

UNIVERZITET U BEOGRADU
MATEMATIČKI FAKULTET



Lazar Jovanović

PRIMENA RAČUNARSKOG VIDA U
IDENTIFIKACIJI FUNKCIONALNIH REGIJA
NA SAMOUSLUŽNIM KASAMA

master rad

Beograd, 2024.

Mentor:

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

Članovi komisije:

dr Jovana Kovačević, vanredni profesor
Univerzitet u Beogradu, Matematički fakultet

dr Aleksandar Kartelj, vanredni profesor
Univerzitet u Beogradu, Matematički fakultet

Datum odbrane: _____

Naslov master rada: Primena računarskog vida u identifikaciji funkcionalnih regija na samouslužnim kasama

Rezime:

Rad istražuje primenu dubokog učenja u segmentaciji funkcionalnih regija na samouslužnim kasama, sa fokusom na modele Mask R-CNN i YOLOv8. Rad obuhvata kompletan proces od prikupljanja i anotacije podataka do implementacije i evaluacije modela.

Kreiran je specijalizovani skup podataka ekstrahovan iz video snimaka samouslužnih kasa koji uključuje anotacije ključnih regija. Ovaj skup podataka ne samo da je poslužio za treniranje i evaluaciju modela, već je i pružio uvid u specifične izazove segmentacije u kontekstu maloprodaje.

Rezultati su pokazali da oba modela uspešno detektuju i segmentiraju ključne regije, ali se suočavaju sa izazovima u razlikovanju vizuelno sličnih zona za preuzimanje i odlaganje proizvoda.

Rad zaključuje da razvijeni modeli imaju značajan potencijal za primenu u sistemima za automatizaciju i optimizaciju procesa na samouslužnim kasama, otvarajući mogućnosti za napredne analize ponašanja kupaca i detekciju neregularnosti. Ovo istraživanje doprinosi razvoju inteligentnih sistema u maloprodaji, pružajući osnovu za dalja istraživanja i praktične primene u ovoj oblasti.

Ključne reči: mašinsko učenje, duboko učenje, računarski vid, segmentacija, YOLOv8, Mask R-CNN

Sadržaj

1	Uvod	1
1.1	Uloga samouslužnih kasa i rast zastupljenosti	2
1.2	Regije na samouslužnim kasama	2
1.3	Cilj rada	3
2	Pregled osnovnih gradivnih elemenata	4
2.1	Neuronske mreže	4
2.2	Konvolutivne neuronske mreže	5
2.3	Detekcija objekata	7
2.4	Segmentacija	8
3	Pregled korišćenih modela	10
3.1	Mask R-CNN	10
3.2	YOLO	17
4	Metrike za evaluaciju modela	25
4.1	Osnovne metrike i koncepti	25
4.2	Napredne metrike za evaluaciju performansi	27
4.3	Specifične metrike za segmentaciju	28
5	Pripema podataka	29
5.1	Prikupljanje podataka	29
5.2	Anotiranje podataka	31
5.3	Podela podataka	32
6	Razvoj modela i rezultati	33
6.1	Konfiguracija modela	33
6.2	Funkcija greške YOLOv8 modela	34
6.3	Funkcija greške Mask R-CNN modela	35
6.4	Hardverska i softverska infrastruktura	35
6.5	Evaluacija i rezultati	35

SADRŽAJ

6.6	Vizualizacija rezultata	36
6.7	Diskusija o ograničenjima i primeni	39
7	Zaključak	40
	Bibliografija	41

Glava 1

Uvod

Samouslužna kasa (eng. *self-checkout* – *SCO*) je sistem koji omogućava kupcima da samostalno obave kupovinu bez potrebe za kasirima. Njeno korišćenje podrazumeva samostalno skeniranje proizvoda kao i samostalno plaćanje. Najčešće se koriste u supermarketima, iako nisu neuobičajene ni u drugim prodajnim objektima poput robnih kuća (npr. *Ikea*). Ovi sistemi dizajnirani su da poboljšaju efikasnost procesa plaćanja, smanje vreme čekanja i optimizuju korisničko iskustvo pri kupovini.

Ključne karakteristike samouslužnih kasa su:

- Interaktivni interfejs: SCO sistemi su opremljeni korisničkim interfejsom koji vodi kupce kroz proces skeniranja proizvoda, biranja načina plaćanja i završetka transakcije.
- Tehnologija skeniranja: upotreba različitih sofisticiranih tehnologija za identifikaciju proizvoda pomoću računarskog vida (eng. *computer vision* - *CV*), RFID tehnologije kao i skenera.
- Metode plaćanja: podrška različitih metoda plaćanja, uključujući gotovinu, kreditne ili debitne kartice i mobilne aplikacije.
- Sigurnosne funkcije: sigurnosni mehanizmi poput težinskih senzora i video nadzora za verifikaciju proizvoda i sprečavanje potencijalnih zloupotreba.

Za efikasno funkcionisanje sistema poput detekcije krađe ili personalizacije preporuka, ključno je da sistem veštačke inteligencije precizno prepozna specifične regije unutar prodajnog prostora, poput vage na samouslužnim kasama. Razumevanje lokacija ovih regija omogućava sistemu da, na primer, identifikuje da je kupac stavio limun na vagu i na osnovu toga ponudi relevantnu preporuku. Ovaj proces zahteva od sistema ne samo prepoznavanje objekta, već i precizno lociranje regije u kojoj se objekat nalazi.

Predmet ovog istraživanja je problem automatske detekcije ključnih regija unutar samouslužnih kasa. Fokus rada je na razvijanju i implementaciji metoda dubokog učenja

za segmentaciju i identifikaciju specifičnih regija od značaja, kao i na pripremi podataka za njihovo obučavanje.

1.1 Uloga samouslužnih kasa i rast zastupljenosti

Samouslužne kase su postale ključni element modernih maloprodajnih objekata, nudeći brzu alternativu tradicionalnim kasama. Posebno su korisne u vreme velikih gužvi, smanjujući redove i pružajući kupcima veću kontrolu nad procesom kupovine. Osim toga, SCO sistemi omogućavaju maloprodajnim lancima da optimizuju raspodelu radne snage, usmeravajući osoblje na druge zadatke koji mogu dodatno poboljšati korisničko iskustvo i operativnu efikasnost.

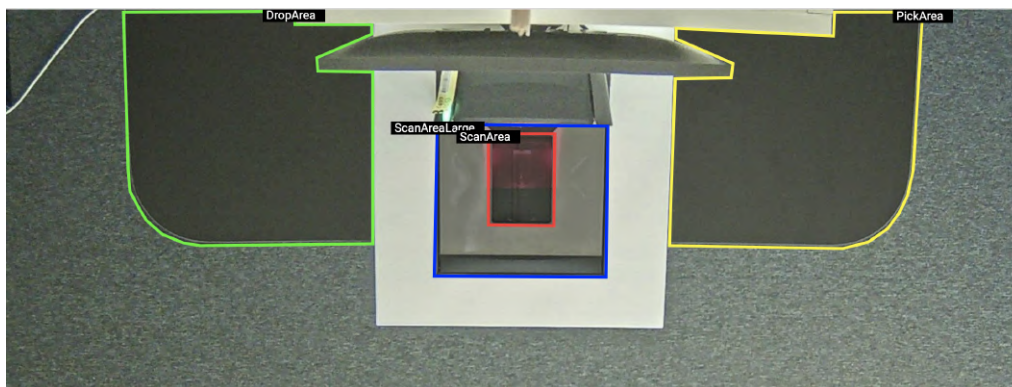
U SAD, udeo samouslužnih kasa se skoro dvostruko povećao u odnosu na period pre pandemije, predstavljajući 30% svih transakcija u prodavnicama u 2021. godini [1]. SCO mašine se sada nalaze u 96% od 38 000 uzorkovanih prodavnica u okviru 96 kompanija [1].

Usled rasta broja samouslužnih kasa, javlja se potreba i za rastom naprednih sistema koji ih podržavaju, poput upotrebe mašinskog učenja. Ovo ističe važnost razvoja generičkih i skalabilnih rešenja, što je jedan od motiva ovog rada. Razvoj takvih tehnologija ključan je za podršku sve većem broju SCO kasa, obezbeđujući njihovu efikasnost i unapređenje korisničkog iskustva u maloprodajnom sektoru.

1.2 Regije na samouslužnim kasama

U okviru samouslužnih kasa, postoje četiri glavne regije koje igraju ključnu ulogu u procesu skeniranja i plaćanja proizvoda:

- Velika skener zona (eng. *larger scanner area*): Ova regija predstavlja glavni skener koji se koristi za skeniranje proizvoda (plava regija na slici 1.1).
- Skener zona (eng. *scanner area*): Ovo je stakleni deo na skeneru unutar velike skener zone (crvena regija na slici 1.1).
- Zona za preuzimanje proizvoda (eng. *pick area*): Ovo je mesto gde se spušta korpa sa proizvodima pre skeniranja (žuta regija na slici 1.1).
- Zona za odlaganje proizvoda (eng. *drop area*): Ova regija je namenjena za odlaganje skeniranih proizvoda, gde se najčešće nalaze kese za pakovanje (zelena regija na slici 1.1).



Slika 1.1: Regije na samouslužnoj kasi

1.3 Cilj rada

Primarni cilj ovog rada je razvoj i implementacija naprednih pristupa segmentacije ključnih regija na samouslužnim kasama koristeći tehnike dubokog učenja. Rad se fokusira na analizu i poređenje performansi dva modela u oblasti računaskog vida: Mask R-CNN [2] i YOLOv8 [3]. Ovi modeli, koji predstavljaju različite pristupe detekciji i segmentaciji objekata, primenjeni su na specifičan problem identifikacije funkcionalnih zona na samouslužnim kasama, uključujući skener zonu, veliku skener zonu, zonu za preuzimanje i zonu za odlaganje proizvoda.

Posebna pažnja u ovom radu posvećena je procesu prikupljanja i pripreme podataka, što predstavlja ključni aspekt u razvoju efikasnih modela mašinskog učenja. Ovaj proces obuhvata ekstrahovanje slika iz video snimaka i njihovo anotiranje. Kreiranje ovog skupa podataka ne samo da omogućava treniranje i evaluaciju modela, već doprinosi i boljem razumevanju specifičnih izazova vezanih za segmentaciju regija na samouslužnim kasama.

Centralni deo istraživanja predstavlja razvoj i optimizaciju modela dubokog učenja. Kroz implementaciju i fino podešavanje, cilj je istražiti prednosti i mane svakog pristupa. Ovaj komparativni pristup omogućava dublje razumevanje sposobnosti modela da se nose sa specifičnim izazovima u realnom okruženju maloprodaje, kao što su varijacije u dizajnu kasa, uslovima osvetljenja i prisustvu različitih objekata.

Konačno, cilj ovog rada je da pruži ne samo teorijski uvid, već i da istraži praktičnu primenljivost ovih rešenja.

Glava 2

Pregled osnovnih gradivnih elemenata

U ovoj glavi predstavljeni su osnovni teorijski koncepti. Fokus je stavljen na fundamentalne elemente dubokog učenja i računarskog vida koji su neophodni za razumevanje arhitektura i funkcionisanja modela korišćenih u ovom radu. Uvedeni su pojmovi počevši od osnovnih principa neuronskih mreža, preko specifičnosti konvolutivnih neuronskih mreža, do koncepata u detekciji objekata i segmentaciji slika.

2.1 Neuronske mreže

Neuronske mreže su osnovni alat u mašinskom učenju. Inspirisane su strukturom ljudskog mozga i sastoje se od slojeva veštačkih neurona povezanih težinama. Svaki neuron prima ulaze, računa njihovu linearnu transformaciju koju propušta kroz aktivacionu funkciju i šalje izlaz dalje kroz mrežu. Ove mreže su široko primenjene u mnogim oblastima, uključujući kategorizaciju teksta, medicinsku dijagnostiku, prepoznavanje objekata na slikama, autonomnu vožnju, igranje igara, mašinsko prevođenje prirodnih jezika i modelovanje semantike reči [4].

Neuronske mreže funkcionišu kao parametrizovane reprezentacije koje mogu aproksimirati različite funkcije. Parametri mreže se podešavaju matematičkom optimizacijom, što može biti računski vrlo zahtevno.

Duboko učenje

Duboko učenje je posebna grana mašinskog učenja koja se fokusira na neuronske mreže sa velikim brojem slojeva, poznate kao duboke mreže. Ove mreže su posebno efikasne kada se radi sa velikim količinama sirovih podataka kao što su tekst ili slike. Jedna od glavnih

prednosti dubokog učenja je sposobnost mreža da automatski uče relevantne karakteristike iz podataka, što smanjuje potrebu za ručnim definisanjem tih karakteristika [4].

Osnovni model u dubokom učenju je potpuno povezana neuronska mreža (eng. *fully connected neural network - FCN*), koja može da aproksimira bilo koju neprekidnu funkciju. Ova mreža se sastoji od više slojeva neurona, gde se vrednost svakog neurona u sledećem sloju izračunava kao linearna kombinacija vrednosti svih neurona iz prethodnog sloja. Time se obezbeđuje da svaki neuron u sledećem sloju ima vezu sa svakim neuronom iz prethodnog sloja. Nakon računanja linearnih kombinacija, koristi se aktivaciona funkcija tj. nelinearna transformacija. Aktivaciona funkcija je ključna za aproksimaciju nelinearnih funkcija, jer samo kombinacija linearnih funkcija nije dovoljna za modelovanje nelinearnih odnosa [4].

