

# Prepoznavanje objekata primjenom dubokog učenja

---

**Baće, Tara**

**Undergraduate thesis / Završni rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:168:212650>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-03-21**



*Repository / Repozitorij:*

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1515

**PREPOZNAVANJE OBJEKATA PRIMJENOM DUBOKOG  
UČENJA**

Tara Baće

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1515

**PREPOZNAVANJE OBJEKATA PRIMJENOM DUBOKOG  
UČENJA**

Tara Baće

Zagreb, lipanj 2024.

## ZAVRŠNI ZADATAK br. 1515

Pristupnica: **Tara Baće (0036538042)**  
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo  
Modul: Računarstvo  
Mentor: izv. prof. dr. sc. Goran Delač

Zadatak: **Prepoznavanje objekata primjenom dubokog učenja**

### Opis zadatka:

Istražiti i opisati osnovne modele dubokog učenja za prepoznavanje objekata na slikama s posebnim naglaskom na model YOLO. Odabrati premjereni skup podataka koji sadrži slike na kojima se nalaze određene klase objekata. Programski ostvariti i opisati pokazni primjer rada sustava na odabranom skupu podataka. Navesti korištenu literaturu i primijenu pomoć.

Rok za predaju rada: 14. lipnja 2024.

*Zahvaljujem se mentoru izv.prof.dr.sc Goranu Delaču*

# Sadržaj

<b>1. Uvod</b>	<b>3</b>
<b>2. Detekcija objekata u slici</b>	<b>4</b>
2.1. Definicija i podjela	4
<b>3. Detektori u jednoj fazi</b>	<b>5</b>
3.1. Model Yolo	5
3.1.1. Ideja algoritma	6
<b>4. Programsko ostvarenje</b>	<b>7</b>
4.1. Generiranje ulaznih podataka	7
4.1.1. Sastav ulaznog skupa podataka	9
4.2. Google Colab bilježnica	10
4.2.1. Treniranje	11
4.2.2. Vrednovanje	11
<b>5. Provođenje eksperimenata</b>	<b>13</b>
5.1. Vrednovanje rezultata	13
5.1.1. Matrica zabune	14
5.1.2. Preciznost	15
5.1.3. Odziv	17
5.1.4. Krivulja preciznosti i odziva	18
5.1.5. F1-mjera	19
5.1.6. Srednja prosječna preciznost	20
5.2. Eksperimenti nad YOLOv5 modelu	20
5.2.1. Utjecaj broja epoha treninga	20

5.2.2. Utjecaj veličine ulaznog skupa podataka . . . . .	23
5.2.3. Diskusija rezultata . . . . .	27
<b>6. Zaključak . . . . .</b>	<b>28</b>
<b>Literatura . . . . .</b>	<b>29</b>
<b>Sažetak . . . . .</b>	<b>31</b>
<b>Abstract . . . . .</b>	<b>32</b>

# 1. Uvod

U posljednjih nekoliko desetljeća, vrlo brz napredak u području umjetne inteligencije (AI) doveo je do značajnih inovacija u raznim domenama ljudskog djelovanja. Jedno od najizazovnijih i najintragantnijih područja unutar AI-a je računalni vid (CV), disciplina koja se bavi razvojem algoritama i tehnika za analizu, interpretaciju i razumijevanje vizualnih informacija.

Detekcija objekata, jedna od ključnih grana računalnog vida, igra važnu ulogu u mnogim aplikacijama, od sigurnosnih sustava i automobilske industrije do medicinske dijagnostike i robotike. Cilj detekcije objekata je identificirati prisutnost i lokalizirati objekte unutar slika ili videozapisa, što omogućuje računalima da razumiju i interagiraju s okolinom na sličan način kao i ljudi.

U tom kontekstu, YOLO (*You Only Look Once*)[3] model predstavlja jedan od najinovativnijih pristupa detekciji objekata. Razvijen kao odgovor na potrebu za brzim i preciznim algoritmima, YOLO modeli kombiniraju napredne tehnike dubokog učenja s visokom brzinom obrade slika. Njihova sposobnost da identificiraju objekte u stvarnom vremenu čini ih nezamjenjivim alatom za razvoj naprednih aplikacija u području računalnog vida.

U radu je analiziran jednoprolazni model za detekciju objekata YOLO. Rad započinje definicijom i podjelom područja detekcije objekata u slici. Slijedi opis detektora u jednoj fazi gdje je detaljnije objašnjen model YOLO. U eksperimentalnom dijelu rada opisan je postupak treniranja na vlastitom skupu podataka te vrednovanje i usporedba dobivenih rezultata.



## 2. Detekcija objekata u slici

### 2.1. Definicija i podjela

Detekcija objekata u slici predstavlja jedno od ključnih područja ravoja i istraživanja u svijetu računalnog vida. Ova tehnika omogućava računalima da se približe ljudima te na unos fotografije ili videozapisa znaju identificirati gdje i što se nalazi u slici. Priroda problema nalaže i smjer rješenja, a to je kako što prije i što točnije identificirati položaj objekta te dodatno klasificirati koji objekt je detektiran.

