SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Diplomski projekt

DETEKCIJA POLICA NA SLIKAMA

Antun Herkov

Zagreb, siječanj 2023.

Sažetak

Ovaj rad se bavi problemom detekcija objekata - specifično detekcijom polica na slikama iz supermarketa. Kroz rad je dan opis problema te pregled arhitekture, metoda i procesa učenja modela. Prikazani su rezultati dobiveni kroz nekoliko provedenih eksperimenata te postavljena pitanja na koji način bi bilo moguće ostvariti bolje rezultate.

Sadržaj

1. Uvod

Detekcija objekata na slikama je tehnika računalnog vida korištena za lociranje instanci objekata na slikama ili videima (koji su zapravo kolekcija većega broja slika). Algoritmi detekcije objekata generalno koriste tehnike strojnoga ili dubokog učenja kako bi postigli dobre rezultate. Kada ljudske oči gledaju slike ili videosnimku, potrebno je samo par trenutaka kako bi prepoznale i locirale bitne objekte. To je upravo i cilj problema detekcije objekata - pokušati implementirati ovu vrstu inteligencije koristeći računalo.

Glavni je element naprednih sustava za pomoć u vožnji koji omogućuju automobilima da prepoznaju vozne ceste, pješake i sl. kako bi povećali sigurnost u prometu.

Slika 1: Primjer korištenja algoritma detekcije objekata

Detekcija objekata može se izvesti na nekoliko različitih načina. Popularni pristup je korištenje dubokoga učenja koristeći konvolucijske neuronske mreže (CNN-ove) kao što su R-CNN i YOLOv5. Takve mreže automatski uče detektirati objekte na slikama.

Postoje dva načina korištenja ovakvoga pristupa. Prvi podrazumijeva stvaranje i treniranje prilagođenoga modela detekcije. Kako bi naučili prilagođeni detektor "od nule", potrebno je modelirati arhitekturu mreže koja će učiti značajke objekata koji su nam od interesa. Također je potrebno sastaviti poveći skup označenih podataka za učenje CNN-a. Ovaj način može uroditi odličnim rezultatima, međutim potrebno je ručno postavljati slojeve i težine u mreži, što iziskuje puno vremena i podataka za učenje.

Drugi način je korištenje već naučenog detektora objekata. Većina paradigmi detekcije objekata koje je zasnovano na dubokom učenju iskorištavaju moć prijenosnog učenja (engl. *transfer learning*) koja omogućava da se proces učenja nastavi od nekog već naučenog stanja mreže koje je još moguće dodatno prilagoditi vlastitim zahtjevima. Ovakav način je znatno brži jer su detektori već naučeni na tisućama, ako ne i milijunima slika.

U mrežama poput YOLO-a, CNN stvara mrežna predviđanja za područja diljem cijele slike koja se dekodiraju kako bi se stvorili konačni granični okviri (engl. *bounding box*) za objekte.

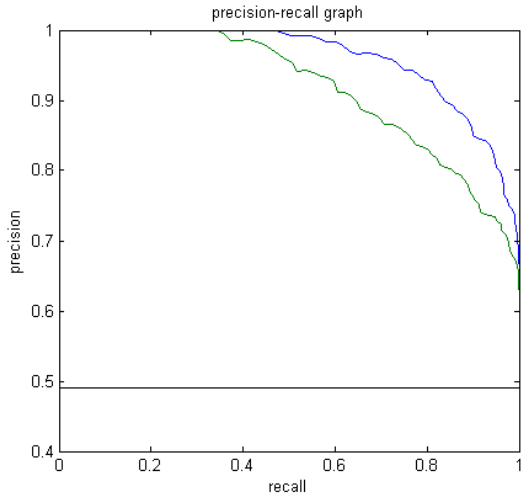
2. You Only Look Once (YOLO) modeli

Za razliku od ostalih pristupa algoritama za detekciju objekata koji su prenamijenili klasifikatora za detekciju, YOLO predlaže korištenje *end-to-end* neuronske mreže koja daje predviđanja graničnih okvira te vjerojatnosti klasa odjednom. Ovakvim pristupom YOLO postiže vrhunske rezultate detekcije objekata u stvarnom vremenu naspram ostalih algoritama.

Međutim, niti jedan algoritam nije savršen. Detekcija YOLO modelima je otežana manjim objektima koji su na slikama grupirani, a modeli se fokusiraju na detekciju jednog objekta. Takvi scenariji su problematični za detektirati i lokalizirati.

YOLO modele također karakterizira niža točnost u usporedbi sa znatno sporijim algoritmima detekcije objekta kao što je R-CNN gdje se vidi kompromis između točnosti predviđanja te dobivanja rezultata u stvarnome vremenu.

Bitna metrika u YOLO modelima je prosječna preciznost (engl. *Average Precision* (AP)) koja se računa kao površina ispod preciznost/odziv krivulje za jednu klasu - prosjek po svim klasama je srednja prosječna preciznost (engl. *mean Average Precision* (mAP)). Preciznost se odnosi na omjer ispravno predviđenih pozitiva te ukupnog broja predviđanja modela dok se metrika odziva računa kao omjer ukupnog broja predviđanja modela za jednu klasu i ukupnog broja postojećih oznaka za tu klasu.



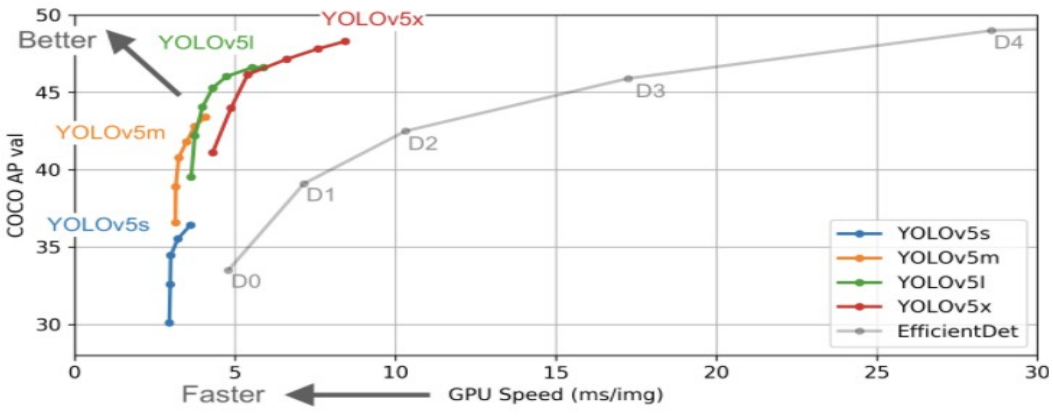
Slika 2: Primjer grafa preciznosti i odziva

Bitno je razumjeti da se u problemu detekcije preciznost i odziv ne odnose na sama predviđanja klase već na predviđanja graničnih okvira za računanje granice odluke pripadnosti okviru.