2.2 Konvolutivne neuronske mreže

Konvolutivne neuronske mreže (eng. *convolutional neural networks - CNN*), su specijalizovane neuronske mreže dizajnirane za obradu podataka koji imaju mrežastu topologiju. Ti podaci mogu biti, na primer, vremenske serije, koje se mogu posmatrati kao jedno-dimenzionalna mreža, ili slike, koje se posmatraju kao dvodimenzionalna mreža piksela. CNN mreže su ostvarile izvanredne rezultate u mnogim praktičnim aplikacijama. Naziv „konvolutivna neuronska mreža” potiče od matematičke operacije konvolucije, koja se koristi u ovim mrežama. Konvolucija je posebna vrsta linearne operacije, a CNN mreže koriste konvoluciju umesto standardnog množenja matrica [5].

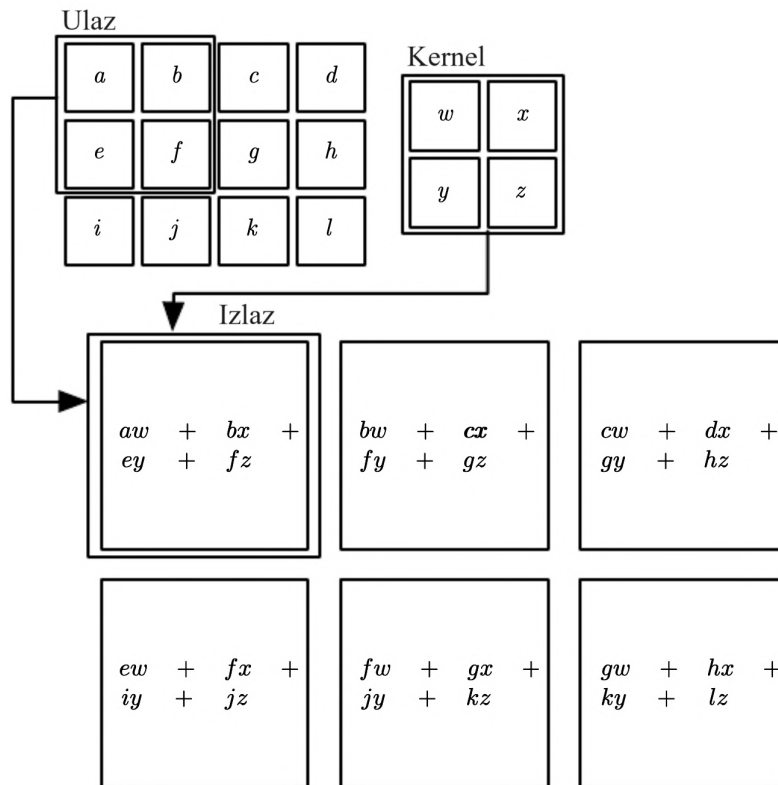
Konvolucija

Neka su date dve matrice I i K , dimenzija $m \times n$ i $p \times q$, tada je konvolucija S matrica koja se dobija na sledeći način:

$$S_{i,j} = (I * K)_{i,j} = \sum_m \sum_n I_{i-m,j-n} K_{m,n}$$

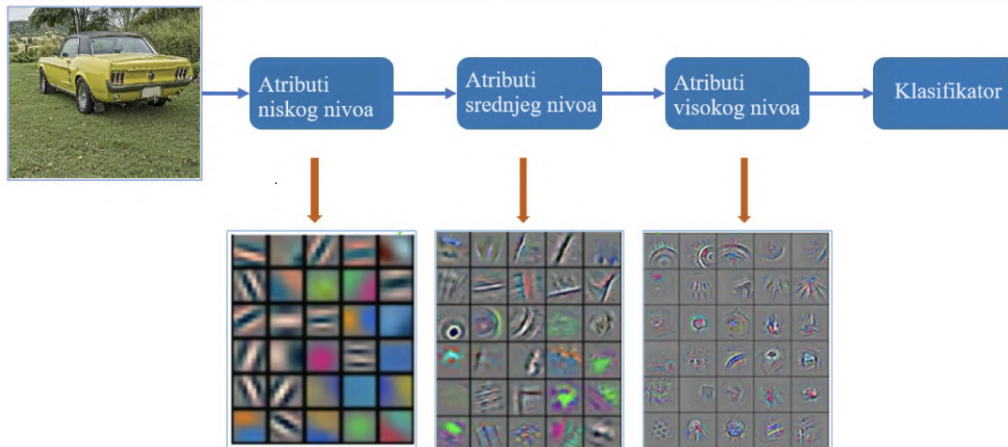
Matrica I obično se naziva ulaz, što je na primer slika, dok se matrica K uglavnom naziva kernel ili filter.

Na slici 2.1 prikazan je proces računanja konvolucije. Ulaz je matrica dimenzije 4×3 , a kernel matrica dimenzije 2×2 . Kernel se postavlja na prvi deo ulazne matrice, izračunava se konvolucija na toj poziciji tako što se elementi kernela množe sa odgovarajućim elementima ulazne matrice, a zatim se zbir tih proizvoda upisuje u odgovarajuću poziciju izlazne matrice. Nakon toga, kernel se pomera horizontalno i vertikalno izračunavajući sumu za svaku novu poziciju kernela. Tako se dobija izlazna matrica veličine 3×2 .



Slika 2.1: Računanje 2D konvolucije [5]

Konvolutivni kerneli služe za prepoznavanje atributa u ulaznim podacima. Atributi koji se prepoznaju se ne definišu već se uče. U početnim slojevima mreže, ovi kerneli detektuju jednostavne karakteristike, kao što su ivice i osnovni oblici. Kako se ide dublje u mrežu, kerneli postaju sposobni da prepoznaju sve složenije oblike i strukture. Na primer, u dubljim slojevima, mreža može detektovati složene karakteristike kao što su oblici lica ili specifični objekti. Ovaj proces omogućava mreži da izgradi hijerarhiju karakteristika, počevši od jednostavnih do složenih, čime se povećava preciznost i efikasnost prepoznavanja i klasifikacije u problemima kao što su analiza slika i videa. Ovo svojstvo prikazano je na slici 2.2.

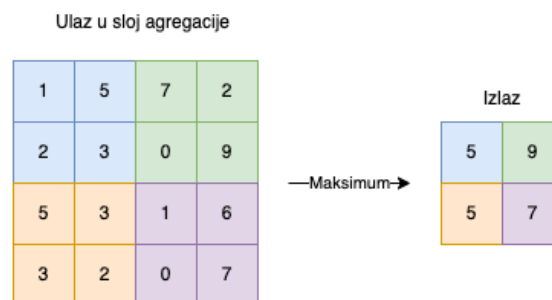


Slika 2.2: Primer kako mreža kretanjem od početnih ka dubljim slojevima detektuje sve složenije karakteristike [6].

Agregacija

Sloj agregacije (eng. *pooling*) objedinjuje informacije koje dobija iz prethodnog sloja primenjujući neku statistiku nad okolnim elementima matrice. Maksimum je najčešći izbor ove funkcije, dok to može biti i prosek ili nekakav drugi oblik agregacije. Uloga agregacije je smanjivanje broja parametara kao i smanjenje broja računskih operacija.

Na slici 2.3 vidimo da ukoliko je ulaz u ovaj sloj 4×4 , a agregira se 2×2 piksela, tada je izlaz 4 puta manji odnosno 2×2 .



Slika 2.3: Primer ulaza 4×4 i agregacije maksimumom 2×2

2.3 Detekcija objekata

Detekcija objekata (eng. *object detection*) je tehnika računarskog vida koja ima za zadatak identifikaciju i lokalizaciju različitih objekata unutar slike ili video zapisa. Primenu nalazi u raznim problemima poput opisivanja slika, praćenja objekata, itd [7]. Postoje dva glavna pristupa detekciji objekata: tradicionalni pristup i moderni pristup zasnovan na dubokim neuronskim mrežama. Tradicionalni pristup se oslanja na ručno definisanje

karakteristika. Jedan od primera tradicionalnog pristupa jeste histogram orijentisanih gradijenata (eng. *histogram of oriented gradients - HOG*) koji se koristio za detekciju pešaka [7]. Moderni pristup zasnovan na dubokim neuronskim mrežama značajno je poboljšao tačnost i efikasnost detekcije objekata. U okviru dubokog učenja, detekcija objekata može se grupisati u dve kategorije: dvostepena detekcija i jednostepena detekcija.

Dvostepena detekcija

Dvostepena detekcija (eng. *two stage detection*) razdvaja detekciju na dva odvojena koraka: generisanje predloga regija i klasifikacija sa lokalizacijom regija. U prvom koraku, koriste se algoritmi poput selektivne pretrage (eng. *selective search*) [8, 9] ili specijalizovane mreže (eng. *region proposal networks - RPN*) [10] koje daju potencijalne regije na kojima se nalaze objekti. U drugom koraku, mreža koristi ove predloge regija kako bi klasifikovala sadržaj svake i precizno lokalizovala objekat unutar nje. Dvostepeni detektori uglavnom postižu veću preciznost u poređenju sa jednostepenim detektorima jer razdvajaju generisanje predloga i klasifikaciju, omogućavajući mreži da se fokusira na analiziranje potencijalnih objekata. Međutim, upravo to razdvajanje je i razlog nešto sporijoj brzini izvršavanja kao i većem vremenu potrebnom za treniranje modela [3].

Jednostepena detekcija

Jednostepena detekcija (eng. *single stage detection*) se oslanja na skup unapred definisanih regija. Mreža sidrišnih tačaka (eng. *anchor points*) se postavlja preko ulazne slike, a svakoj tački dodeljuju se okviri (eng. *bounding box*) različitih dimenzija. Za svaki okvir na svakoj tački sidrišta, model daje predikciju da li postoji objekat unutar njega i modifikaciju lokacije i veličine kako bi se okvir bolje prilagodio objektu. Budući da postoji više okvira na svakoj tački neophodno je izvršiti naknadno procesiranje kako bi se uklonili duplikati i izabrala najbolja predikcija. Najpopularnija tehnika za postprocesiranje je NMS (eng. *non-maximum suppression*).

2.4 Segmentacija

Segmentacija slike je proces određivanja koje komponente prirodno „pripadaju zajedno” [11]. Ovo podrazumeva podelu slike na više semantički značajnih delova ili segmenata. Segmentacija omogućava identifikaciju i klasifikaciju objekata ili regija unutar slike, što je ključno za mnoge primene poput autonomne vožnje [12], medicinske dijagnostike [13], itd.

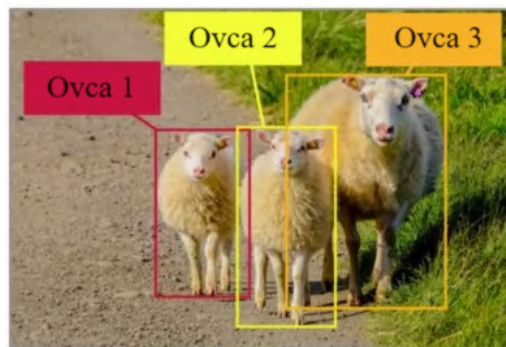
Postoje dva osnovna tipa segmentacije: semantička segmentacija i segmentacija instanci. Semantička segmentacija (eng. *semantic segmentation*) podrazumeva klasifikaciju svakog piksela u jednu od unapred definisanih klasa, dok segmentacija instanci (eng. *in-*

stance segmentation) ne samo da klasifikuje piksele u klase, već i razlikuje pojedinačne instance objekata unutar klase.

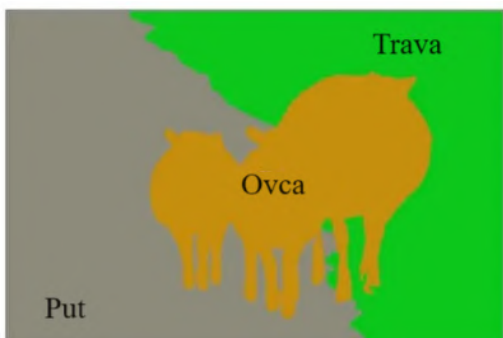
Na slici 2.4 prikazana je razlika između uvedenih pojmova. Na 2.4a nalazi se primer klasifikacije sa lokalizacijom. Slika je klasifikovana kao celina, što znači da se određuje samo jedna klasa. Na 2.4b prikazana je detekcija objekata, gde se svaka ovca nalazi unutar svog okvira. Na 2.4c i 2.4d prikazana je razlika između semantičke i segmentacije instanci.



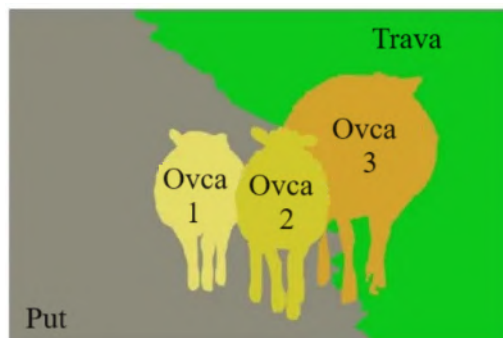
(a) Klasifikacija i lokalizacija



(b) Detekcija objekata



(c) Semantička segmentacija



(d) Segmentacija instanci

Slika 2.4: Komparativni primer klasifikacije, detekcije objekata i segmentacije [14]

Glava 3

Pregled korišćenih modela

U ovoj glavi predstavljena su dva ključna modela korišćena u radu: Mask R-CNN i YOLO. Ovi modeli izabrani su zbog svoje efikasnosti i široke primene u različitim zadacima računarskog vida. Detekcija objekata i segmentacija su ključni za razumevanje i interpretaciju vizuelnih podataka, a korišćenje naprednih modela dubokog učenja omogućava preciznu i brzu analizu slika i video zapisa. U nastavku su detaljno predstavljene karakteristike i prednosti svakog modela, kao i specifične tehnike za postizanje visokih performansi.