Analiza fotografije izazovan je zadatak u svijetu umjetne inteligencije zbog niza problema koji se prilikom analize javljaju. Neki od glavnih problema uključuju prezasićenost fotografije objektima, gdje je prvi problem točno odrediti gdje se i koliko objekata u slici nalazi, a tek onda kojoj klasi sami objekti pripadaju. Slijedi varijabilnost objekata, odnosno raznolikost objekata koji pripadaju istoj klasi. Pozadinska buka, osvjetljenje i razlučivost fotografije koje ponajviše utječu na sposobnos detekcije modela. [1]

Problemi u području su izazovni, no kontinuiranim razvojem hardvera i algoritama za detekciju razvijeno je i unaprijeđeno sve više modela koji se približavaju real-time detekciji objekata. Metode koje se pritom koriste dijele se u dva glavna tipa: detektori u dvije faze (engl.*two-stage detectors*) i detektori u jednoj fazi (engl.*one-stage detectors*). Algoritmi se razlikuju u pristupu procesu detekcije i imaju različite karakteristike brzine i točnosti. Ovisno o području primjene, jedan će performansama biti korisniji od drugog.

## 3. Detektori u jednoj fazi

Jednofazni algoritmi za detekciju objekata počivaju na ideji pojednostavljenja detekcije kako bi se jednim prolaskom što brže generirao prijedlog položaja i klase objekta. Primjeri algoritma su algoritmi iz porodice Yolo, SSD, SSP-net, MultiBox, DSSD i drugi.

Arhitektura ovakvih modela sastoji se od tri glavna dijela, a sama implementacija svakog ovisi o konkretnoj verziji modela. Prvi dio, kralježnica (*engl. backbone*), dio je mreže odgovoran za ekstrakciju značajki iz fotografije. Slijedi vrat (*engl. neck*), koji sakuplja značajke s različitih slojeva mreže. Konačno, glava (*engl. head*) koja je odgovorna za predviđanje okvira i klase objekta. [2]

### 3.1. Model Yolo

Yolo model, skraćeno od *You only look once* [3], svijetu se predstavlja 2016. godine na CVPR konferenciji člankom *You Only Look Once: Unified, Real-Time Object Detection* grupe autora predvođenih Josephom Redmonom. Bio je to tada revolucionarni pristup detekciji jer se po prvi put problemu detekcije pristupa kao regresijskom problemu. Za razliku od dvofaznih detektora, Yolo model koristi jednu neuronsku mrežu kojom istovremeno predviđa okvire objektima i njihovu pripadnost klasi.

Kao glavne prednosti modela, autori navode njegovu njegovu jednostavnost koja sa sobom povlači i veliku brzinu. Nadalje, model gleda sliku kao cjelinu čime uči o objektima u kontekstu promjenjive pozadine pa rijeđe griješi klasificirajući pozadinu kao objekt. Model dobro generalizira, odnosno, na temelju viđenog skupa primjeraka dobro predviđa klase novim primjercima. Optimizirajući brzinu prolaska javljaju se i određeni nedostaci. Neki od nedostataka su lokalizacija i raspoznavanje manjih objekata, kao i analiza fotografija s većim zagušenjima, lošijom oštrinom i slično.

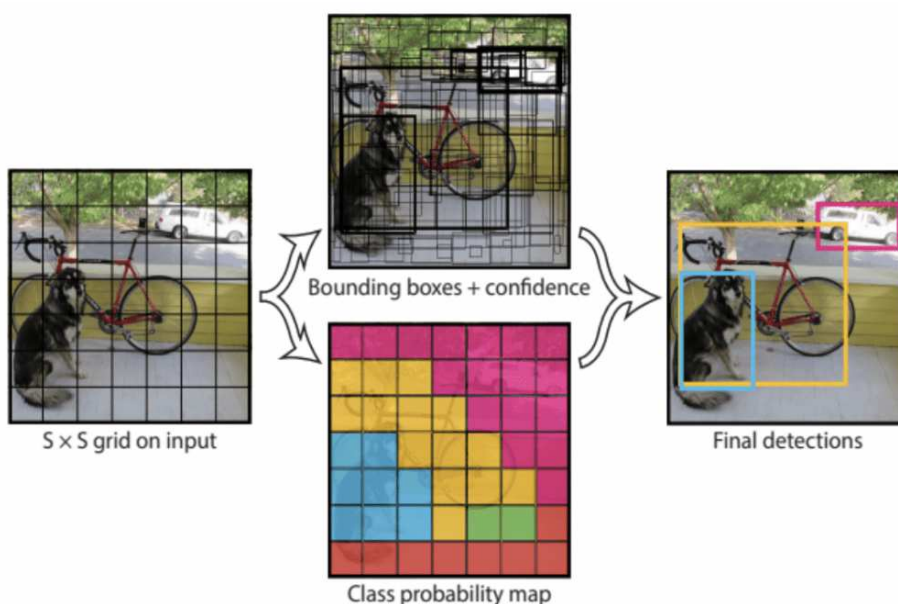
### 3.1.1. Ideja algoritma

Kako bi bolje shvatili građu modela, prvo treba pogledati općeniti algoritam rada sustava. Model na ulaz dobiva fotografiju koju zatim dijeli u mrežu dimenzija  $S \times S$ . Svaka ćelija koja sadržava centar objekta odgovorna je za klasifikaciju samog objekta. Zadaća ćelije je predvidjeti  $B$  definiranih okvira te za svaki iznijeti pouzdanost nalazi li se objekt unutar okvira te koliko točno ga okvir predviđa. Pouzdanost se računa prema formuli:

$$C = P_r(object) * IOU_{pred}^{truth} \quad (3.1)$$

gdje  $P_r(object)$  poprima vrijednost 1 ukoliko okvir sadržava objekt i 0 ako ga ne sadržava. Time je pouzdanost upravo proporcionalna omjeru presjeka i unije predviđenog okvira i stvarne lokacije objekta.