2.1. YOLOv5

YOLOv5, kao i većina ostalih modela detekcije objekata, radi izlučivanje bitnih značajki iz ulaznih slika koje su onda ulaz u sustav predviđanja za crtanje graničnih okvira te vjerojatnosti pripadnosti određenoj klasi.

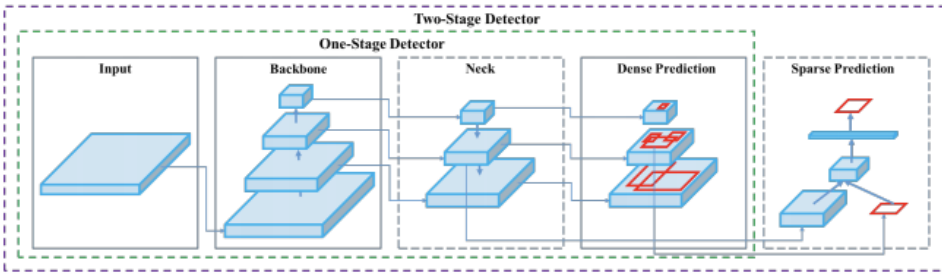
Dolazi u nekoliko glavnih varijanti: *small* (s), *medium* (m), *large* (l) i *extra large* (x). Svaka varijanta nudi različitu točnost pod cijenu različitog vremena potrebnog za učenje modela, ali bitno je napomenuti da čak i najveća varijanta modela, YOLOv5x, može postići sličnu razinu točnosti kao neki drugi detektori (u ovom primjeru EfficientDet) u manje vremena.



Slika 3: Graf performansi varijanti YOLO modela

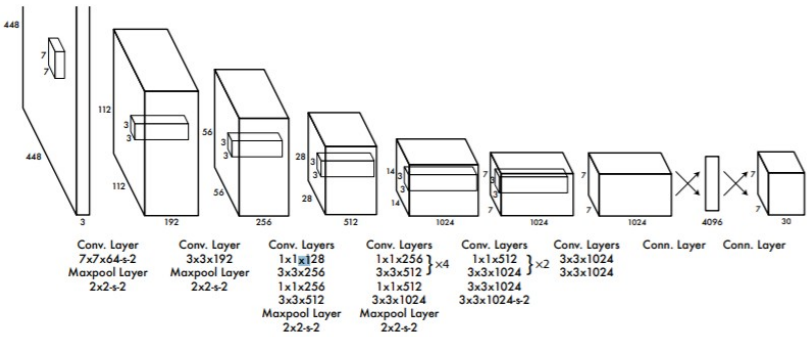
2.1.1. Arhitektura

YOLO mreža se sastoji od tri glavna dijela. *Backbone* - konvolucijska neuronska mreža koja agregira i stvara značajke slika različitih razina granularnosti. *Neck* - niz slojeva za spajanje značajki koje se prosljeđuju predviđanju. *Head* - koristi proslijeđene značajke te radi korake za dobivanje graničnih okvira i vjerojatnosti pripadnosti klasama.



Slika 4: Primjer generalne arhitekture detektora

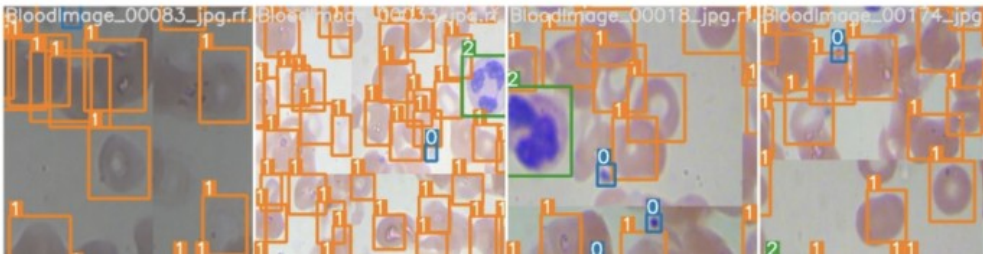
Arhitektura YOLO mreže ima ukupno dvadeset i četiri konvolucijska sloja s dva potpuno povezana sloja na kraju mreže.



Slika 5: Arhitektura YOLOv5 mreže

2.1.2. Priprema i proces učenja

Dvije glavne procedure koje YOLOv5 koristi tijekom procesa učenja su augmentacija podataka te izračuni gubitka. Augmentacija podataka podrazumijeva nasumične transformacije nad originalnim skupom podataka za učenje koje izlažu model širom rasponu semantičkih razlika u podacima.



Slika 6: Primjer augmentiranih podataka

3. Skup podataka

Kao skup podataka u eksperimentima korištena je prilagođena verzija Retail50K skupa podataka slika polica u supermarketima. Originalni Retail50K je kolekcija od 47 tisuća slika iz različitih supermarketa. Anotacije na slikama su rubovi polica, hladnjača i ekrana.

U sklopu ovog projekta fokus eksperimenata je bio na problemu neorijentiranih graničnih okvira pa su u skladu s time iz skupa za učenje maknuti podaci koji bi mogli predstavljati problem u procesu učenja.