3.1 Mask R-CNN

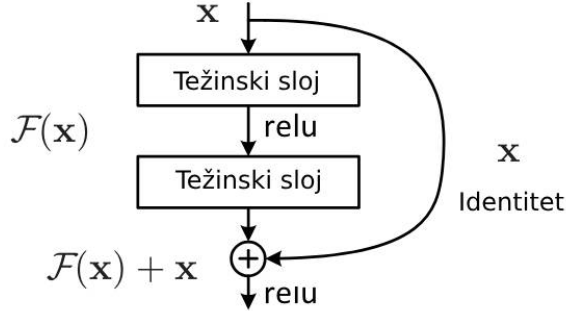
Mask R-CNN je napredni model za detekciju i segmentaciju objekata koji proširuje mogućnosti Faster R-CNN [10] dodavanjem grane za predviđanje maski za svaku regiju od interesa, paralelno sa postojećim granama za klasifikaciju i regresiju ograničavajućih okvira. Razvijen je od strane istraživača iz *Facebook AI Research* (FAIR). Mask R-CNN postiže preciznu detekciju objekata i segmentaciju instanci, što ga čini veoma efikasnim za različite zadatke u računarskom vidu poput detekcije melanoma [15]. U nastavku su detaljno objašnjene glavne komponente ovog modela.

Osnovna mreža

Osnovna mreža korišćena za ekstrakciju karakteristika je *ResNet* [16]. *ResNet* je duboka konvolutivna mreža koja koristi rezidualne blokove kako bi omogućila treniranje veoma dubokih slojeva smanjenjem problema nestajućih gradijenata. Ova mreža koristi se za ekstrakciju karakteristika iz ulazne slike, koje se zatim koriste za dalje procesiranje.

Osnovna ideja rezidualnog bloka je da se izlaz iz jednog sloja direktno kao ulaz u sledeći sloj uz dodavanje originalnog ulaza sloju. To se postiže uz pomoć takozvanih prečica (eng. *shortcut*) ili skip konekcija. Ove konekcije omogućavaju direktno prenošenje gradijenata

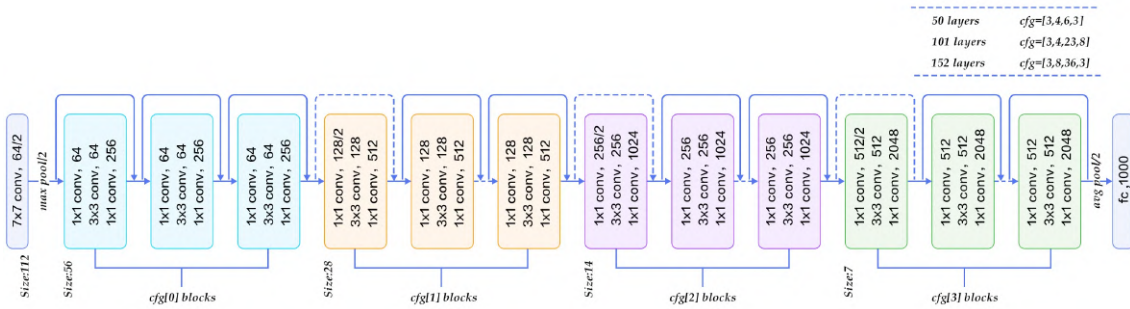
kroz mrežu, čime se olakšava treniranje veoma dubokih mreža. Rezidualni blok prikazan je na slici 3.1.



Slika 3.1: Ulaz u blok je x , a funkcija koja predstavlja operacije u bloku $F(x)$. Klasičan izlaz iz bloka bi bio $F(x)$, dok je u rezidualnom bloku izlaz $H(x) = F(x) + x$ [16].

ResNet arhitektura sastoji se od niza rezidualnih blokova, gde svaki blok ima tri komponente koje čine konvolucija, unutrašnja standardizacija (eng. *batch normalization*) i *ReLU* aktivaciona funkcija.

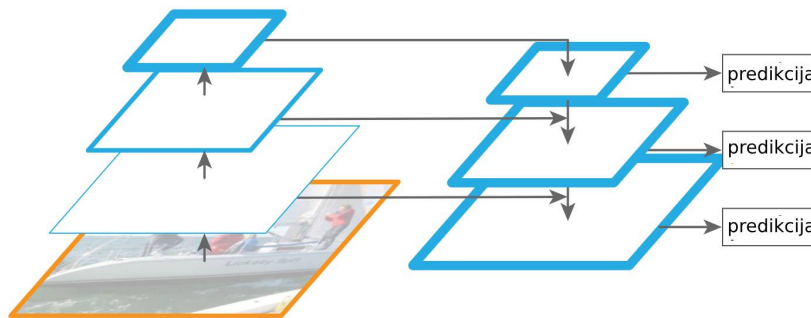
Postoji nekoliko varijanti ResNet arhitekture, kao što su ResNet18, ResNet34, ResNet50, ResNet101, ResNet152, gde broj označava broj slojeva. Na slici 3.2 prikazana je konfigurabilna arhitektura ove mreže. U konfiguraciji ResNet50 ljubičasti blok se ponavlja 6 puta, dok se u konfiguraciji ResNet101 ponavlja 23 puta. Zeleni blok se ponavlja 3 i 8 puta u ResNet50 i ResNet101 redom.



Slika 3.2: Arhitektura mreže *ResNet* [17]

Priširenje osnovne mreže

Feature pyramid network (FPN) je arhitektura koja poboljšava sposobnost modela da detektuje objekte na različitim skalama unutar slike. FPN se koristi u kombinaciji sa dubokim konvolutivnim mrežama kao što je ResNet, pružajući hijerarhije karakteristika koje omogućavaju preciznu detekciju objekata različitih veličina [18]. Na slici 3.3 prikazana je ilustracija ove arhitekture.

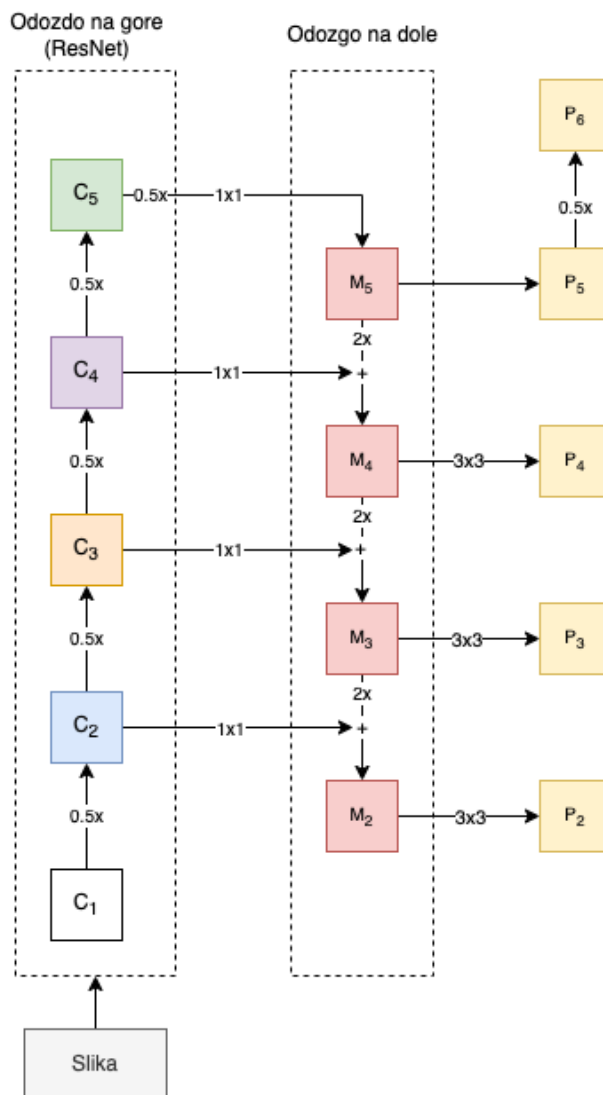


Slika 3.3: Ilustracija FPN arhitekture [18]

Kao što je već objašnjeno u poglavlju 2.2, kovolutivne mreže proizvode mape karakteristika različitim nivoima apstrakcije, gde početni nivoi sadrže karakteristike nižeg nivoa, dok dublji slojevi sadrže karakteristike visokog nivoa. FPN koristi ovu hijerarhiju da konstruiše piramidu karakteristika koja kombinuje informacije sa svih nivoa.

FPN čine dva pravca kretanja podataka: „odozdo na gore” (eng. *bottom-up*) i „odozgo na dole” (eng. *top-down*). Put odozdo na gore je zapravo, u slučaju ovog rada, ResNet. Kako se podaci kreću na gore, tako se rezolucija podataka smanjuje, dok se njihova semantička vrednost povećava. Na slici 3.4, ResNet moduli C_2 , C_3 , C_4 , C_5 obojeni su kao na slici 3.2, a konekcije između njih obeležene su sa $0.5x$ budući da se prostorna veličina ulaza između ovih modula smanjuje duplo.

Tokom prolaska odozgo na dole, rekonstruišu se veći slojevi. To se postiže povećavanjem rezolucije uz pomoć algoritma najbližeg suseda. Nakon uvećavanja, izlaz se sabira sa izlazom iz odgovarajućeg konvolucionog modula provučenog kroz 1×1 konvoluciju da bi se dobio odgovarajući broj kanala. Nad spojenim izlazima (M_i , $i \in \{2, 3, 4\}$, na slici 3.4) primenjuje se 3×3 konvolucija da bi se dobio izlaz iz mreže. Ova konvolucija ima ulogu smanjivanja aliasing efekta koji nastaje kao posledica uvećavanja slike i upravo se zbog toga ne primenjuje nad M_5 . Izlaz iz C_1 bloka nije korišćen budući da je njegova dimenzija velika, a semantička vrednost mala. P_6 dobijen je poduzorkovanjem P_5 sa korakom 2.



Slika 3.4: Arhitektura FPN

Predlaganje regija od interesa

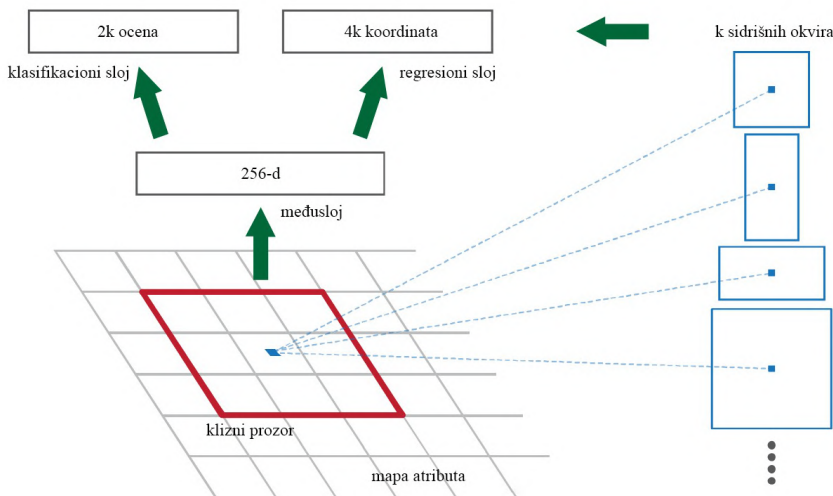
Nakon prolaska kroz FPN sledeći korak je kreiranje regija od interesa pomoću *region proposal network* (RPN). RPN je komponenta koja generiše predloge regija koje potencijalno sadrže objekte i funkcioniše kao podmreža unutar Mask R-CNN modela.

RPN koristi klizni prozor (eng. *sliding window*) na mapama karakteristika proizvedenih od strane FPN kako bi generisao predloge regija. Na svakoj poziciji kliznog prozora, RPN prelaže više regija različitih oblika i veličina koristeći unapred definisane sidrišne okvire. Svaki okvir ima različite odnose širine i visine kako bi pokrile različite proporcije objekta. Ova arhitektura je ilustrovana na slici 3.5.

RPN ima dve grane koje se nalaze na vrhu mreže: klasifikaciona i regresiona grana.

Ove dve grane se nazivaju i glava RPN (eng. *RPN head*). Klasifikaciona grana predviđa da li regija sadrži objekat ili ne (binarna klasifikacija). Regresiona grana predviđa precizne koordinate ograničavajućih okvira za svaku predloženu regiju, prilagođavajući ih tako da bolje odgovaraju stvarnim granicama objekata. Ista RPN glava se koristi na svim skalama mapi atributa generisanim od strane FPN.

Nakon generisanja predloga, koristi se NMS algoritam za filtriranje predloga regija. NMS uklanja preklapajuće predloge kako bi se zadržali samo najbolji.



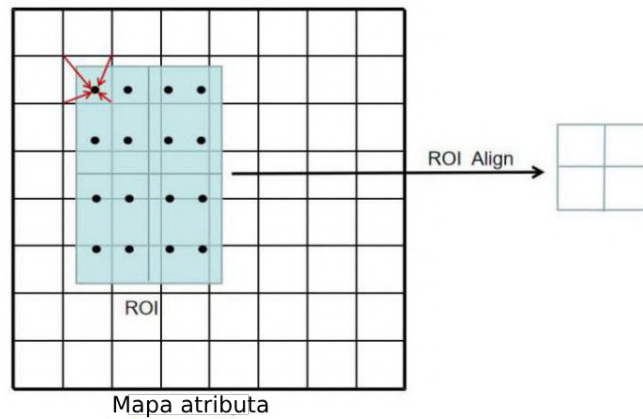
Slika 3.5: *region proposal network* [10]

Precizna agregacija regija od interesa

Generisani predlozi regija dalje prolaze kroz sloj za preciznu agregaciju regiona od interesa (eng. *ROI align*) koji precizno agregira attribute za svaki region od interesa. Svrha ovog sloja jeste da se regioni od interesa različitih dimenzija svedu na regione jednakih veličina budući da kasniji slojevi zahtevaju ulaz fiksne veličine. Sloj za grubo agregiranje regija od interesa (eng. *ROI pooling*), koji je uveden u Faster R-CNN [10] zaokruživao je koordinate prilikom agregacije što ne predstavlja problem kod detekcije objekata. Međutim, ova kvantizacija dovodi do neusklađenosti između ulazne slike i ekstrahovanih atributa koja loše utiče na generisanje segmentacionih maski koja zahteva veću preciznost u lociranju granica objekata.