Svaki okvir sadržava pet predikcija:  $x, y, w, h$  te pouzdanost. Uređeni par  $(x, y)$  predstavlja središte okvira u odnosu na granice ćelije,  $w$  i  $h$  predstavljaju širinu i visinu okvira, a pouzdanost omjer presjeka i unije stvarne i predviđene granice objekta. Osim toga, svaka ćelija sadržava i  $C$  uvjetnih predikcija klasa  $P_r(class|object)$  koje govore o aposterijalnoj vjerojatnosti da dotični objekt pripada klasi  $C_i$ . Neovisno o broju okvira  $B$ , svaka će ćelija dati točno jedan set predikcija i to upravo za onaj okvir kojemu je mjera  $IOU$  najveća (dakle najbolje pokriva objekt koji previđa).



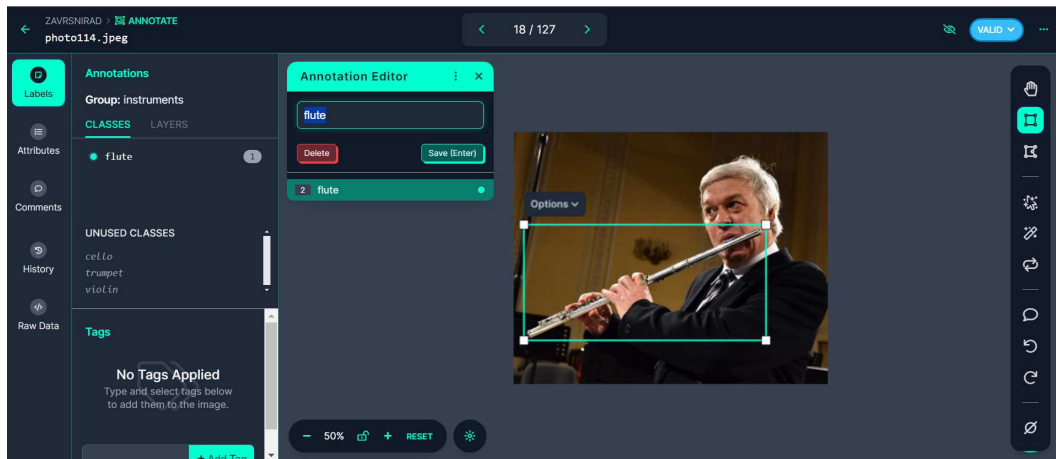
**Slika 3.1.** Princip rada Yolo modela [3]

## 4. Programsko ostvarenje

### 4.1. Generiranje ulaznih podataka

Cilj rada je prikazati performanse dotreniranog modela u ovisnosti o skupu ulaznih podataka. S tim ciljem, vlastiti set podataka je modeliran kako bi se osigurala kontrola nad kvalitetom i kvantitetom slika koje model koristi. Za ovaj zadatak odabran je alat Roboflow, koji putem web preglednika nudi jednostavno i pristupačno sučelje za upload i označavanje fotografija. Nakon obrade, fotografije je moguće izvesti u različitim formatima, kao što su JSON, XML, TXT, CSV i drugi, uz mogućnost specificiranja verzije modela za koji se podaci koriste. U svrhu ovog rada podatci su izvedeni u formatu *YOLO v5 PyTorch*.

Proces pripreme podataka sastoji se od nekoliko ključnih koraka[4]. Prvi korak je skupljanje podataka, što uključuje iscrpno pretraživanje interneta s ciljem stvaranja ravnomjernog skupa podataka, gdje su sve klase približno jednako zastupljene. Cilj je naučiti model da točno raspoznaje četiri klase: violinu, violončelo, flautu i trubu. Nakon prikupljanja i uvoza seta fotografija slijedi proces anotacije. Alat omogućuje jednostavno i učinkovito označavanje fotografija, pri čemu se na svakoj fotografiji identificiraju objekti i dodjeljuju im se odgovarajuće klase. Ovaj se proces ponavlja za svaku fotografiju u skupu podataka, osiguravajući preciznost i dosljednost anotacija.



**Slika 4.1.** Anotacija fotografije

Jednom kad imamo set označenih primjera, potrebno ih je zapakirati u novu verziju. Prilikom generiranja nove verzije, alat nudi nekoliko dodatnih koraka obrade. Odabire se takozvani "train/test split", odnosno postotak seta podataka koji će biti korišten prilikom treniranja, validiranja i testiranja modela.