Prilagodba skupa podataka je rađena tako da se svaka slika na kojoj postoji jedna ili više anotacija pod kutom koji nije u strogo određenoj granici prihvatljivih kutova makne iz skupa podataka. Pošto su anotacije Retail50K-a prilagođene poligonima, a ne pravokutnicima, također su maknute sve slike čije anotacije nisu odgovarale problemu definiranim ovim radom.


Slika 7: Primjer odbijene i prihvaćene slike

3.1. YOLOv5 format anotacija

YOLOv5 očekuje specifičan format anotacija. Svaka slika treba imati svoj par u obliku tekstualne datoteke (.txt) gdje svaka linija sadrži opis jednoga graničnog okvira.

```
0 0.49908867478370667 0.027524137869477272 0.9967171549797058 0.019219372421503067
0 0.4990845322608948 0.3371339738368988 0.9966669678688049 0.03205119073390961
0 0.4990317225456238 0.5685399770736694 0.9965845346450806 0.04103429988026619
0 0.49917617440223694 0.803165853023529 0.9965887069702148 0.04744872450828552
```

Slika 8: Primjer tekstualne datoteke anotacija

Na primjeru postoje četiri označena objekta (u ovom problemu postoji samo jedna klasa, polica, označena nulom). Svaki redak predstavlja jednu označenu policu.

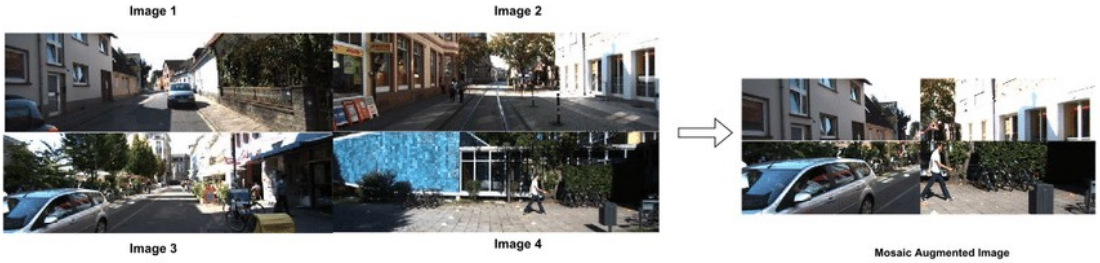
Sadržaj svakog retka je formata: *klasa centar_x centar_y širina visina*. Numeričke vrijednosti moraju biti normalizirane dimenzijama ulazne slike (između 0 i 1). Oznake klasa počinju od nultoga indeksa.

3.2. Augmentacija podataka

YOLOv5 svaki batch u procesu učenja šalje kroz učitavač podataka koji nad ulaznim podacima vrši augmentaciju. Može se definirati nekoliko vrsta augmentacija koje YOLO vrši kao što su skaliranje, prilagodbe prostora boja, mozaička augmentacija (engl. *mosaic augmentation*) koja spaja četiri slike u četiri "pločice" u nasumičnome omjeru te *mixup,* transformacija zamišljena za probleme klasifikacije, ali se pokazalo da je primjenjiva na problem detekcije objekata.

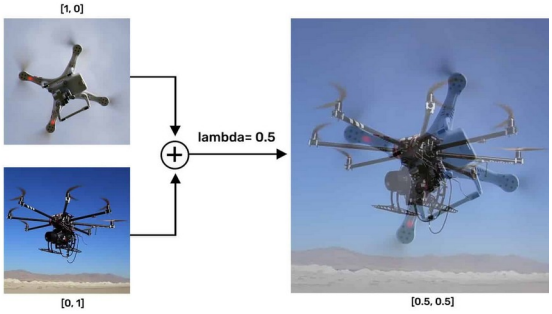
3.2.1. Mozaička augmentacija

Postupak uzima četiri ulazne slike koje kombinira u jednu na način da simulira četiri nasumična izrezivanja slike (ali nastoji održati relativnu skalu objekta naspram ostatka slike), kombinira klase tako da postoje slike s varijantnim pojavljivanjem klasa na njima te varira broj objekata koji se pojavljuju na slikama.



3.2.2. Mixup

Mixup u suštini radi prosjek između dvije slike s obzirom na dani parametar težine. Pošto se problem detekcije objekata u našem slučaju svodi na traženje graničnih okvira, kombiniramo i oznake za obje slike u jednu što rezultira kombiniranom slikom na kojoj su sve oznake graničnih okvira i prve i druge slike.



Slika 10: Primjer mixup augmentacije

4. Eksperimenti i rezultati

Eksperimenti provedeni u sklopu ovoga projekta zamišljeni su kao usporedba postojećih varijanti YOLO modela te utjecaja vremena učenja na iste kada imaju pristup dijelu odnosno ukupnom skupu podataka za učenje, ispitivanje i validaciju modela.

U nastavku je dan pregled skupa nekoliko bitnijih hiperparametara s kojim je pokrenuto učenje svakoga eksperimenta. Kao optimizator u procesu učenja korišten je stohastički gradijentni spust (SGD) s momentumom od 0.937. Inicijalna stopa učenja postavljena je na vrijednost od 0.01.

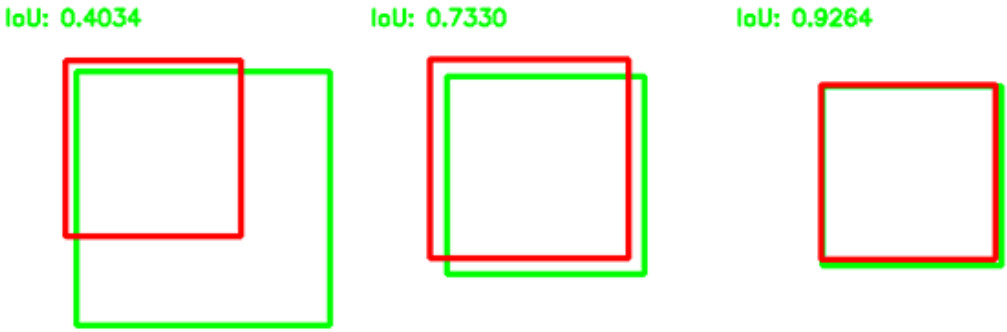
U tablici 1 vidljivi su hiperparametri korišteni za augmentaciju podataka tijekom učenja modela. HSV hiperparametri utječju na *hue*, *saturation* i *value* augmentacije u prostoru boja. *Degrees*, *translate* i *scale* su hiperparametri za transformaciju rotacije, translacije te skaliranja. *Mosaic* i *mixup* hiperparametri nam govore kolika je vjerojatnost korištenja mozaičke odnosno *mixup* augmentacije za svaki *batch* podataka.