Ovaj sloj prima predložene regije od strane RPN i mape atributa generisane od strane FPN. Svaki predloženi region se mapira na mapu atributa bez zaokruživanja koordinata, zatim se regioni dele na fiksni broj ćelija (7×7 za klasifikaciju i 14×14 za maske). Unutar svake ćelije biraju se 4 tačke za uzorkovanje. Koordinate ovih tačaka računaju se precizno, bez zaokruživanja. Za svaku tačku, vrednost se računa pomoću bilinearne interpolacije okolnih piksela. Konačna vrednost svake ćelije dobija se maksimalnom ili prosečnom

agregacijom 4 interpolirane vrednosti. Ovim postupkom dobija se mapa atributa fiksne veličine za svaki region od interesa. Ovaj postupak ilustrovan je na slici 3.6.



Slika 3.6: Primer računanja *ROI align* gde je dimenzija agregacije 2×2 [19]

Dvojna predikcija: detekcija i segmentacija

Nakon ROI Align sloja, Mask R-CNN arhitektura se grana u dve paralelne komponente: detekciona glava i segmentaciona glava. Ove dve komponente rade simultano, omogućujući mreži da istovremeno vrši detekciju objekata i segmentaciju. Obe komponente kao ulaz primaju mape atributa fiksne veličine dobijene kao izlaz iz ROI Align sloja, ali ih obrađuju na različite načine specifične za svoje zadatke.

Detekciona glava (eng. *box head*) sastoji se od nekoliko potpuno povezanih slojeva. Ima dvostruku ulogu:

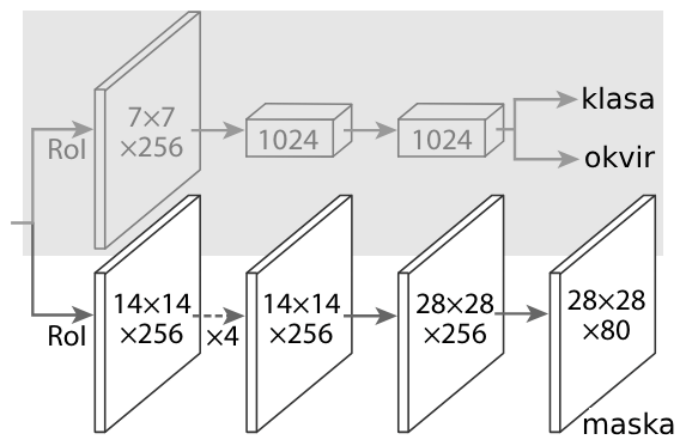
- Klasifikacija: predviđa klasu za svaki predloženi region.
- Regresija graničnog okvira: fino podešavanje koordinata graničnog okvira za precizniju lokalizaciju objekta.

Izlaz iz ovog sloja je vektor verovatnoća pripadnosti klasama i vrednosti koje definišu granične okvire (koordinate gornje leve tačke kao i širina i visina okvira).

Segmentaciona glava je inovativna komponenta koja omogućava Mask R-CNN da vrši segmentaciju. Sastoji se od 4 konvolutivna sloja praćena transponovanim konvolucionim slojem koji povećava rezoluciju. Završava se 1×1 konvolucijom koja proizvodi binarnu masku. Segmentaciona maska je mapa veličine 28×28 i generiše se za svaki objekat, nezavisno od klase kojoj pripada.

Ova paralelna struktura, prikazana na slici 3.7, omogućava modelu da istovremeno vrši detekciju objekata i njihovu segmentaciju, održavajući visoku preciznost u oba zadatka. Razdvajanje ovih zadataka u dve paralelne grane omogućava specijalizaciju svake

grane za svoj specifični zadatak, što se pokazuje efikasnijim u poređenju sa sekvencijalnim pristupom [2].



Slika 3.7: Paralelna detekciona i segmentaciona glava [2]

Celokupna arhitektura Mask R-CNN

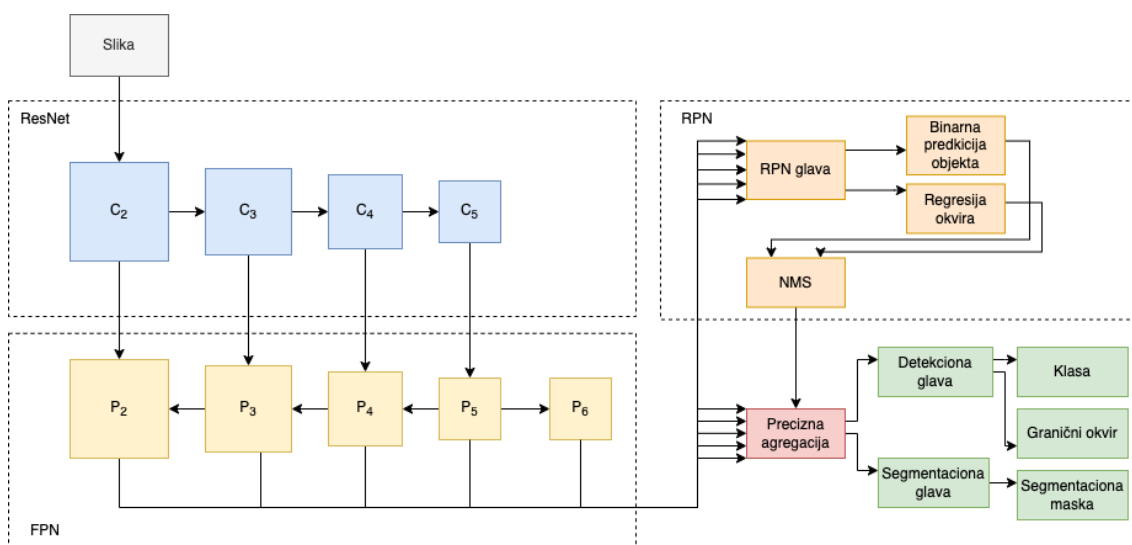
Mask R-CNN predstavlja sofisticiranu arhitekturu dubokog učenja dizajniranu za istovremeno detektovanje objekata i segmentaciju postizući veliku preciznost. Slika 3.8 prikazuje celokupnu arhitekturu ovog modela, ilustrujući sve ključne komponente i njihove interakcije.

Tok podataka kroz mrežu:

- Ulazna slika prolazi kroz osnovnu mrežu proizvodeći mape atributa.
- FPN obrađuje ove mape atributa, kreirajući piramidu mapa atributa na različitim skalama.
- RPN koristi FPN piramidu atributa da predloži potencijalne regije od interesa.
- Sloj za precinu agregaciju ekstrahuje mape atributa fiksne veličine za svaki predloženi region.
- Ekstrahovane mape atributa se paralelno obrađuju od strane glave za detekciju i glave za segmentaciju.
- Detekciona glava proizvodi klase objekata i precizne granične okvire.
- Segmentaciona glava generiše binarne maske segmentacije za svaki detektovani objekat.

Ključna snaga Mask R-CNN modela leži u efikasnoj interakciji između njegovih komponenti. FPN omogućava RPN-u da generiše kvalitetne predloge regiona na različitim skalama. *ROI align* obezbeđuje precizno agregiranje atributa, što je esencijalno za preciznu segmentaciju. Paralelni rad glava za segmentaciju i detekciju omogućava efikasno istovremeno rešavanje zadataka detakcije i segmentacije.

Mask R-CNN proizvodi tri izlaza za svaki detektovani objekat: klasu, granični okvir i binarnu segmentacionu masku.



Slika 3.8: Arhitektura Mask R-CNN

3.2 YOLO

YOLO (eng. *you only look once*) je familija modela za detekciju objekata u realnom vremenu. Za razliku od dvostepenih detektora poput Mask R-CNN, YOLO je jednostepeni detektor koji istovremeno predviđa granične okvire, klase objekata i njihove segmentacione maske. YOLOv8 je, u trenutku pisanja ovog rada, najnovija verzija u YOLO porodici i korišćen je u ovom radu. U nastavku su detaljno objašnjene glavne komponente YOLOv8 arhitekture.

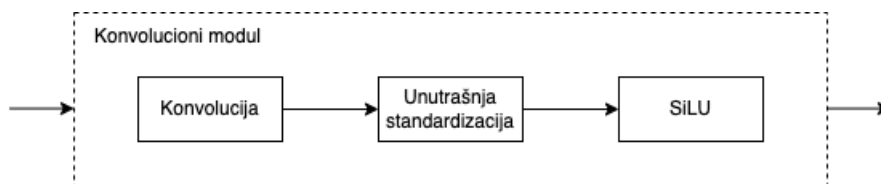
Gradivni blokovi

YOLOv8 arhitektura se sastoji od nekoliko ključnih gradivnih blokova koji zajedno omogućavaju efikasnu ekstrakciju karakteristika. Svaki od ovih blokova ima specifičnu ulogu u unapređenju performansi i efikasnosti modela.

Konvolucioni modul

Konvolucioni modul, prikazan na slici 3.9, je osnovni gradivni blok ove arhitekture. Sastoji se od sledećih komponenti:

- Konvolucioni sloj: vrši ekstrakciju lokalnih karakteristika iz ulaza.
- Unutrašnja standardizacija: normalizuje aktivacije, stabilizujući trening i ubrzavajući konvergenciju.
- *SiLU* aktivaciona funkcija: vrši nelinearnu transformaciju koja poboljšava protok gradijenata kroz mrežu.



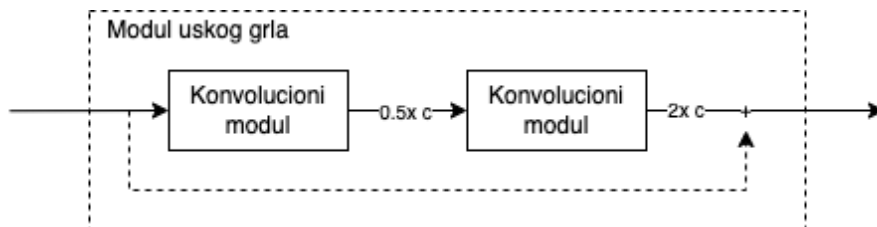
Slika 3.9: Konvolucioni modul YOLO arhitekture

Modul uskog grla

Modul uskog grla (eng. *bottleneck*) je specijalizovan da vrši kompresiju i ekspanziju informacija. Njegove komponente su:

- ulazni konvolucioni modul koji smanjuje broj kanala (prikazano kao $0.5x\ c$ na slici 3.10),
- izlazni konvolucioni modul koji povećava broj kanala tako da bude jednak broju kanala na ulasku u usko grlo (prikazano kao $2x\ c$ na slici 3.10).

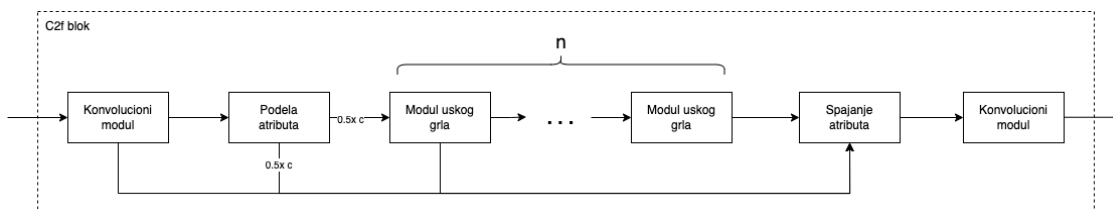
Ovaj modul može imati opcionu rezidualnu konekciju prikazanu isprekidanom linijom na slici 3.10.



Slika 3.10: Modul uskog grla YOLO arhitekture

Modul C2f

C2f (eng. *cross stage partial fusion fast*) blok predstavlja ključnu komponentu u YOLOv8 arhitekturi. Ovaj blok je evolucija originalnog CSP (eng. *cross stage partial*) bloka [20]. Smanjuje broj parametara i računskih operacija u poređenju sa standardnim CSP blokom uz održavanje visokih performansi. Na slici 3.11, prikazano je grananje podataka unutar C2f bloka. Jedan deo atributa prolazi kroz glavni put obrade, dok se drugi direktno prosleđuje ka izlazu. Parcijalno procesiranje podataka se pokazalo da smanjuje ukupnu računsku složenost uz minimalan gubitak informacija. Broj modula uskog grla, označen kao n na slici 3.11, je različit za različite veličine YOLO modela (dostupne veličine su: *nano*, *small*, *medium*, *large*, *extra large*).

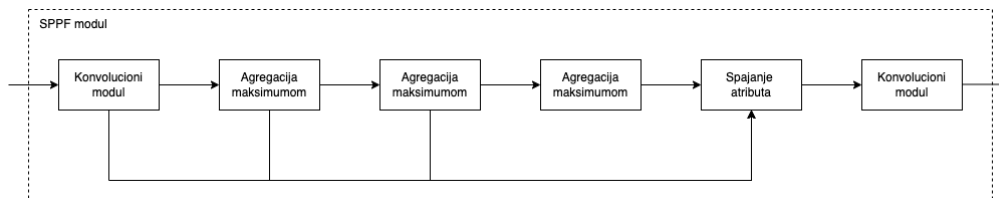


Slika 3.11: C2f modul YOLO arhitekture

Modul brze prostorne piramidalne agregacije

SPPF (eng. *spatial pyramid pooling - fast*) je unapređena verzija tradicionalnog SPP (eng. *spatial pyramid pooling*) modula, optimizovana za veću efikasnost i brže izvršavanje. Ovaj modul omogućava modelu da obrađuje karakteristike na različitim prostornim skalama čime povećava robustnost detekcije objekata različitih veličina.