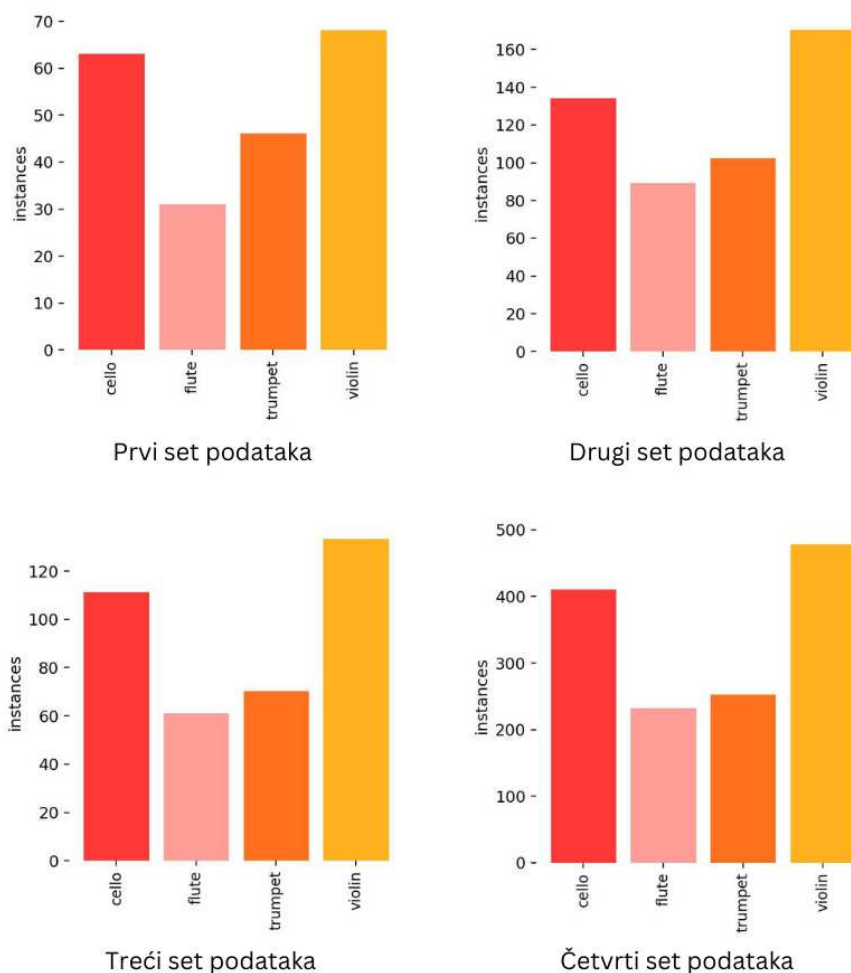
Slijedi korak odabira pretprocesiranja fotografija. U ovom koraku odabrano je automatsko orijentiranje fotografija i promjena dimenzija fotografije na 640 x 640 piksela.

Korak augmentacije[5] omogućuje generiranje novih verzija fotografija za model, ovisno o izabranim postupcima koji će biti primijenjeni na fotografije. Neki od prednosti dodatne obrade fotografija prije samog treninga uključuju bolju reproducibilnost modela, budući da su sve augmentirane fotografije pohranjene. Vrijeme treninga se skraćuje jer je proces augmentacije CPU intenzivan, pa GPU često mora čekati da CPU završi. Pretprocesiranjem se ovaj korak čekanja tijekom treninga izbjegava. Konačno, troškovi treninga su niži jer se GPU optimalnije koristi.

Neke od često korištenih augmentacija uključuju rotacije fotografije, promjenu ekspozicije, prekrivanje objekata, zamučivanje fotografije i druge. U svrhu eksperimenta, u prvim je eksperimentima korišten set fotografija bez primijenjenih augmentacija kako bi se testirala kvaliteta samog seta, dok je u kasnijim eksperimentima uključena opcija augmentacije radi testiranja performansi modela s raznolikijim setom podataka.

### 4.1.1. Sastav ulaznog skupa podataka

Pripremljen je ulazni skup podataka podijeljen u četiri klase: violina, violončelo, truba i flauta. Napravljene su četiri verzije skupa za potrebe eksperimenata, koje se razlikuju po broju označenih primjera i načinu generiranja fotografija. Prvi skup sastoji se od 126, drugi i treći od 295, a posljednji od 844 označenih primjera. Drugi skup podataka dobiven je iz prvog primjenom augmentacija nad fotografijama, dok su sve fotografije iz trećeg skupa pojedinačno prikupljene i međusobno različite po sadržaju. Primjenom augmentacija na drugi skup dobiven je četvrti skup. Ideja ovako sastavljenih skupova podataka je testirati kako različiti sklopovi podataka utječu na treniranje i performanse modela.



Slika 4.2. Raspodjela labela klasa unutar setova

Na slici 4.2. prikazana je zastupljenost pojedine klase u ulaznim setovima. Iz grafova je vidljivo da je omjer violončela i violine približno jednak i veći od zastupljenosti flaute i trube koje su također približno jednake.

Kako bismo maksimizirali generalizacijske performanse modela, koristimo pristup *unakrsne provjere*. Da bismo osigurali da model pravilno radi i na do tada neviđenim primjerima, ulazni skup podataka podijelili smo na tri podskupa. Najveći dio skupa (70%) čini skup za učenje (engl.*training set*), 20% primjera čini validacijski skup (engl.*validation set*), dok preostalih 10% čini skup za testiranje (engl.*test set*).