Tablica 1: Pregled hiperparametara augmentacije

hsv_h	0.015
hsv_s	0.7
hsv_v	0.4
degrees	0.0
translate	0.1
scale	0.9
mosaic	1.0
mixup	0.1

Također je dan pregled ostvarenih rezultata svih eksperimenata u Tablici 2, a detaljniji pregled rezultata za svaki eksperiment dan je u nastavku u posebnim odjeljcima.

Kao dvije glavne metrike korištene su preciznost te uprosječena srednja preciznost (mAP) nad različitim omjerima površine presjeka i unije (IoU). To nam pomaže u evaluaciji rezultata jer vidimo koliko se predviđeni granični okviri preklapaju sa stvarnim oznakama istih.



Slika 11: Različiti omjeri površina presjeka i unije graničnog okvira

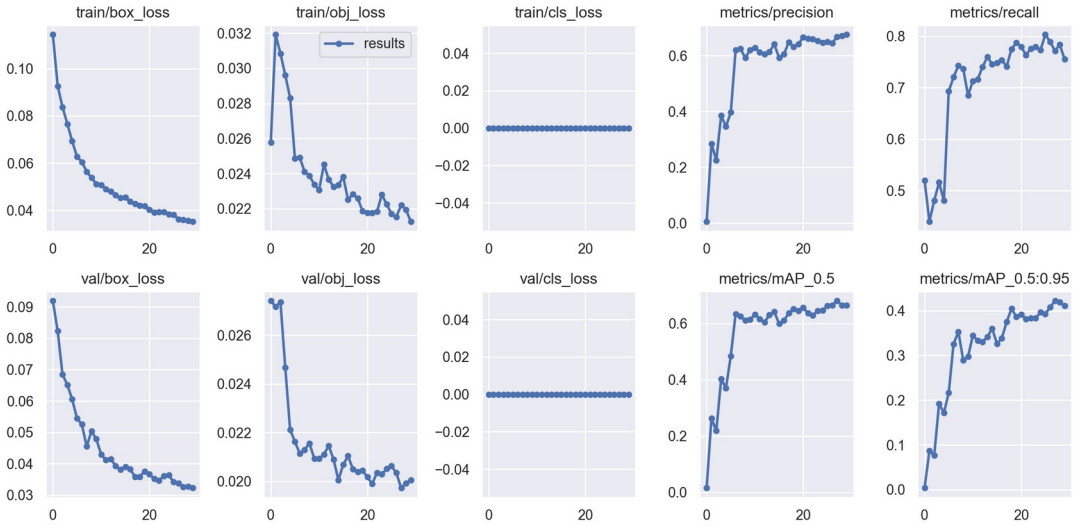
Tablica 2: Pregleda rezultata eksperimenata

	Preciznost	**mAP[0.5:0.95]**
S - 30 epoha	0.67579	0.41096
S - 50 epoha	0.69078	0.44459
S - 100 epoha	0.69317	0.46630
S - 30 epoha, svi podaci	0.67241	0.45916
M - 30 epoha	0.68366	0.43452
M - 50 epoha	0.70071	0.45004
M - 100 epoha	0.71033	0.48212
M - 30 epoha, svi podaci	0.69943	0.48974
L - 30 epoha	0.72916	0.46250
L - 50 epoha	0.70997	0.46387
L - 100 epoha	0.72711	0.49432

4.1. YOLOv5s

4.1.1. 30 epoha (podskup skupa podataka)

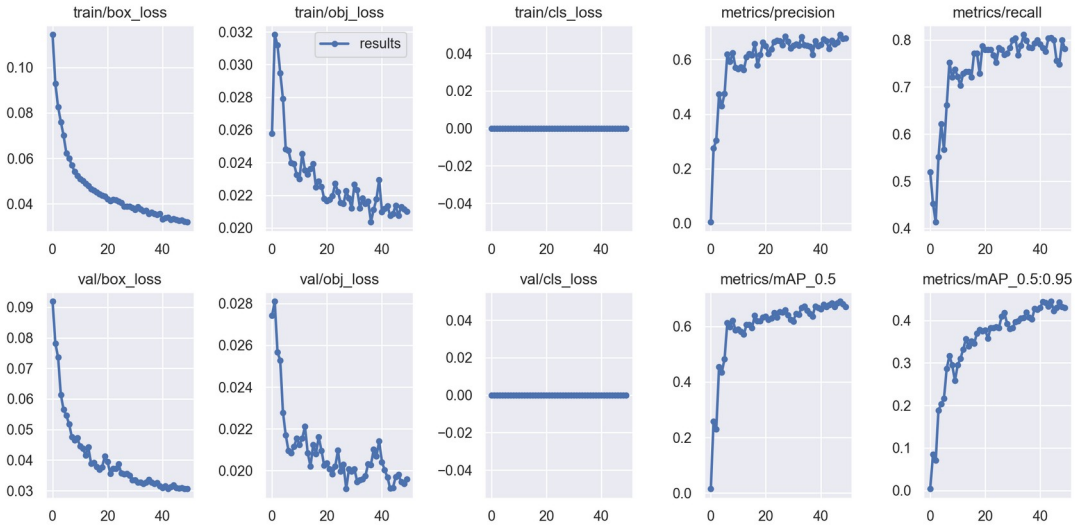
Uz korištenje SGD optimizatora s momentumom 0.937 te inicijalnom stopom učenja 0.01, eksperiment od 30 epoha pokrenut nad dijelom podataka dosegao je preciznost od 0.67579 te mAP nad različitim pragovima IoU-a u rangu [0.5, 0.95] u vrijednosti od 0.42186.