Na slici 3.12, prikazan je ovaj modul. Operacije agregacije se primenjuju kaskadno sa parametrima postavljenim tako da je dimenzija izlaza ista kao dimenzija ulaza. Ovo omogućava jednostavno spajanje atributa bez potrebe za dodatnim operacijama skaliranja ili interpolacije.

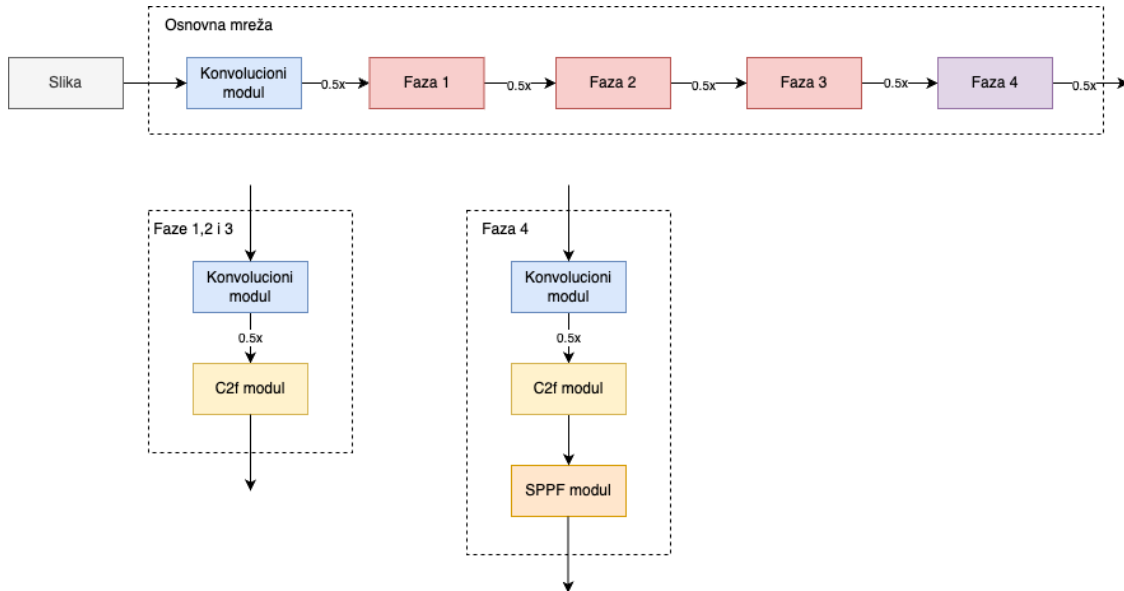


Slika 3.12: SPPF modul YOLO arhitekture

Osnovna mreža

Osnovna mreža korišćena za ekstrakciju karakteristika je modifikacija CSPDarknet53 [20], koja koristi napredne C2f module umesto CSP modula za smanjenje računskih opera-

cija uz održavanje performansi. Mreža proizvodi mape karakteristika na različitim nivoima (izlazi iz odgovarajućih faza prikazanih na 3.13) što poboljšava detekciju objekata različitih veličina.



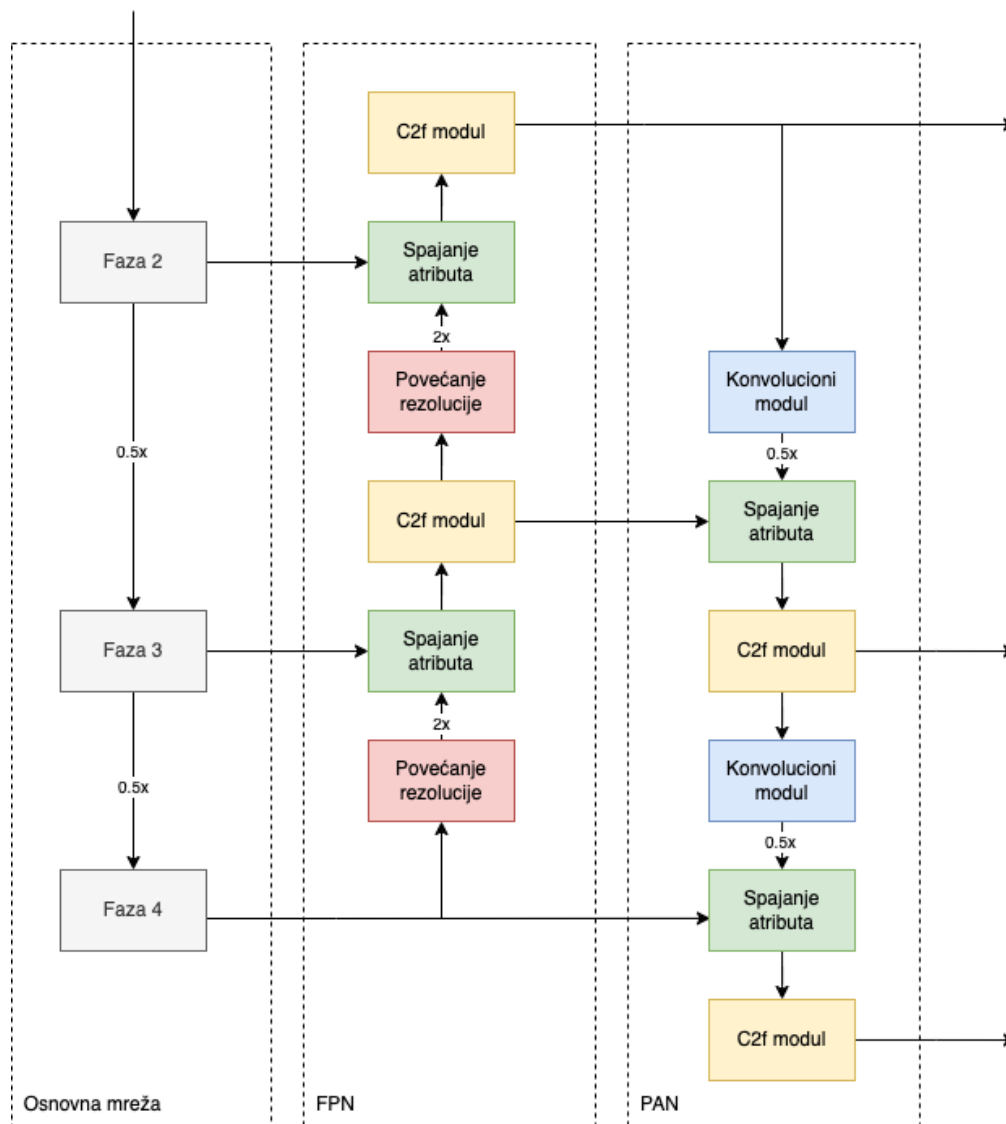
Slika 3.13: Osnovna mreža YOLOv8 arhitekture

Vrat mreže

Vrat (eng. *neck*) mreže u YOLOv8 arhitekturi kombinuje modifikovane verzije FPN i PAN (eng. *path aggregation network*) [21] struktura. Ova hibridna arhitektura, često nazivana FPN-PAN, omogućava efikasnu agregaciju i protok informacija između različitih nivoa karakteristika.

Uvođenje PAN mreže motivisano je potrebom za preciznijim očuvanjem prostornih informacija, što je ključno za poboljšanje lokalizacije piksela u zadacima segmentacije. Duboki slojevi neuronskih mreža, zbog svoje male prostorne rezolucije, često imaju poteškoća u preciznoj segmentaciji objekata. PAN rešava ovaj problem uvođenjem dodatnog prolaza „odozgo na dole”, koji se nadovezuje na postojeći „odozgo na dole” prolaz kroz FPN. Ključna inovacija PAN-a leži u korišćenju lateralnih konekcija, koje značajno skraćuju put između nižih i viših slojeva mreže. Ove konekcije dodatno poboljšavaju efikasno kombinovanje prostorno preciznih informacija iz nižih slojeva sa semantički bogatijim informacijama iz viših slojeva. Rezultat je bolja integracija lokalnih detalja i globalnog konteksta, što vodi ka preciznijoj segmentaciji objekata različitih veličina. Ovakva arhitektura posebno doprinosi poboljšanju performansi u segmentaciji malih objekata i finih detalja, koji su često izazovni za standardne arhitekture.

Arhitektura vrata prikazana je na slici 3.14.



Slika 3.14: Vrat YOLOv8 arhitekture

Glava mreže

Glava mreže predstavlja ključnu komponentu koja transformiše mape atributa dobijene iz vrata mreže u konačne predikcije. Dizajnirana je da istovremeno obavlja dva zadatka: detekciju objekata i njihovu segmentaciju. Ova dvostruka funkcionalnost omogućava modelu da ne samo locira i klasifikuje objekte, već i da precizno odredi njihove granice na nivou piksela.

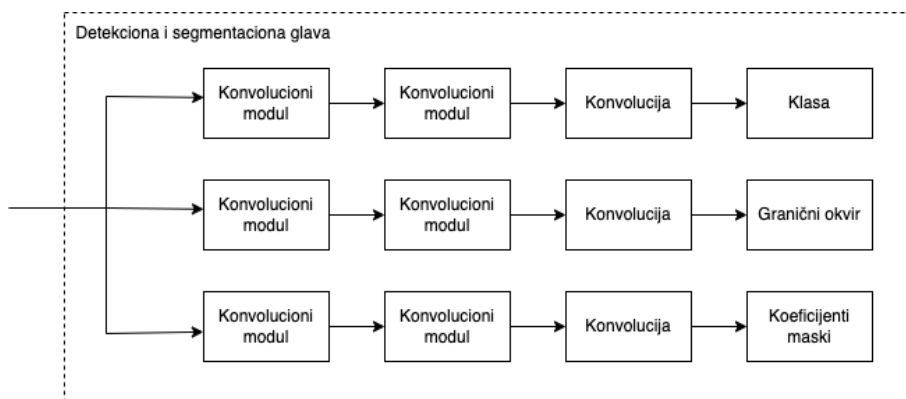
Struktura glave može se podeliti na dva glavna dela:

- Detekcioni deo: odgovoran za lociranje objekata i njihovu klasifikaciju. Ovaj deo proizvodi granične okvire i klase za detektovane objekte.

- Segmentacioni deo: odgovoran za generisanje preciznih segmentacionih maski za svaki detektovani objekat.

Oba dela rade paralelno, deleći zajednički ulaz, što omogućava efikasno izvršavanje oba zadatka.

Kompletna glava, prikazana na slici 3.15, izlaz daje u obliku konvolutivnih slojeva. Specifično je, za razliku od prethodnih verzija YOLO modela, da se za detekciju ne koriste prethodno definisane sidrišne tačke.



Slika 3.15: Glava za detekciju i segmentaciju

Ovaj model takođe uključuje granu *ProtoNet*, inspirisanu YOLACT [22] arhitekturom, koja omogućava efikasnu segmentaciju. Ova komponenta igra ključnu ulogu u generisanju segmentacionih maski uz održavanje računске efikasnosti. *ProtoNet* je mala potpuno konvoluciona mreža koja generiše prototipove maski uzimajući u obzir celu sliku. Izlaz iz ove mreže se kombinuje sa koeficijentima maski kako bi se proizvela segmentaciona maska instanci. Ovaj pristup omogućava mreži da održi prostornu koherenciju karakteristika, dok je izlaz brz i efikasan, jer se izbegava upotreba složene naknadne obrade. Dok *Mask R-CNN* zahteva lokalizaciju atributa unutar predloženih regiona kao i njihovu agregaciju, *YOLO* to izbegava generisanjem prototipova za celu sliku i direktnim korišćenjem ovih prototipova za segmentaciju.