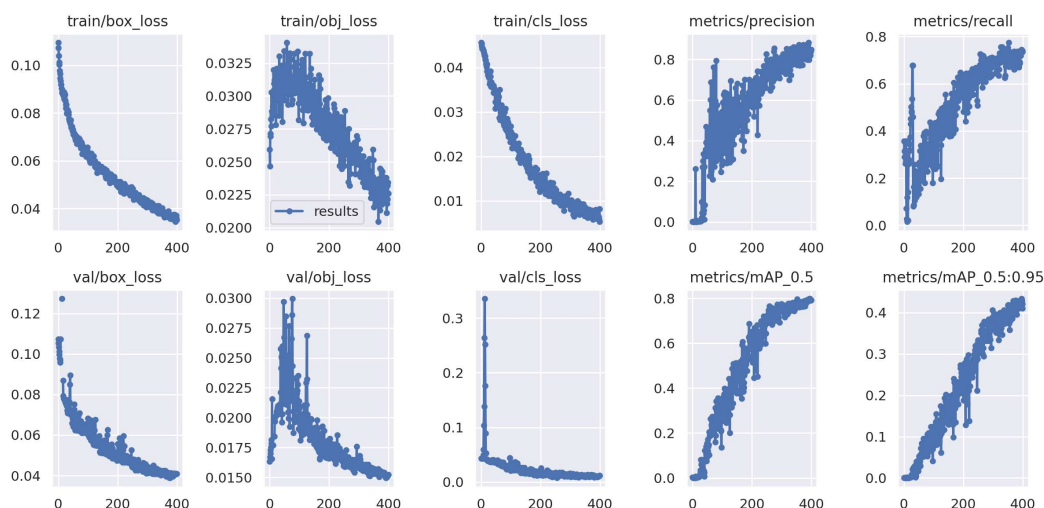
## 4.2. Google Colab bilježnica

Za potrebe učenja modela i provođenja eksperimenata korištena je Google Colab bilježnica, koja vodi korisnika kroz proces unošenja vlastitog skupa podataka do treniranja i evaluacije modela. Osim jednostavnosti korištenja, velika prednost ovakvog pristupa je besplatan GPU, koji korisniku omogućuje ubrzavanje procesa treniranja.[4]

Prvi korak je kreiranje vlastitog skupa podataka koji će kasnije biti učitani u bilježnicu. Potrebno je klonirati repozitorij YOLOv5 i pozicionirati se u njega te preuzeti vlastiti skup podataka u ispravnom formatu. Generirana je *yaml* datoteka koja specificira lokaciju svakog skupa podataka, kao i imena i broj klasa.

Sljedeći korak je konfiguriranje strukture mreže, koja uključuje kralježnicu za izdvajanje značajki i glavu za detekciju. Ovaj je korak preddefiniran u bilježnici te nije bilo potrebe za izmjenom. Prije samog treniranja potrebno je predati nekoliko argumenata funkciji za treniranje: *-size* koji označava veličinu na koju se fotografija reformatira prije ulaska u mrežu, *-batch-size* koji definira broj slika u jednoj seriji, *-epochs* koji označava broj epoha treniranja, putanju do *yaml* konfiguracijske datoteke ulaznog skupa te dodatne parametre kao što su ime izlazne datoteke s rezultatima i drugi.

Po završetku treniranja izgenerirani su grafovi s rezultatima procesa treniranja i validacije, a primjerak grafa prikazan je fotografijom 4.3.



**Slika 4.3.** Rezultati treniranja i validacije modela

### 4.2.1. Treniranje

Prilikom treniranja prate se krivulje gubitka kod treniranja okvira (`train/box_loss`), gubitka kod treniranja objekta (`train/obj_loss`) i gubitka kod treniranja klase (`train/cls_loss`). Praćenje ovih krivulja od iznimne je važnosti jer pruža uvid u napredak modela tijekom učenja.

Krivulja gubitka kod treniranja okvira prikazuje učenje modela u prepoznavanju okvira objekata te smanjenje pogreške kroz proces učenja. Smanjeni gubitak kod treniranja okvira ukazuje na veću točnost i preciznost u lociranju objekata.

Krivulja gubitka kod treniranja objekta mjeri sposobnost modela da identificira nalazi li se u ćeliji objekt koji je potrebno klasificirati ili je to pozadina. Smanjenje ove greške ukazuje na poboljšanje u performansama modela u razlikovanju objekata od pozadine.

Krivulja gubitka kod treniranja klase prati napredak modela u klasificiranju detektiranih objekata. Smanjenje gubitka kod treniranja klase ukazuje na uspješno učenje modela u klasificiranju objekata.[2]

### 4.2.2. Vrednovanje

Kako bi se testirale performanse modela na dosad neviđenim primjerima i odredila optimalna složenost modela, koristi se proces validacije. Rezultati modela na validacijskom



skupu prate se kroz krivulje gubitaka kod treniranja okvira, objekata i klasa. Padajući trend ovih krivulja ukazuje na poboljšanje modela u lociranju objekata, detekciji ćelija, odnosno razlikovanju objekata od pozadine, te konačno u klasifikaciji detektiranih objekata.

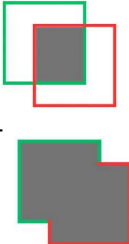
## 5. Provođenje eksperimenata

Cilj sljedećih eksperimenata bio je pokazati kako promjene parametara prilikom testiranja modela utječu na njegove performanse. Izabrana su dva eksperimentalna slučaja. Prvi eksperiment istražuje utjecaj broja epoha treninga, dok drugi eksperiment ispituje utjecaj veličine ulaznog seta podataka. Korišteni model je predtreniran na COCO skupu podataka, koji ne uključuje odabrane klase, što sugerira da će bez adekvatnog treninga model biti nesposoban ispravno klasificirati predane objekte. Rezultati eksperimenata analiziraju se usporedbom evaluacijskih mjera, a nalaze se predstavljene u nastavku.

