaključak

U ovom radu je rešavan problem proširivanja skupa podataka koji se koriste za obučavanje modela za detekciju objekata. Skup podataka je proširivan slikama koje su dobijene standardnim transformacijama (translacija, okretanje, menjanje boja i itd.) i sintetički generisanim slikama od strane generativnih suparničkih mreža. Ukupno je dodato oko pet hiljada slika u skup podataka. Tako prošireni skupovi podataka, BVSstandard i BVSstyle, zasebno su korišćeni za obučavanje modela za detekciju objekata YOLO, dok je model VGG16 obučen na originalnom skupu podataka BVS. Radi poboljšanja rezultata kod modela YOLO, korišćena je i obuka sa prenosom znanja. Težine modela koje su se koristile su bile naučene na skupu podataka COCO.

Rezultati koji se dobijaju nakon obuke modela YOLO na skupovima podataka BVS, BVSstandard i BVSstyle su bolji od rezultata koji se dobijaju sa modelom VGG16. Kod modela YOLO koji su obučeni na skupovima podataka BVS, BVSstandard i BVSstyle, iz rezultata može da se zaključi da se preciznost popravila kod klasa *car* i *motorbike* u odnosu na model YOLO_{COCO}. Kod modela YOLO koji je obučen na skupovima podataka BVS i BVSstandard, srednja prosečna preciznost je opala jer se instance klase *truck* više ne prepoznaju, što može da bude uzrok deljenja karakteristika klase *truck* sa klasom *car*. Model YOLO koji je obučen na skupu podataka BVSstyle uspešno predviđa i instancu klase *bus*. Može se zaključiti da model $\text{YOLO}_{\text{BVS}style}$ daje najbolje rezultate.

Bolji rezultati za detekcije objekta mogu se dobiti povećanjem količine podataka koja je dostupna modelima *StlyeGAN2* i njihovom obukom duži vremenski period. Time bi se skup za obuku proširio kvalitetnijim i raznovrsnijim slikama. Unapređenje takođe može da se postigne korišćenjem unije skupova podataka BVSstandard i

BVSstyle.

Zahvaljujući razvoju novih tehnologija, možda će biti moguće generisati sintetičke podatke potpuno novim metodama. Na primer, modeli *DALL-E 2* [42] i *MidJourney* [43] su u stanju da generišu realistične slike na osnovu njihovih opisa u tekstualnom obliku. Ovakav pristup, međutim, zahteva nove podatke, tj. tekstualne opise slika.

Bibliografija

[1] Yoshua Bengio, Yann LeCun, et al. Scaling learning algorithms towards ai. *Large-scale kernel machines*, 34(5):1–41, 2007.

[2] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[3] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

[4] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[5] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 17–36. JMLR Workshop and Conference Proceedings, 2012.

[6] Lorenzo Brigato and Luca Iocchi. A close look at deep learning with small data. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2490–2497. IEEE, 2021.

[7] Lanlan Liu, Michael Muelly, Jia Deng, Tomas Pfister, and Li-Jia Li. Generative modeling for small-data object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6073–6081, 2019.

[8] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *Advances in Neural Information Processing Systems*, 33:7559–7570, 2020.

[9] Dana H Ballard and Christopher M Brown. Computer vision. englewood cliffs. *J: Prentice Hall*, 1982.

[10] Thomas Huang. Computer vision: Evolution and promise, 1996.

[11] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[12] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019.

[13] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[15] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, et al. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2896–2907, 2017.

[16] Yali Amit, Pedro Felzenszwalb, and Ross Girshick. Object detection. *Computer Vision: A Reference Guide*, pages 1–9, 2020.

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[18] Imagenet challange. https://image-net.org/challenges/LSVRC/2013/.

[19] Deepthi Narayan, Srikanta Murthy, and G Hemantha Kumar. Image segmentation based on graph theoretical approach to improve the quality of image segmentation. *International Journal of Computer and Information Engineering*, 2(6):1803–1806, 2008.

[20] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[21] Rohith Gandhi. R-cnn, fast r-cnn, faster r-cnn, yolo - object detection algorithms, Jul 2018.

[22] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[23] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[24] Jalel Ktari, Tarek Frikha, Monia Hamdi, Hela Elmannai, and Habib Hmam. Lightweight ai framework for industry 4.0 case study: Water meter recognition. *Big Data and Cognitive Computing*, 6(3), 2022.

[25] Mae Mu. Orange fruit photo, 2019. [Online; accessed August 16, 2022].

[26] Anđelka Zečević Mladen Nikolić. Naučno izračunavanje, 2019.

[27] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[28] Jie Feng, Xueliang Feng, Jiantong Chen, Xianghai Cao, Xiangrong Zhang, Licheng Jiao, and Tao Yu. Generative adversarial networks based on collaborative learning and attention mechanism for hyperspectral image classification, 2020.

[29] Anđelka Zečević Mladen Nikolić. Mašinsko učenje, 2019.

[30] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[31] Geoffrey M Hodgson. Microeconomics: Behavior, institutions, and evolution, samuel bowles, princeton university press and russell sage foundation, 2004, 584 pages. *Economics & Philosophy*, 22(1):166–171, 2006.

[32] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[33] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[34] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

[35] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[36] Yutaka Sasaki et al. The truth of the f-measure. *Teach tutor mater*, 1(5):1–5, 2007.

[37] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[38] DC Dowson and BV666017 Landau. The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455, 1982.

[39] Nikola Grulovic. Vehicle detection github, 2022. Available at `https://github.com/Grula/vehicle-detection`.

[40] Nikola Grulovic. Vehicle detection data, 2022. Available at `https://drive.google.com/file/d/1EkrO28-iBCVWyPqy5mJXn8P8EoKQ6xUH/view?usp=sharing`.

[41] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Data-efficient gans with diffaugment. Technical report, 2020.

[42] Mr D Murahari Reddy, Mr Sk Masthan Basha, Mr M Chinnaiahgari Hari, and Mr N Penchalaiah. Dall-e: Creating images from text. *UGC Care Group I Journal*, 8(14):71–75, 2021.

[43] Jack Daniel, Max. midjourney creating images from text, 2022.

Biografija autora

Nikola Grulović (*Beograd, Srbija, 14. jul 1993.*) je diplomirani Informatičar Univerziteta u Beogradu. Završio je „Petu Beogradsku Gimnaziju", prirodni smer 2012. godine u Beogradu. Matematički fakultet u Beogradu je upisao 2013. godine, smer Informatika i diplomirao je u septembru 2018. godine. Po završetku akademskih studija upisao je master studije, i u toku akademske 2019-2020 položio sve ispite predviđene planom i programom master studija. U periodu od novembra 2020. godine do aprila 2022. godine je radio kao stažista u kompaniji *AM Energia*, Sorokaba, Brazil. Tokom svoj prakse je radio na poboljšanju postojećih sistema i nadogradnjom sistema novim tehnologijama.