Celokupna arhitektura YOLO

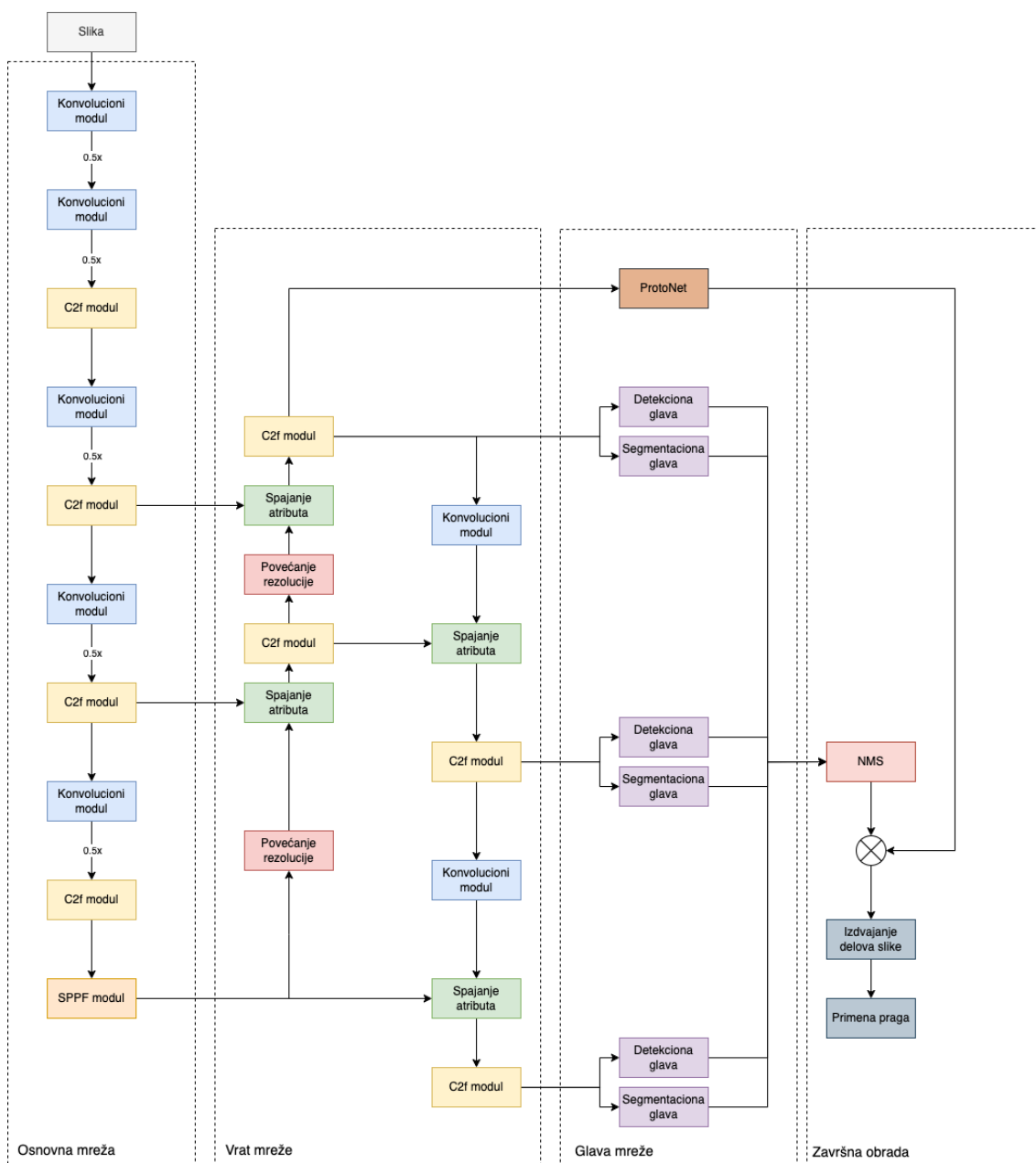
YOLOv8-seg predstavlja arhitekturu za istovremenu detekciju objekata i segmentaciju instanci. Ova mreža kombinuje efikasnost *YOLO* modela sa naprednim tehnikama za preciznu segmentaciju. Slika 3.16 prikazuje celokupnu arhitekturu *YOLOv8* modela za segmentaciju.

Tok podataka kroz mrežu:

- Ulazna slika prolazi kroz osnovnu mrežu proizvodeći mape atributa.

- Vrat mreže obrađuje ove mape atributa kombinujući informacije sa različitih nivoa.
- Glava mreže paralelno obrađuje informacije za detekciju i segmentaciju:
 - Detekciona glava predviđa granične okvire i klase objekata.
 - Segmentaciona glava predviđa koeficijente za kombinovanje sa prototipovima.
 - ProtoNet generiše prototipove maski.
- U toku završne obrade generiše se konačni izlaz tako što se:
 - primenjuje NMS za filtriranje preklapajućih okvira,
 - generišu finalne maske linearnom kombinacijom prototipova i koeficijenata,
 - zatim se maske seku u odnosu na predviđene granične okvire i,
 - primenjuje se prag (eng. *threshold*) za odstranjivanje predviđanja za koje model nije siguran.

YOLOv8 arhitektura za segmentaciju predstavlja značajan napredak u oblasti računarskog vida, omogućavajući brzu i preciznu detekciju objekata i segmentaciju instanci. Njegova sposobnost da efikasno obradi oba zadatka u jednom prolazu čini ga pogodnom za širok spektar primena, od autonomne vožnje do robotike i analize medicinskih slika.



Slika 3.16: Arhitektura YOLOv8 za segmentaciju

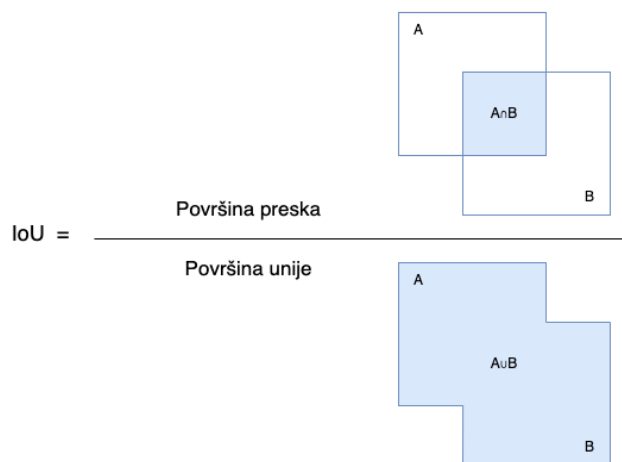
Glava 4

Metrike za evaluaciju modela

. Precizna evaluacija modela ključna je za merenje performansi i poređenje različitih pristupa. U oblasti detekcije objekata i segmentacije, metrike evaluacije pružaju kvantitativnu meru uspešnosti modela u lokalizaciji objekata, njihovoj klasifikaciji i preciznom određivanju granica na nivou piksela. U ovoj glavi predstavljene su ključne metrike korišćene za evaluaciju, a njihovo razumevanje neophodno je za objektivnu procenu i optimizaciju modela.

4.1 Osnovne metrike i koncepti

Presek nad unijom (eng. *intersection over union - IoU*) je osnovna metrika u računarskom vidu koja se koristi za merenje tačnosti detekcije objekata i segmentacije instanci. IoU kvantifikuje stepen preklapanja između predviđenog i stvarnog (eng. *ground truth*) graničnog okvira. Definiše se kao odnos površine preseka predviđenog i stvarnog regiona i površine njihove unije. Metrika je ilustrovana na slici 4.1.



Slika 4.1: Ilustracija IoU metrike

U kontekstu detekcije objekata, klasifikacija predviđanja kao tačnih ili netačnih zasniva se na stepenu preklapanja između predviđenih i stvarnih graničnih okvira, što se meri pomoću IoU metrike. Postavlja se prag za određivanje tačnosti detekcije i na osnovu njega definišu se sledeće kategorije:

- Stvarno pozitivne (eng. *true positive* - *TP*) detekcije su one gde je model uspešno detektovao objekat, a IoU između predviđenog i stvarnog graničnog okvira prelazi definisani prag. Na primer, ako IoU iznosi 0.6, a prag je 0.5, detekcija se smatra tačnom.
- Lažno pozitivne (eng. *false positive* - *FP*) detekcije su slučajevi kada je model predvideo objekat koji ne postoji.
- Lažno negativne (eng. *false negative* - *FN*) instance su objekti prisutni u stvarnosti koje model nije uspeo da detektuje ili je detektovao sa IoU ispod definisanog praga.
- Stvarno negativne (eng. *true negative* - *TN*) instance su negativne instance koje je model označio kao negativne. U detekciji objekata, koncept stvarno negativnih instanci nije direktno primenljiv jer se fokus stavlja na uspešno detektovane objekte, a ne područja slike gde objekti nisu prisutni.

Tačnost (eng. *accuracy*) se može definisati kao procenat pravilno klasifikovanih objekata u odnosu na ukupan broj objekata:

$$Acc = \frac{TP+TN}{TP+FP+FN+TN}$$

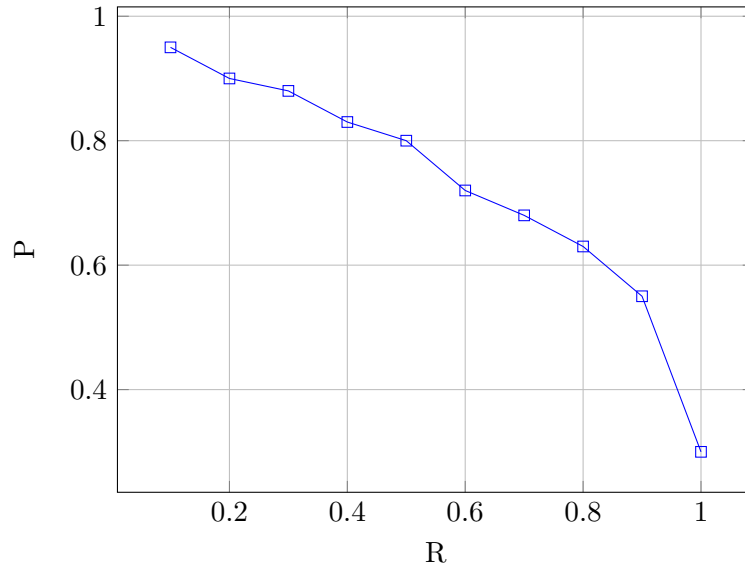
Preciznost (eng. *precision*) meri sposobnost modela da tačno identifikuje pozitivne instance, odnosno koliko je model precizan u svojim pozitivnim predikcijama. Visoka preciznost znači da kada model predvidi prisustvo objekta, ta predikcija je često tačna. Matematički, preciznost se izražava kao:

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

Odziv (eng. *recall*) je udeo pronađenih pozitivnih instanci od svih svih pozitivnih instanci. U kontekstu detekcije objekata, odziv meri koliko dobro model pronalazi sve prisutne objekte na slici. Matematički, preciznost se izražava kao:

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

Uglavnom je potrebno pronaći kompromis između preciznosti i odziva jer povećanje jednog često dovodi do smanjenja drugog.



Slika 4.2: Primer krive preciznosti i odziva

4.2 Napredne metrike za evaluaciju performansi

Kriva preciznosti i odziva (eng. *precision-recall curve*) predstavlja grafički prikaz koji ilustruje odnos između preciznosti i odziva modela pri različitim pragovima pouzdanosti. Ključna je za razumevanje balansa između tačnosti detekcija i sposobnosti pronalaženja svih relevantnih objekata. Konstruiše se tako što model generiše predviđanja sa različitim ocenama pouzdanosti. Za svaki prag pouzdanosti, detekcije iznad praga se smatraju pozitivnim i na osnovu njih se računaju preciznost i odziv. Nakon ovoga se tačke (preciznost i odziv) nanose na graf, formirajući krivu. Primer ove krive prikazan je na slici 4.2.

Prosečna preciznost (eng. *average precision - AP*) predstavlja površinu ispod krive preciznosti i odziva. Ona pruža jedinstvenu numeričku vrednost koja sumira informacije sadržane na krivoj tj. kvantifikuje sposobnost modela da tačno detektuje objekte kroz različite pragove pouzdanosti. Matematički se aproksimira kao:

$$AP = \sum_{i=2}^n (Rec_i - Rec_{i-1}) * Prec_i$$

gde su Rec_i i $Prec_i$ odziv i preciznost na i -tom pragu pouzdanosti, a n ukupan broj pragova. Vrednosti AP kreću se od 0 do 1, gde više vrednosti ukazuju na bolje performanse modela. Ova metrika omogućava poređenje različitih modela za detekciju objekata. Postoji više varijacija AP koje se koriste. Neke od njih su:

- AP@.50: AP izračunat pri IoU pragu od 0.5,
- AP@.75: AP izračunat pri IoU pragu od 0.75,

- AP@[.5:.95]: prosek AP vrednosti za IoU pragove od 0.5 do 0.95 sa korakom od 0.05.

Srednja prosečna preciznost (eng. *mean average precision* - *mAP*) je metrika koja se koristi za evaluaciju modela u scenarijima sa više klasa. Ona pruža jedinstvenu meru performansi modela preko svih klasa objekata. Predstavlja aritmetičku sredinu AP vrednosti izračunatih za svaku klasu objekata. Izačunava se kao:

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^N \text{AP}_i$$

gde N predstavlja ukupan broj klasa, a AP_i AP za i -tu klasu. Vrednosti mAP se takođe kreću između 0 i 1, a više vrednosti ukazuju na bolje ukupne performanse modela preko svih klasa. Analogno AP, postoji više varijacija mAP koje se koriste:

- mAP@.50: srednja vrednost AP@.50 za sve klase,
- mAP@.75: srednja vrednost AP@.75 za sve klase,
- mAP@[.5:.95]: srednja vrednost AP@[.5:.95] za sve klase.

mAP ima široku primenu u takmičenjima i referentim testovima (eng. *Benchmark*) za detekciju objekata i segmentaciju instanci poput COCO [23].

4.3 Specifične metrike za segmentaciju

Mask IoU je metrika za evaluaciju tačnosti segmentacije instanci, koja proširuje koncept IoU sa graničnih okvira na precizne maske segmentacije. Izačunava se kao odnos površine preseka predviđene maske segmentacije i stvarne maske za dati objekat, gde je površina preseka broj piksela označenih kao deo objekta u obe maske, a površina uni je ukupan broj piksela označenih kao deo objekta u bilo kojoj od ove dve maske (bez duplikata).

Za razliku od IoU nad graničnim okvirima, IoU nad maskama meri precizno poklapanje na nivou piksela i osetljivija je na male razlike u segmentaciji.

Glava 5

Priprema podataka

Priprema podataka predstavlja ključni korak u razvoju modela. Kvalitet i reprezentativnost pripremljenih podataka direktno utiču na performanse i generalizaciju modela. Ovaj proces obuhvata niz koraka, od prikupljanja sirovih podataka do njihove transformacije u format pogodan za treniranje Mask R-CNN i YOLOv8 modela.

Cilj ove faze je kreiranje visokokvalitetnog skupa podataka koji će omogućiti efikasno treniranje modela sposobnog da precizno detektuje i segmentuje ključne regione na samouslužnim kasama u realnim uslovima.

U ovoj glavi detaljno su opisane sve faze pripreme podataka.

5.1 Prikupljanje podataka

Prikupljanje podataka za ovaj rad bilo je dugotrajno i obuhvata širok spektar samouslužnih kasa u različitim maloprodajnim objektima širom sveta.

Izvori podataka

Snimci su prikupljeni sa kamera na samouslužnim kasama. Pozicionirane su direktno iznad kupca i snimaju u jednoj od dve rezolucije: 1920×1080 (eng. *full HD*) ili 1280×720 (eng. *HD ready*), generišući 30 slika svake sekunde.