### 5.1. Vrednovanje rezultata

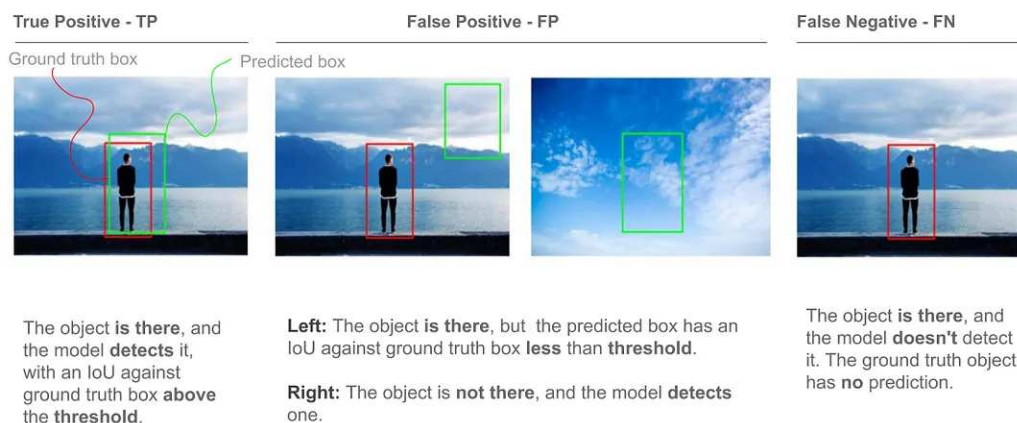
Kako bi se bolje razumjele performanse modela za detekciju objekata, koriste se razne mjere koje omogućuju detaljnu analizu problema i prednosti modela. Osnovne mjere vrednovanja[6] uključuju matricu zabune (*confusion matrix*), točnost (*accuracy*), preciznost (*precision*), odziv (*recall*) i F1 mjeru. Ključna mjera za određivanje točnosti predikcije položaja objekta je omjer presjeka i unije predviđenog i stvarnog okvira objekta (*Intersection over Union* - IoU).

IoU mjeri stupanj preklapanja između predviđenog okvira (*bounding box*) i stvarnog okvira objekta (*ground truth*). Izračunava se kao omjer površine presjeka tih dvaju okvira i površine njihove unije. Što je vrijednost mjere IoU veća, to je predikcija točnija. Idealno, savršeno preklapanje predviđenog i stvarnog okvira rezultirat će vrijednošću 1, dok će potpuni nedostatak preklapanja rezultirati vrijednošću 0.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


**Slika 5.1.** Formula za računanje omjera presjeka i unije okvira[6]

Koristeći mjeru IoU, definiraju se 4 kategorije rezultata predikcije: točno pozitivan (*true positive* - TP), lažno pozitivan (*false positive* - FP), lažno negativan (*false negative* - FN) i točno negativan (*true negative* - TN). Točno pozitivan rezultat znači da model točno identificira i lokalizira objekt te da mu je vrijednost metrike IoU iznad zadanog praga točnosti. Lažno pozitivan model identificira objekt kojeg nema ili je vrijednost mjere IoU ispod praga. Lažno negativan model ne prepoznaje objekt koji se nalazi u slici. Točno negativan ispravno ignorira detekciju objekta. Međutim u kontekstu detekcije objekata, ova mjera nije relevantna jer se kosi s ciljem detekcije objekata.

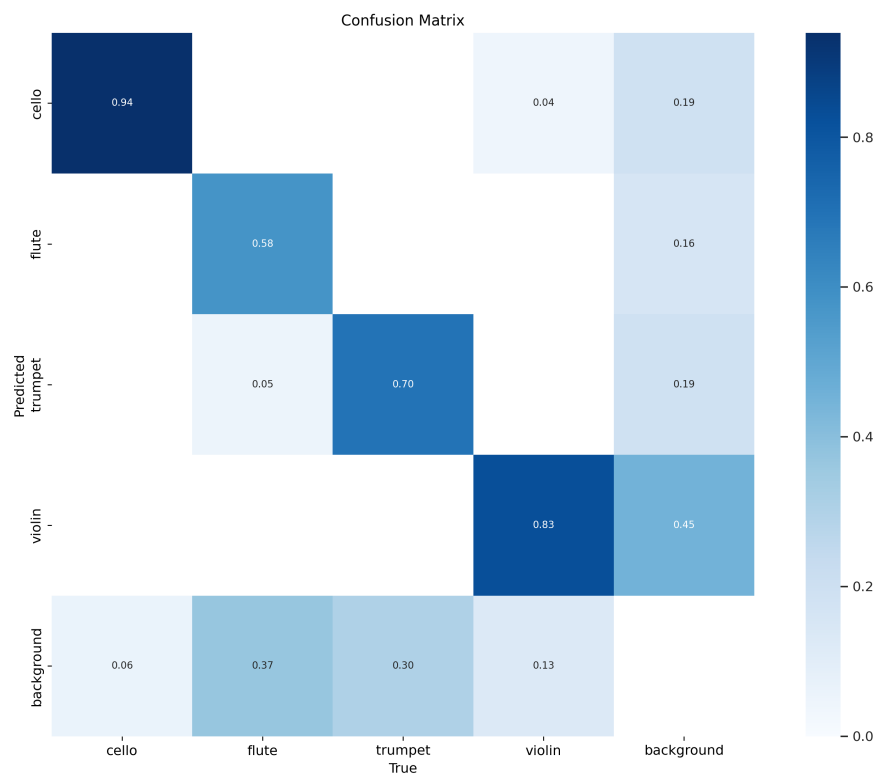


**Slika 5.2.** Slikovni prikaz kategorija TP, FP i FN[6]

### 5.1.1. Matrica zabune

Matrica zabune (*confusion matrix*) koristi se za usporedbu stvarnih oznaka i predikcije modela. Na slikovit način preključuje sažetak predikcija, pokazujući koliko su predikcije ispravno, a koliko neispravno klasificirane. Matrica pomaže u razumijevanju koje su

klase često pogrešno interpretirane ili neprepoznate, što omogućuje analizu performansi i identificiranje područja koja zahtjevaju poboljšanje.