Slika 12: Pregled metrika modela

4.1.2. 50 epoha (podskup skupa podataka)

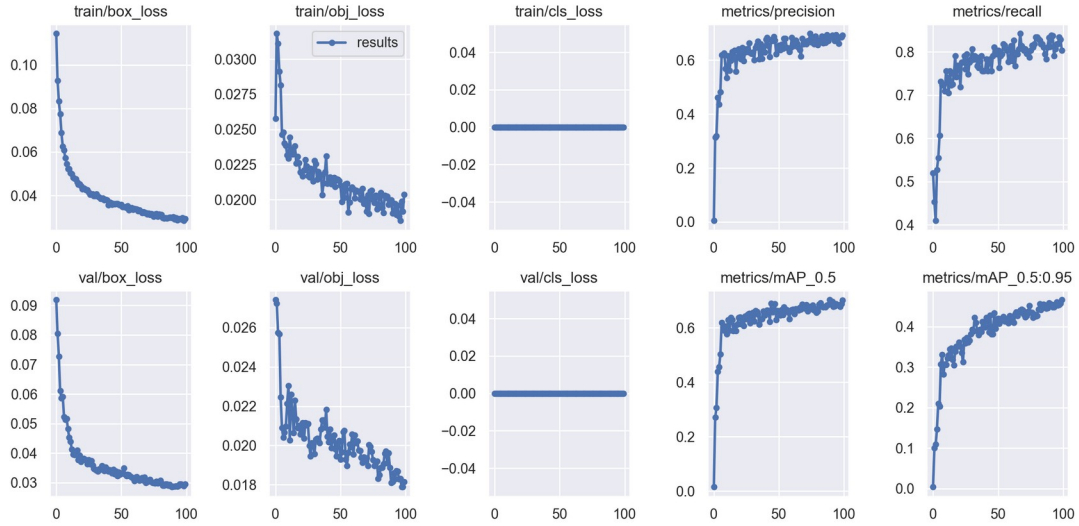
Uz korištenje SGD optimizatora s momentumom 0.937 te inicijalnom stopom učenja 0.01, eksperiment od 50 epoha pokrenut nad dijelom podataka dosegao je preciznost od 0.69078 te mAP nad različitim pragovima IoU-a u rangu [0.5, 0.95] u vrijednosti od 0.44459.



Slika 13: Pregled metrika modela

4.1.3. 100 epoha (podskup skupa podataka)

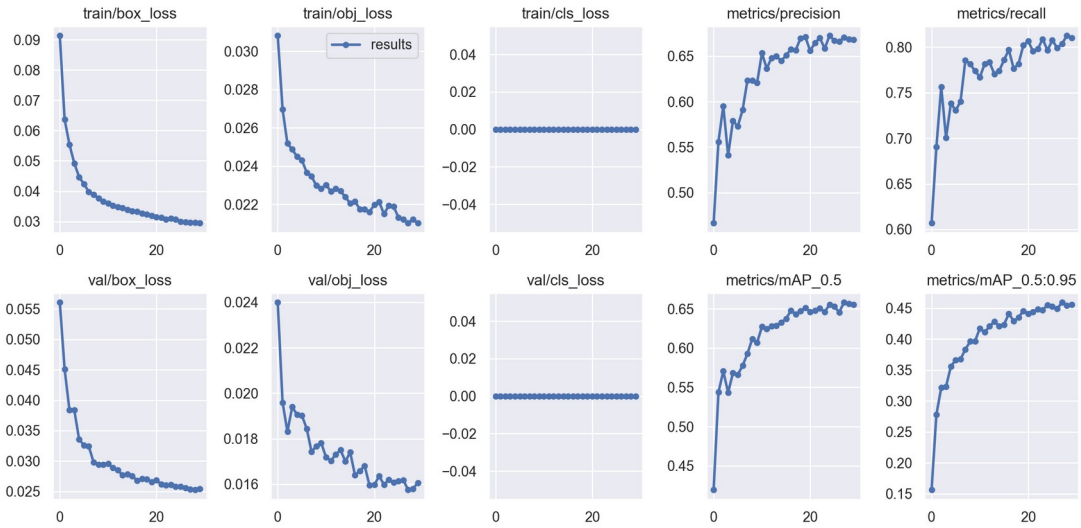
Uz korištenje SGD optimizatora s momentumom 0.937 te inicijalnom stopom učenja 0.01, eksperiment od 100 epoha pokrenut nad dijelom podataka dosegao je preciznost od 0.69317 te mAP nad različitim pragovima IoU-a u rangu [0.5, 0.95] u vrijednosti od 0.4663.



Slika 14: Pregled metrika modela

4.1.4. 30 epoha (potpuni skup podataka)

Uz korištenje SGD optimizatora s momentumom 0.937 te inicijalnom stopom učenja 0.01, eksperiment od 30 epoha pokrenut nad potpunim skupom podataka dosegao je preciznost od 0.67241 te mAP nad različitim pragovima IoU-a u rangu [0.5, 0.95] u vrijednosti od 0.45916.