Obuhvaćeno je 9 različitih maloprodajnih lanaca (u daljem tekstu referisanih kao klijenti). U okviru tih 9 klijenata, podaci su prikupljeni iz 165 prodajnih objekata, odnosno 1265 samouslužnih kasa. Prodavnice su locirane širom sveta pružajući raznolikost u tipovima klasa.

Podaci su prikupljeni u periodu od 3 godine, od septembra 2020. do septembra 2023. obezbeđujući sezonske varijacije.

Raznolikost skupa podataka proizilazi iz raznovrsnosti okruženja različitih radnji. Tako se u podacima nalaze kase sa kružnom kao i kase sa četvrtastom zonom za odlaganje

```
import av
import random

def ekstrakcija_slike(putanja_do_videa, putanja_izlazne_slike):
    # Otvaranje video fajla
    video_fajl = av.open(putanja_do_videa)

    # Dohvatanje toka video podataka
    video_tok = video_fajl.streams.video[0]

    # Trajanje snimka u sekundama
    trajanje_videa = video_tok.duration * video_tok.time_base

    # Biranje nasumičnog trenutka unutar snimka
    nasumicno_vreme = random.uniform(0, trajanje_videa)

    # Premotavanje do izabranog trenutka
    video_fajl.seek(int(nasumicno_vreme * av.time_base))

    # Dekodiranje slike
    slika = next(video_fajl.decode(video_tok))

    # Cuvanje slike
    slika.to_image().save(putanja_izlazne_slike)

    # Zatvaranje video fajla
    video_fajl.close()
```

Slika 5.1: Fukncija za ekstrakciju slika iz video fajla

proizvoda. Neke kase imaju kесе, dok druge nemaju. Osvetljenje i uglovi kamere variraju od kase do kase, pružajući raznolikost u uslovima snimanja.

Metode prikupljanja

Tokom svake transakcije na samouslužnoj kasi, paralelno se pravi video zapis. Iz svakog video zapisa ekstrahuje se jedna slika u nasumičnom trenutku. Video zapisi su u DVR video formatu, a ekstrahovane slike u JPG formatu. Kod za ekstrakciju slika iz video zapisa prikazan je na 5.1.

Pravna usklađenost

Korišćeni podaci su strogo poverljivi i nisu prikazani u radu. Imena klijenata su cenzurisana i označena kao „klijent 1”, „klijent 2” itd. Za potrebe demonstracije korišćeni su isključivo podaci iz demonstracionog okruženja unutar kancelarija kompanije *Everseen*.

5.2 Anotiranje podataka

Anotiranje podataka je proces označavanja sirovih podataka, u ovom slučaju slika, kako bi se identifikovali i precizno definisali objekti ili regioni od interesa. U kontekstu ovog rada, to podrazumeva precizno označavanje granica različitih zona definisanih u 1.2.

Proces anotiranja

Anotirane su četiri ključne klase, svaka predstavljena poligonom za preciznu segmentaciju. Alat korišćen za anotiranje je interni alat razvijen u okviru kompanije *Everseen*, sa funkcionalnostima sličnim alatu CVAT (eng. *Computer Vision Annotation Tool*) [24].

Anotiranje je odrađeno interno, sa zaposlenima kompanije *Everseen*, osiguravajući visok nivo kvaliteta anotacija.

Kontrola kvaliteta

Proces kontrole kvaliteta bio je dvostepen:

1. Tokom procesa anotacije vršena je selekcija slika. Odstranjene su mutne slike, kao i slike sa lošim vidnim poljem zbog neadekvatno postavljenih kamera.
2. Nakon završetka anotiranja, drugi anotatori su pregledali završene anotacije, osiguravajući konzistentnost i tačnost.

Ovaj pristup je omogućio identifikaciju i korekciju potencijalnih grešaka, kao i održavanje visokog kvaliteta anotacija.

Format anotacija

Anotacije su prvenstveno generisane u internom formatu kompanije *Everseen*. Za potrebe treninga YOLOv8 i Mask R-CNN modela, anotacije su konvertovane u COCO (eng. *common objects in context*) format [23].

Izazovi u procesu anotiranja

Glavni izazovi u procesu anotiranja uključivali su:

- identifikaciju i uklanjanje slika lošeg kvaliteta,
- osiguravanje konzistentnosti u anotacijama između različitih anotatora i,
- precizno definisanje granica zona, posebno u slučajevima gde su granice manje očigledne.

Navedeni problemi razrešeni su kroz proces kontrole kvaliteta i kontinualnu komunikaciju sa anotorima.

5.3 Podela podataka

U radu primenjen je pristup podele podataka baziran na klijentima, što predstavlja realističniji scenario primene modela u praksi. Ovakav pristup obezbeđuje da model bude evaluiran na potpuno nezavisnim podacima, simulirajući realan scenario primene na novim, prethodno nevidenim klijentima. Na tabeli 5.1 prikazana je podela broja slika po klijentima.

Klijent	Broj slika
Klijent 1	2,066
Klijent 2	304
Klijent 3	266
Klijent 4	14,766
Klijent 5	494
Klijent 6	1,829
Klijent 7	206
Klijent 8	3,909
Klijent 9	4,034

Tabela 5.1: Pregled broja slika po klijentu

Ukupan broj slika je 27,847. Podaci su podeljeni na trening i test skup. Testni skup čine podaci od klijenta 8. Ovaj izbor napravljen je strateški, s obzirom na to da klijent 8 sadrži značajnu raznovrsnost u izgledu kasa. Ova raznolikost omogućava procenu sposobnosti modela da generalizuje na različitim konfiguracijama samouslužnih kasa. U okviru podataka za trening, podaci klijenta 6 izdvojeni su kao validacioni skup (ovaj klijent izabran je nasumično), dok su ostali podaci korišćeni za treniranje modela.

Osiguravanje reprezentativnosti

Uključivanjem podataka od različitih klijenata, osigurava se da model uči iz raznovrsnih postavki samouslužnih kasa. Broj slika po klijentu odražava njegovu relevantnost i veličinu unutar *Everseen*-ovog ekosistema. Ovo osigurava da model bude prilagođen realnoj raspodeli podataka u praksi.

Izdvajanjem celokupnog klijenta za testni skup, osigurava se stroga evaluacija generalizacije modela na nove, nevidene postavke kasa.

Glava 6

Razvoj modela i rezultati

U ovoj glavi predstavljeni su razvoj i evaluacija modela, kao i vizualizacija njihove primene.

6.1 Konfiguracija modela

Za svaki od korišćenih modela, testirano je nekoliko varijanti kako bi se bolje uporedile njihove performanse.

Svi modeli su inicijalizovani sa težinama prethodno istreniranim na COCO skupu podataka. Ulazne slike skalirane su na veličinu 640×640 .

Konfiguracije YOLOv8

Trenirane su 3 varijante YOLOv8 modela:

- YOLOv8n (*nano*): varijanta sa najmanjim brojem parametara, optimizovana za brzo izvršavanje.
- YOLOv8m (*medium*): srednja varijanta koja balansira između brzine i preciznosti.
- YOLOv8x (*extra large*): najveća varijanta, dizajnirana za maksimalnu preciznost.

Konfiguracije Mask R-CNN

Trenirane su 2 varijante Mask R-CNN modela:

- Mask R-CNN R50: varijanta sa manjim brojem parametara koja koristi ResNet50 kao osnovnu mrežu.
- Mask R-CNN R101: varijanta sa većim brojem parametara koja koristi ResNet101 kao osnovnu mrežu.

6.2 Funkcija greške YOLOv8 modela

Funkcija greške YOLOv8 segmentacionog modela je dizajnirana kako bi obuhvatila sve aspekte zadatka detekcije objekata i segmentacije instanci. Ova funkcija se sastoji od nekoliko komponenti, od kojih svaka adresira specifičan aspekt problema.

Funkcija binarne unakrsne entropije (eng. *binary cross-entropy* - *BCE*) koristi se za klasifikaciju objekata unutar graničnih okvira. Računa se sledećom formulom:

$$L_{cls} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (6.1)$$

gde je N ukupan broj klasa, y_i stvarna vrednost za i -tu klasu (0 ili 1), a p_i predviđena vrednost i -te klase (vrednost između 0 i 1).

Funkcija raspodele fokalne greske (eng. *distribution focal loss* - *DFL*) tretira problem regresije graničnih okvira kao predviđanja raspodele verovatnoća, umesto direktnog predviđanja koordinata. Ovo omogućava modelu da izrazi nesigurnost u svojim predikcijama i da preciznije lokalizuje objekte. Formula je prikazna u [25], a u ovom radu označena je kao L_{DFL} .

Sledeća komponenta je funkcija greške za regresiju graničnih okvira, CIoU (eng. *complete IoU*) koja predstavlja unapređenje IoU funkcije greške uzimajući u obzir nekoliko aspekata preklapanja graničnih okvira. Matematička formulacija je:

$$L_{CIoU} = 1 - \text{IoU} + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} + \alpha v$$

gde je IoU već uveden presek nad unijom, ρ euklidsko rastojanje između centra predviđenog i stvarnog graničnog okvira, c dijagonala najmanjeg okvira koji obuhvata oba okvira, α težinski faktor, a v mera konzistentnosti odnosa širine i visine između predviđenog i stvarnog okvira.

Segmentaciona komponenta greške je BCE primenjena nad pikselima maski. Formula je ista kao 6.1, s tim da je N ukupan broj piksela u maski, y_i stvarna vrednost i -tog piksela, a p_i predviđena verovatnoća da piksel pripada objektu. Ova formula označena je kao L_{seg} .

Konačna formula funkcije greške YOLOv8 segmentacionog modela je:

$$L_{YOLOv8seg} = w_1 * L_{cls} + w_2 * L_{DFL} + w_3 * L_{CIoU} + w_4 * L_{seg}$$

gde su w_1 , w_2 , w_3 i w_4 težinski hiperparametri.

6.3 Funkcija greške Mask R-CNN modela

Klasifikaciona i segmentaciona komponenta su iste kao i kod YOLOv8 modela. Izbor funkcije za regresiju graničnih okvira je drugačiji i glasi:

$$L_{bbox} = \sum_i \text{SmoothL1}(t_i - t_i^{gt})$$

gde je t_i predviđena i -ta koordinata graničnog okvira, a t_i^{gt} stvarna i -ta koordinata graničnog okvira. Funkcija SmoothL1 predstavlja modifikaciju L1 funkcije greške, koja kombinuje karakteristike L1 i L2 funkcija greške.

$$\text{SmoothL1}(x) = \begin{cases} 0.5 * x^2 & \text{ako } |x| < 1, \\ |x| - 0.5 & \text{u suprotnom} \end{cases}$$

Ukupna funkcija greške Mask R-CNN modela izražena je kao:

$$L_{Mask\ R-CNN} = v_1 * L_{cls} + v_2 * L_{bbox} + v_3 * L_{seg}$$

gde su v_1 , v_2 , i v_3 težinski hiperparametri.

6.4 Hardverska i softverska infrastruktura

Za trening i evaluaciju modela korišćen je sistem opremljen sa dve *nVidia RTX 5000* grafičke kartice, dva *Intel Xeon E5-2630* procesora i 64 gigabajta radne memorije.

Softversko okruženje bazira se na *CUDA 12* platformi. Specifično za treniranje YOLOv8 modela, korišćen je *Ultralytics* paket koji pruža optimizovanu implementaciju i alate za trening YOLOv8 arhitekture [26]. Za treniranje Mask R-CNN modela, primenjen je *Detectron2* paket, koji nudi visoko prilagodljivo okruženje za treniranje [27].

6.5 Evaluacija i rezultati

U okviru ovog istraživanja, sprovedena je serija eksperimenata s ciljem optimizacije performansi treniranih modela. Fokus je bio na ispitivanju uticaja različitih hiperparametara i konfiguracija na efikasnost modela u zadatku segmentacije regiona na samouslužnim kasama.

Dužina treninga varirana je od 10 do 150 epoha, što je omogućilo detaljnu analizu konvergencije modela i potencijalnog preprilagođavanja. Kod YOLO modela, posebna pažnja posvećena je uticaju segmentacione greške na ukupne performanse. U tu svrhu, izvršen je eksperiment sa povećanjem koeficijenta uz L_{seg} , s ciljem da se model optimizuje za precizniju segmentaciju. Međutim, ova modifikacija nije rezultirala poboljšanjem

performansi. Varijacije stope učenja (eng. *learning rate*) takođe nisu pokazale značajan uticaj na konačne performanse modela. S druge strane, povećanje dimenzija ulazne slike dovelo je do primetnog poboljšanja performansi, ali uz očekivano povećanje vremena treniranja i izvršavanja modela. Veličina grupe (eng. *batch size*) prilagođena je memorijskim ograničenjima korišćenih grafičkih kartica.

Uprkos ekstenzivnom eksperimentisanju sa različitim hiperparametrima, većina modifikacija nije dovela do značajnih promena u performansama modela. Ovo sugeriše da su originalne konfiguracije modela bile blizu optimalnih za dati zadatak.

Na tabeli 6.1 prikazana je komparativna analiza najboljih rezultata za svaki od testiranih modela. Oznaka *B* označava da se metrika odnosi na granični okvir, dok oznaka *M* označava da se metrika odnosi na segmentacione maske.