**Slika 5.3.** Primjer matrice zabune

Prikazana na slici 5.3. matrica zabune sastoji se od dvije osi gdje X-os predstavlja stvarne klase, a Y-os predviđene. Glavna dijagonala matrice predstavlja točne predikcije, dok je sve izvan dijagonale pogrešna predikcija. Iz analize matrice konfuzije slijede ostale validacijske mjere specijalizirane za specifične karakteristike.

### 5.1.2. Preciznost

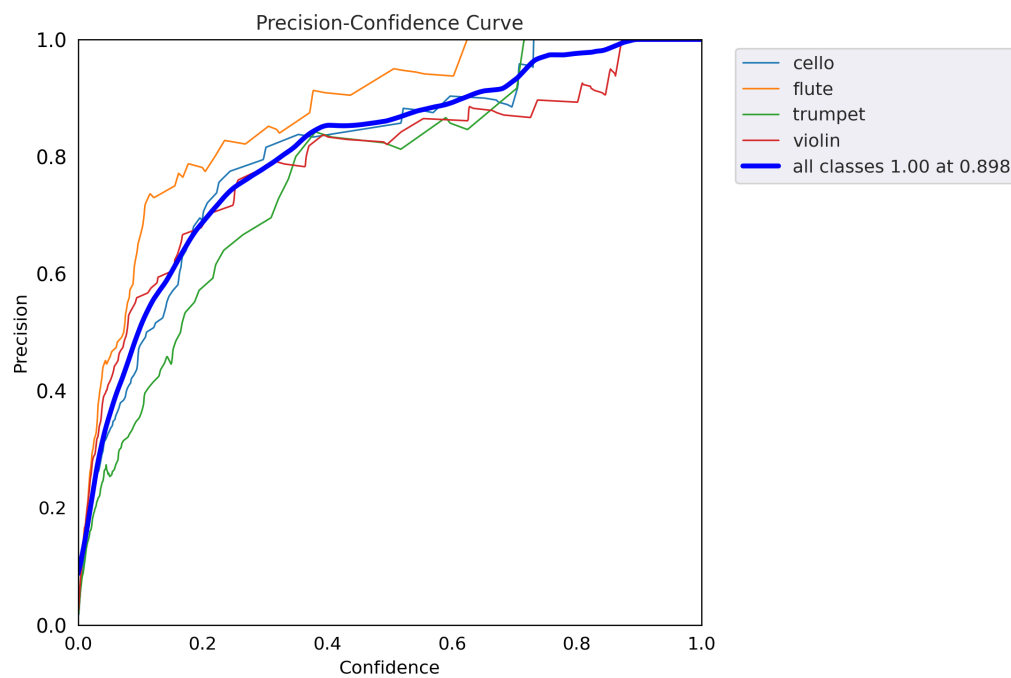
Mjera preciznosti (engl.*precision*) pokazuje udio točnih predikcija u odnosu na sve predikcije koje model daje. Visoki stupanj preciznosti znači da model daje pouzdane točne predikcije[7].

$$P = \frac{TP}{TP + FP} \quad (5.1)$$

Parametar preciznosti zanimljivo je razmatrati u kontekstu razine pouzdanosti koja se koristi prilikom određivanja točnosti predikcije. U svrhu toga, razmotrit ćemo kri-

vulju preciznosti i pouzdanosti (engl. *precision-confidence curve*) prikazanu na slici 5.4. Na grafu, X-os prikazuje nivo razine pouzdanosti, odnosno prag iznad kojeg se predikcija smatra pozitivnom, dok Y-os prikazuje iznos preciznosti. Krivulja ilustrira kako se preciznost mijenja ovisno o razini pouzdanosti.

Što je prag viši, to je i preciznost veća jer se samo vrlo sigurne predikcije smatraju tačnima. S nižom razinom pouzdanosti, preciznost se smanjuje jer se generira sve više lažno pozitivnih primjera. Ako prag ostavimo visokim, dobivamo veću preciznost jer model generira manje lažno pozitivnih predikcija, ali time smanjujemo efikasnost. Analizom krivulje nastoji se pronaći kompromis između preciznosti i efikasnosti, ovisno o razini praga pouzdanosti[2].