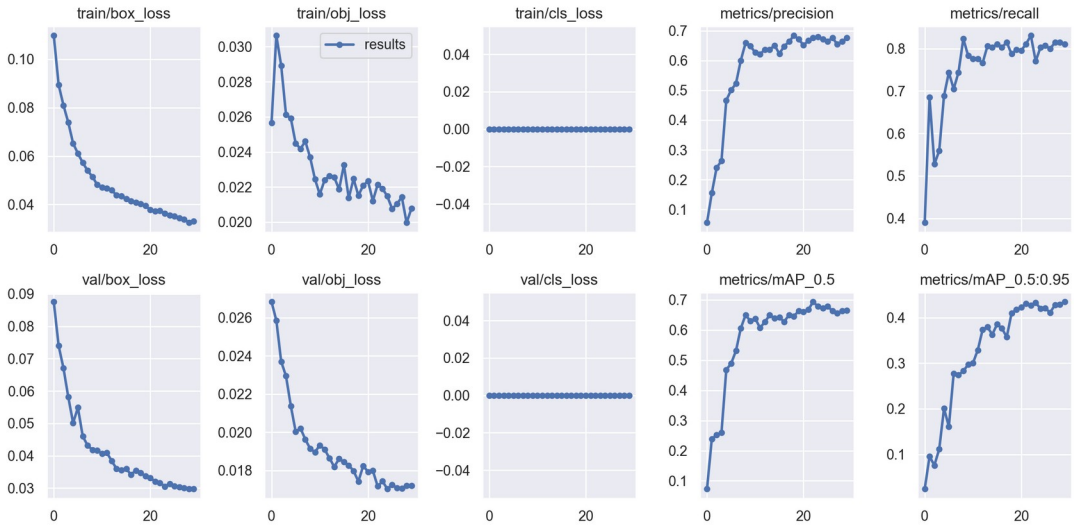


Slika 15: Pregled metrika modela

4.2. YOLOv5m

4.2.1. 30 epoha (podskup skupa podataka)

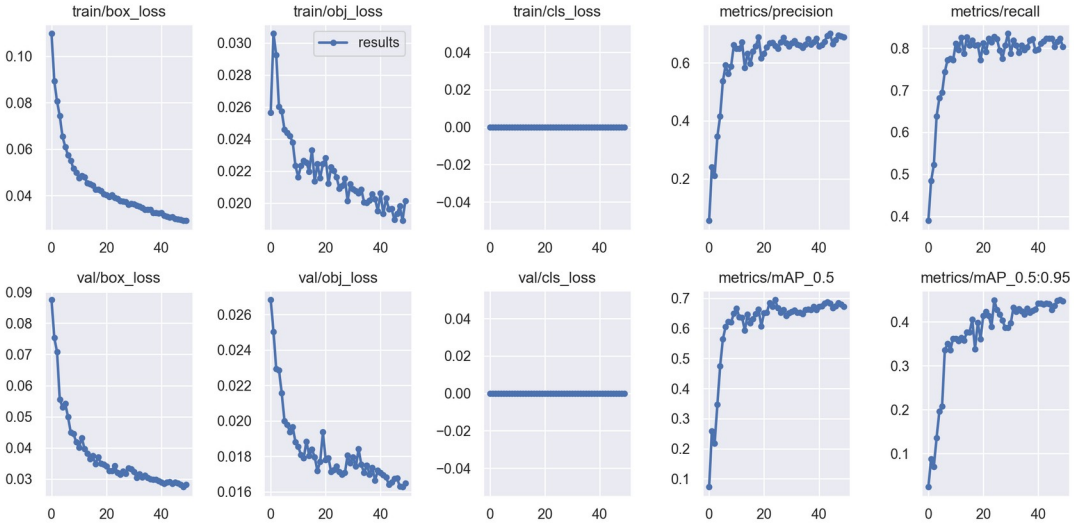
Uz korištenje SGD optimizatora s momentumom 0.937 te inicijalnom stopom učenja 0.01, eksperiment od 30 epoha pokrenut nad dijelom podataka dosegao je preciznost od 0.68366 te mAP nad različitim pragovima IoU-a u rangu [0.5, 0.95] u vrijednosti od 0.43452.



Slika 16: Pregled metrika modela

4.2.2. 50 epoha (podskup skupa podataka)

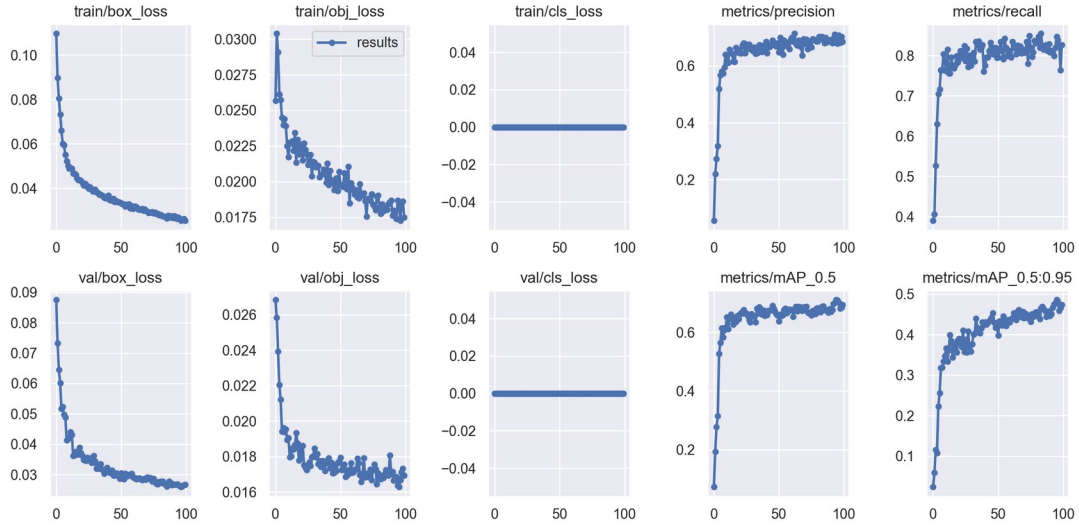
Uz korištenje SGD optimizatora s momentumom 0.937 te inicijalnom stopom učenja 0.01, eksperiment od 50 epoha pokrenut nad dijelom podataka dosegao je preciznost od 0.70071 te mAP nad različitim pragovima IoU-a u rangu [0.5, 0.95] u vrijednosti od 0.45004.