Tabela 6.1: Rezultati merenja performansi modela

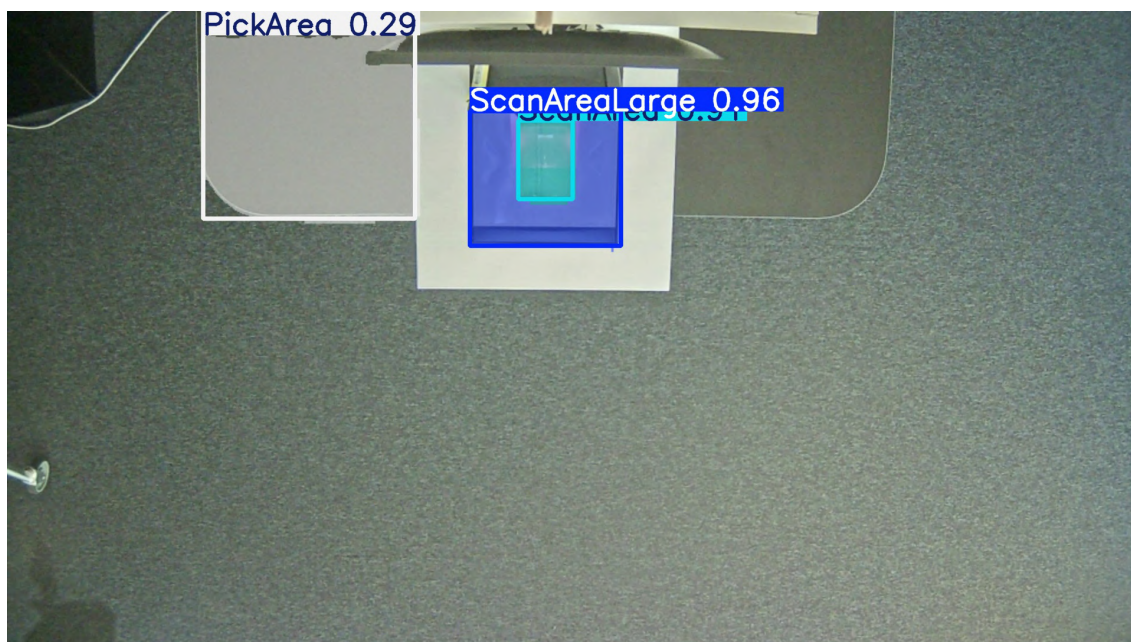
Model	mAP50 (B)	mAP50-95 (B)	mAP50 (M)	mAP50-95 (M)
YOLO <i>Nano</i>	0.9667	0.8634	0.9636	0.7736
YOLO <i>Medium</i>	0.9712	0.8852	0.9683	0.7845
YOLO <i>Extra</i>	0.9811	0.9056	0.9778	0.7995
Mask R-CNN R50	0.9215	0.7219	0.9081	0.7292
Mask R-CNN R101	0.9184	0.7421	0.9067	0.7517

6.6 Vizualizacija rezultata

U ovoj sekciji prikazani su rezultati najboljeg YOLOv8 modela na demonstrativnom okruženju kompanije *Everseen*. Važno je istaći da ovo okruženje ima određena ograničenja u reprezentaciji realnih scenarija, posebno u pogledu razlikovanja zona za preuzimanje i zona za odlaganje proizvoda. Naime, ove dve zone su vizuelno identične, što ne odražava stvarnu situaciju na većini samouslužnih kasa.

Na slici 6.1, prikazana je detekcija i segmentacija ključnih regija na praznoj kasi. Regija za odlaganje proizvoda nije detektovana zbog identičnog izgleda sa regijom za preuzimanje proizvoda. Na slici 6.2, je problem i dalje prisutan iako su proizvodi prisutni. Na ovoj slici se takođe može primetiti da je model u stanju da aproksimira ivicu zone iako je ona zaklonjena korpom.

Na slici 6.3, prikazan je rezultat predviđanja u okviru drugog demonstrativnog okruženja demonstrirajući robustnost modela na varijacije u osvetljenju. Primećujemo da je ovde model dosta siguran u predviđanju zone za preuzimanje proizvoda. To je najverovatnije posledica toga da se na njoj nalazi ručni skener.



Slika 6.1: Rezultat predviđanja na praznoj kasi



Slika 6.2: Rezultat predviđanja na kasi sa proizvodima



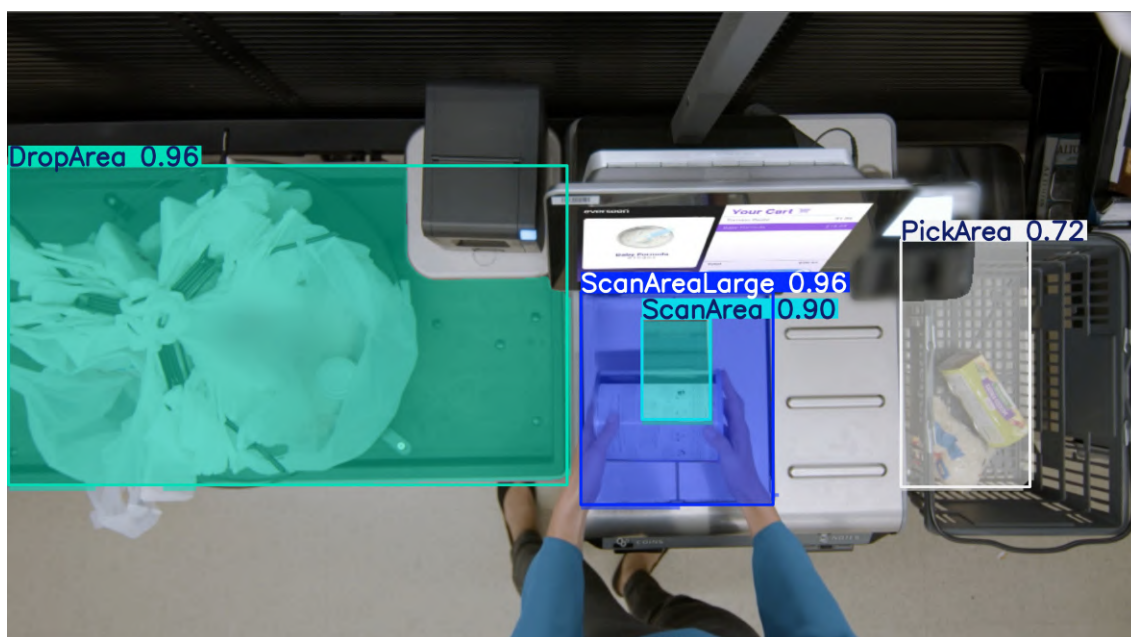
Slika 6.3: Rezultat predviđanja sa ručnim skenerom

Na slikama 6.4 i 6.5, prikazan je rezultat izvršavanja u realnom okruženju¹. Na slici 6.4, ne nalazi se ništa na zoni za preuzimanje proizvoda, dok je na slici 6.5 prisutna korpa na navedenoj zoni.



Slika 6.4: Slika iz realnog okruženja sa praznom regijom za preuzimanje proizvoda

¹Slike su preuzete sa: <https://everseen.com/>



Slika 6.5: Slika iz realnog okruženja sa korpom na zoni za preuzimanje proizvoda

6.7 Diskusija o ograničenjima i primeni

Važno je naglasiti da se ograničenja uočena na demonstrativnom okruženju ne odražavaju nužno na performanse modela u realnim scenarijima. U stvarnim scenarijima, gde postoje jasnije vizualne razlike između zona za preuzimanje i odlaganje proizvoda, model pokazuje značajno bolju sposobnost razlikovanja ovih regiona.

Neka od ograničenja demonstrativnog okruženja su:

- Nedostatak vizuelne razlike između zona za preuzimanje i odlaganje proizvoda.
- Ograničen raspon varijacija u dizajnu samouslužnih kasa.
- Potencijalno pojednostavljeni uslovi osvetljenja i okruženja.

Glava 7

Zaključak

U ovom radu predstavljeno je istraživanje primene modela dubokog učenja za identifikaciju i segmentaciju funkcionalnih regija na samouslužnim kasama. Fokus istraživanja bio je na dva ključna modela koji predstavljaju različite pristupe: Mask R-CNN i YOLO. Mask R-CNN, zasnovan na principu dvostepene detekcije, prvo generiše predloge od interesa, a zatim vrši klasifikaciju i segmentaciju tih regija. Sa druge strane, model YOLOv8, oslanja se na princip jednostepene detekcije, simultano vršeći lokalizaciju i klasifikaciju objekata na celoj slici u jednom prolazu kroz mrežu.

Vremenska razlika u nastanku ova dva modela odražava se na njihove performanse i arhitekturne karakteristike. Mask R-CNN, predstavljen 2017. godine, bio je revolucionaran u svoje vreme, uvodeći mogućnost istovremene detekcije objekata i segmentacije instanci sa visokom preciznošću. Međutim, YOLOv8, objavljen 2023. godine, ima prednost višegodišnjeg napretka u oblasti dubokog učenja. Ova evolucija omogućila je modelu YOLOv8 da koristi naprednije tehnike novijih arhitektura osnovnih mreža i sofisticiranijih funkcija greške. Kao rezultat, YOLOv8 generalno proizvodi bolje rezultate.

Dobijeni rezultati sugerišu da se razvijeni modeli mogu efikasno primeniti u sistemima za automatizaciju i optimizaciju procesa na samouslužnim kasama. Precizna segmentacija funkcionalnih regija otvara mogućnost za napredne analize ponašanja kupaca, detekciju neregularnosti u korišćenoj kasi i potencijalno unapređenje korisničkog iskustva.

Jedno od ključnih područja za potencijalno unapređenje performansi modela leži u strateškom proširenju skupa podataka za trening. Prikupljanje veće količine raznovrsnih podataka može značajno doprineti robusnosti i generalizaciji modela. Posebno je važno fokusirati se na identifikaciju specifičnih scenarija i konfiguracija samouslužnih kasa gde model trenutno pokazuje slabije performanse, poput demonstrativnog okruženja gde su regije za preuzimanje i odlaganje identične. Ciljanim povećanjem broja slika koje pokazuju izazovne konfiguracije moguće je unaprediti sposobnost modela da precizno odredi ove funkcionalne regije.

Bibliografija

- [1] R. Wolfe, “More self-checkout is coming, no matter how much you hate it,” *Wall Street Journal*, 2022. Pristupljeno: 2024-06-14.
- [2] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, “Mask R-CNN,” *CoRR*, vol. abs/1703.06870, 2017.
- [3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.
- [4] M. Nikolić and A. Zečević, *Mašinsko Učenje*. MATF, 2019.
- [5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [6] F. S. D. W. Fanjie Meng, Xinqing Wang and X. Hua, “Energy-efficient gabor kernels in neural networks with genetic algorithm training method,” 2019.
- [7] Z. Zou, Z. Shi, Y. Guo, and J. Ye, “Object detection in 20 years: A survey,” *CoRR*, vol. abs/1905.05055, 2019.
- [8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 00, pp. 580–587, June 2014.
- [9] R. B. Girshick, “Fast R-CNN,” *CoRR*, vol. abs/1504.08083, 2015.
- [10] R. Shaoqing, H. Kaiming, G. Ross, and S. Jian, “Faster r-cnn: Towards real-time object detection with region proposal networks.,” in *NIPS* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 91–99, 2015.
- [11] D. A. Forsyth and J. Ponce, *Computer Vision - A Modern Approach, Second Edition*. Pitman, 2012.
- [12] S. Cakir, M. Gauß, K. Häppeler, Y. Ounaïjar, F. Heinle, and R. Marchthaler, “Semantic segmentation for autonomous driving: Model evaluation, dataset generation, perspective comparison, and real-time capability.,” *CoRR*, vol. abs/2207.12939, 2022.

- [13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation.," *CoRR*, vol. abs/1505.04597, 2015.
- [14] N. Murali, "Image classification vs semantic segmentation vs instance segmentation," 2021. Pristupljeno: 2024-06-14.
- [15] X. Cao, J.-S. Pan, Z. Wang, Z. Sun, A. ul Haq, W. Deng, and S. Yang, "Application of generated mask method based on mask r-cnn in classification and detection of melanoma," *Computer Methods and Programs in Biomedicine*, vol. 207, p. 106174, 2021.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.
- [17] A. Rastogi, "Resnet50," 2021. Pristupljeno: 2024-06-14.
- [18] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," 2016.
- [19] Q. Wen, Z. Luo, R. Chen, Y. Yang, and G. Li, "Deep learning approaches on defect detection in high resolution aerial images of insulators," *Sensors*, vol. 21, p. 1033, 02 2021.
- [20] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "Cspnet: A new backbone that can enhance learning capability of cnn.," in *CVPR Workshops*, pp. 1571–1580, Computer Vision Foundation / IEEE, 2020.
- [21] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation.," *CoRR*, vol. abs/1803.01534, 2018.
- [22] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation.," *CoRR*, vol. abs/1904.02689, 2019.
- [23] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014.
- [24] CVAT.ai Corporation, "Computer vision annotation tool (cvat)." Dostupno na: <https://github.com/cvat-ai/cvat>.
- [25] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection.," in *NeurIPS* (H. Larochelle, M. Ranzato, R. Hadsell, M.-F. Balcan, and H.-T. Lin, eds.), 2020.

BIBLIOGRAFIJA

- [26] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics yolov8,” 2023.
- [27] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, “Detectron2,” 2019.

Biografija autora

Lazar Jovanović, rođen 4. septembra 1996. u Beogradu, završio je Gimnaziju „Sveti Sava” u Beogradu. Nakon toga, upisuje Matematički fakultet Univerziteta u Beogradu na smeru Informatika 2015. godine i završava ga 2019. godine sa prosekom 8.50, čime stiče zvanje Diplomirani informatičar. Profesionalnu karijeru započinje 2018. godine u kompaniji *RT-RK* gde radi na razvoju algoritama neuronskih mreža u okviru *embedded* sistema. Godine 2022. prelazi u kompaniju *Everseen* na poziciju softverski inženjer, a 2024. godine postaje lider inženjerskog tima za mašinsko učenje. Neizostavno je napomenuti da u trenutku pisanja ovog rada poseduje plavi pojas iz brazilske džiu-džice.