Slika 17: Pregled metrika modela

4.2.3. 100 epoha (podskup skupa podataka)

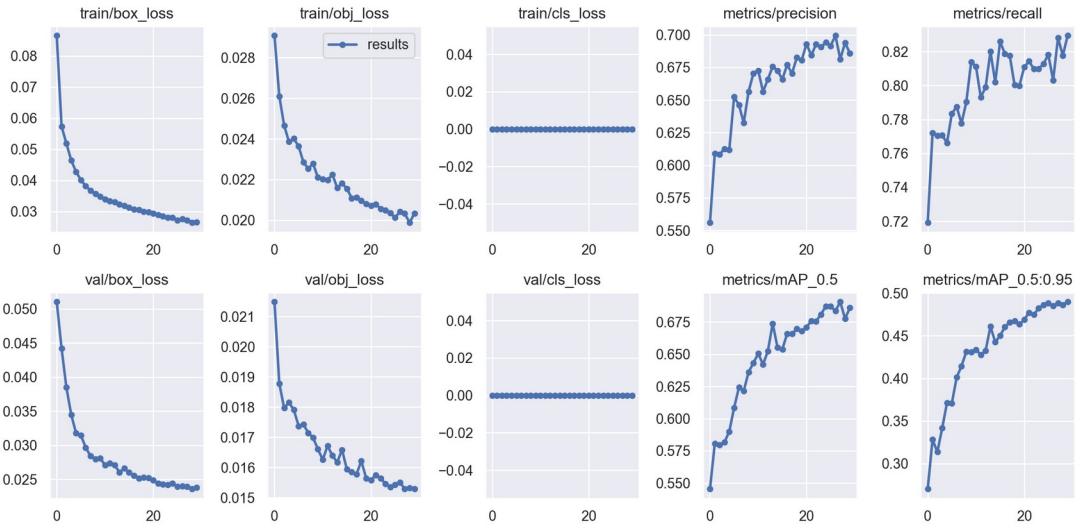
Uz korištenje SGD optimizatora s momentumom 0.937 te inicijalnom stopom učenja 0.01, eksperiment od 100 epoha pokrenut nad dijelom podataka dosegao je preciznost od 0.71033 te mAP nad različitim pragovima IoU-a u rangu [0.5, 0.95] u vrijednosti od 0.48212.



Slika 18: Pregled metrika modela

4.2.4. 30 epoha (potpuni skup podataka)

Uz korištenje SGD optimizatora s momentumom 0.937 te inicijalnom stopom učenja 0.01, eksperiment od 30 epoha pokrenut nad potpunim skupom podataka dosegao je preciznost od 0.69943 te mAP nad različitim pragovima IoU-a u rangu [0.5, 0.95] u vrijednosti od 0.48974.

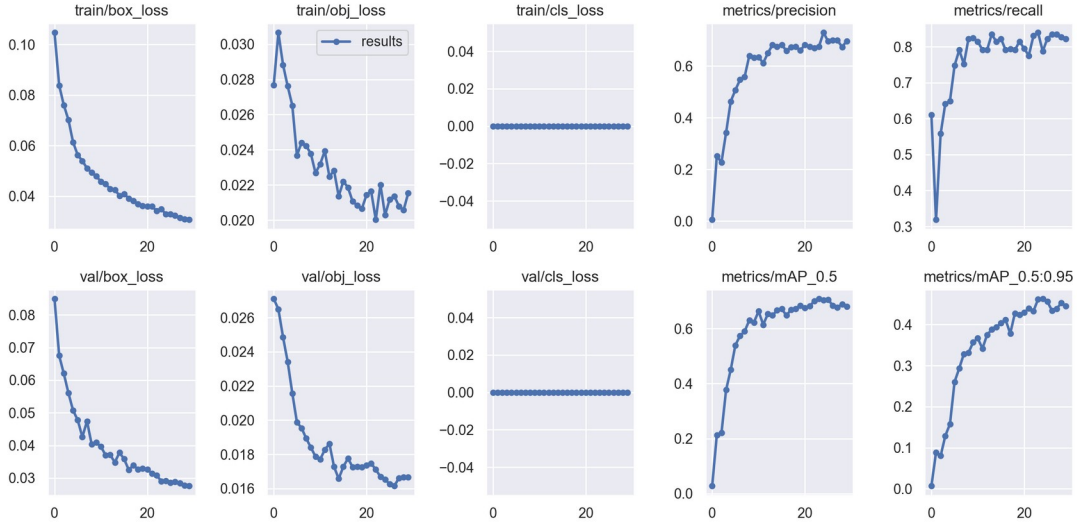


Slika 19: Pregled metrika modela

4.3. YOLOv5l

4.3.1. 30 epoha (podskup skupa podataka)

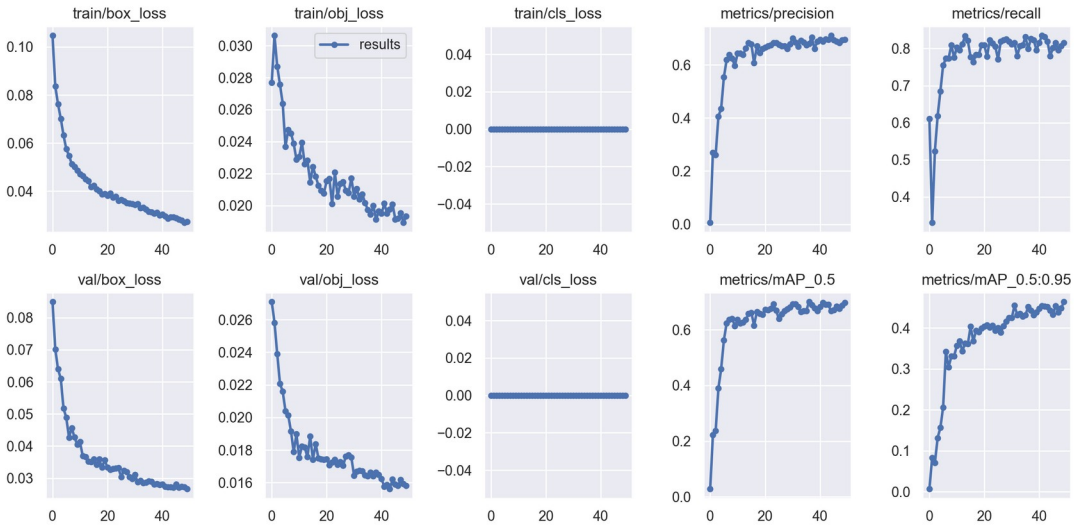
Uz korištenje SGD optimizatora s momentumom 0.937 te inicijalnom stopom učenja 0.01, eksperiment od 30 epoha pokrenut nad dijelom podataka dosegao je preciznost od 0.72916 te mAP nad različitim pragovima IoU-a u rangu [0.5, 0.95] u vrijednosti od 0.4625.



Slika 20: Pregled metrika modela

4.3.2. 50 epoha (podskup skupa podataka)

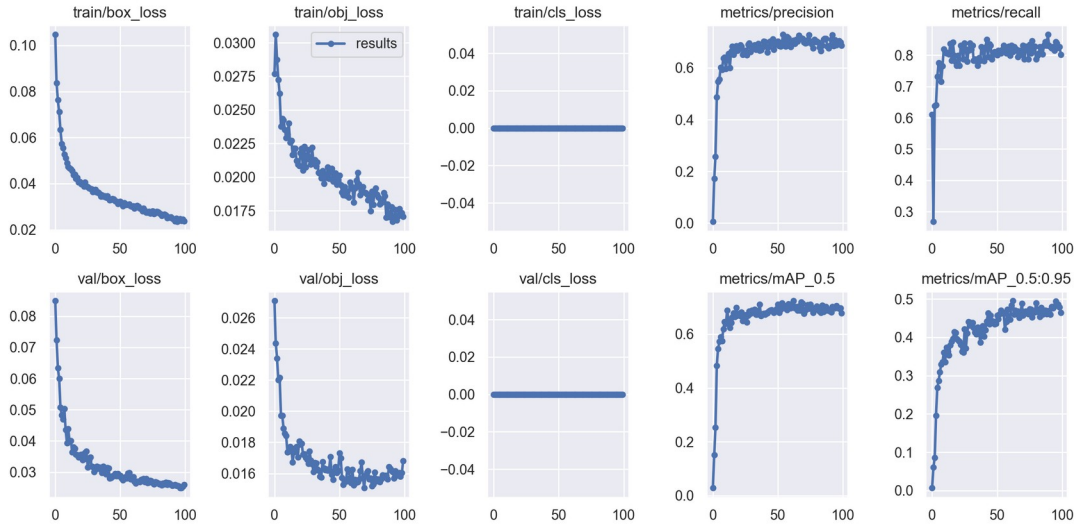
Uz korištenje SGD optimizatora s momentumom 0.937 te inicijalnom stopom učenja 0.01, eksperiment od 50 epoha pokrenut nad dijelom podataka dosegao je preciznost od 0.70997 te mAP nad različitim pragovima IoU-a u rangu [0.5, 0.95] u vrijednosti od 0.46387.



Slika 21: Pregled metrika modela

4.3.3. 100 epoha (podskup skupa podataka)

Uz korištenje SGD optimizatora s momentumom 0.937 te inicijalnom stopom učenja 0.01, eksperiment od 100 epoha pokrenut nad dijelom podataka dosegao je preciznost od 0.72711 te mAP nad različitim pragovima IoU-a u rangu [0.5, 0.95] u vrijednosti od 0.49432.



Slika 22: Pregled metrika modela

4.3.4. Usporedba rezultata

Uspoređujući rezultate između S, M i L eksperimenata od 100 epoha vidljiv je napredak u mogućnostima modela što se tiče metrika postignutih rezultata svakoga modela. YOLOv5l kao varijanta YOLO modela s najvećim mogućnostima, ali i time najvećom kompleksnošću te vremenom potrebnom za učenjem modela je postignuo najbolje rezultate u usporedbi s manjima varijantama modela nad istima podacima, YOLOv5m te YOLOv5s.

Tablica 3: Usporedba varijanti modela

	Preciznost	**mAP[0.5:0.95]**
S - 100 epoha	0.69317	0.46630
M - 100 epoha	0.71033	0.48212
L - 100 epoha	0.72711	0.49432

5. Zaključak

Kroz provedene eksperimente i njihove rezultate da se zaključiti kako modeli imaju lošije performanse na lošije osvjetljenim slikama s nejasno definiranim policama te jako uskim policama. Nažalost, zbog limitacija resursa nisu provedeni svi zamišljeni eksperimenti te bi u daljnjemu radu valjalo provjeriti performanse većih varijanti YOLO modela na potpunom skupu podataka. Unatoč tome, rezultati su zadovoljavajući s prosječnom preciznošću od oko 0.7.

Kako ovo nije nužno problem koji je potrebno riješavati u stvarnome vremenu, u daljnjemu radu ne bi bilo na odmet usporediti performanse "sporijih" modela za problem detekcije objekata s YOLO modelima te usporediti njihove postignute rezultate i metrike.

6. Literatura

1. Kathuria, A: How to Train YOLO v5 on a Custom Dataset, s Interneta, https://blog.paperspace.com/train-yolov5-custom-data/, 2020.
2. Rehman, A: Converting a custom dataset from COCO format to YOLO format, s Interneta, https://medium.com/red-buffer/converting-a-custom-dataset-from-coco-format-to-yolo-format-6d98a4fd43fc, 2022.
3. Bandyopadhyay, H: YOLO: Real-Time Object Detection, s Interneta, https://www.v7labs.com/blog/yolo-object-detection, 2023.
4. Gutta, S: Object Detection Algorithm — YOLO v5 Architecture, s Interneta, https://medium.com/analytics-vidhya/object-detection-algorithm-yolo-v5-architecture-89e0a35472ef, 2021.
5. Mathworks: What Is Object Detection?, s Interneta, https://www.mathworks.com/discovery/object-detection.html, 2019.
6. Renukasoni: Image detection, recognition and image classification with machine learning, s Interneta, https://medium.com/ai-techsystems/image-detection-recognition-and-image-classification-with-machine-learning-92226ea5f595, 2019.
7. Yang, C et al.: PIoU Loss: Towards Accurate Oriented Object Detection in Complex Environments, ECCV, 2020.
8. Girschick, R et al.: Rich feature hierarchies for accurate object detection and semantic segmentation, Berkeley, 2013.
9. Redmon, J et al.: YOLOv3: An Incremental Improvement, 2018.