**Slika 5.4.** Graf odnosa preciznosti i razine pouzdanosti

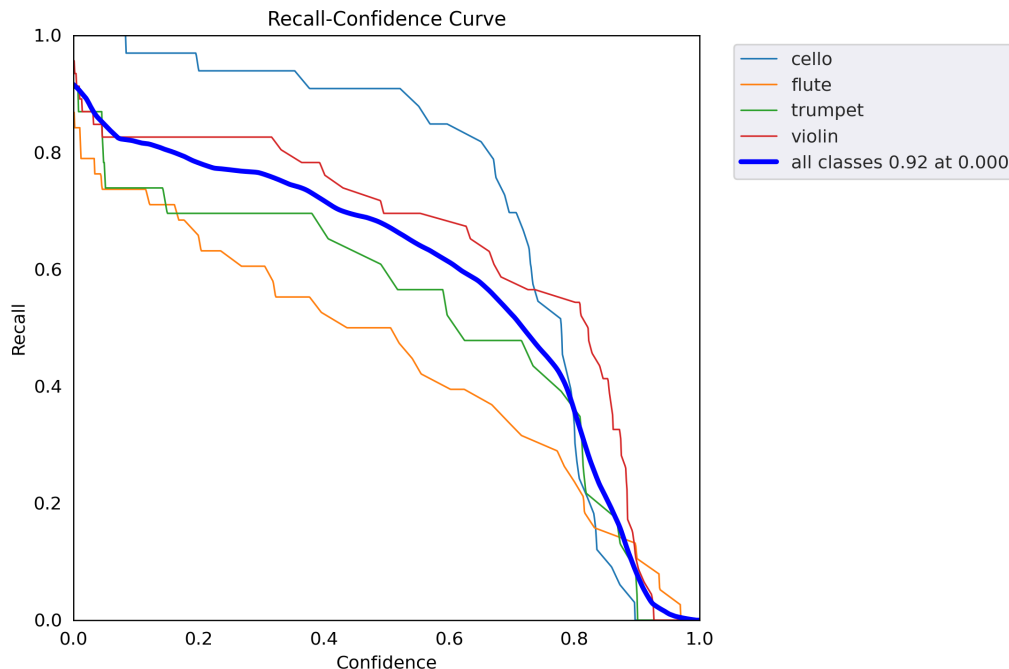
### 5.1.3. Odziv

Odziv (engl.*recall*) je mjera koja iskazuje sposobnost modela da identificira sve relevantne objekte u slici. Sukladno tome, odziv je definiran kao udio točnih predikcija u odnosu na ukupan broj mogućih predikcija[7].

$$R = \frac{TP}{TP + FN} \quad (5.2)$$

U kontekstu razine pouzdanosti, razmatra se krivulja odziva i pouzdanosti (engl.*recall-confidence curve*). Ova krivulja prikazuje kako se odziv mijenja s variranjem razine pouzdanosti.

Kada je razina pouzdanosti niža, model lakše generira predikcije, te je veći broj predikcija klasificiran kao validan. Međutim, niža razina pouzdanosti također znači i veći broj lažno pozitivnih predikcija, što smanjuje efikasnost modela. Analiza krivulje ima za cilj pronaći najbolji balans između postizanja većeg broja točnih predikcija i minimiziranja lažno pozitivnih klasifikacija[2].



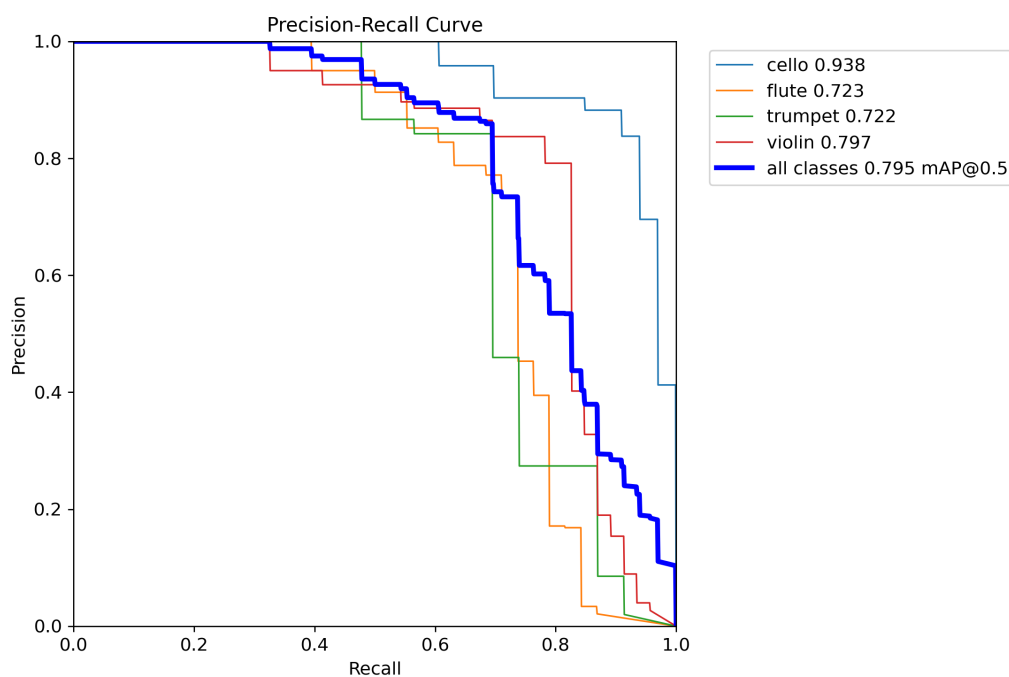
**Slika 5.5.** Graf odnosa odziva i razine pouzdanosti

### 5.1.4. Krivulja preciznosti i odziva

Analizu mjera preciznosti i odziva moguće je sažeto prikazati na istom grafu. Graf na slici 5.6. prikazuje promjenu preciznosti modela ovisno o povećanju odziva. Na X-osi grafa nalazi se odziv, dok je na Y-osi prikazana preciznost.

Analizom grafa primjećuje se trend smanjenja preciznosti kako se odziv povećava. Veći odziv podrazumijeva veći broj detektiranih objekata i smanjenje broja lažno negativnih predikcija. Međutim, povećanjem odziva dolazi do pada preciznosti jer se povećanje odziva postiže smanjenjem praga pouzdanosti.

Ova krivulja preciznosti i odziva (engl.*precision-recall curve*) omogućava bolje razumijevanje kompromisa između ovih dviju mjera, te pomaže u pronalaženju optimalnog praga pouzdanosti za uravnoteženje preciznosti i odziva u modelu.



**Slika 5.6.** Graf odnosa preciznosti i odziva

### 5.1.5. F1-mjera

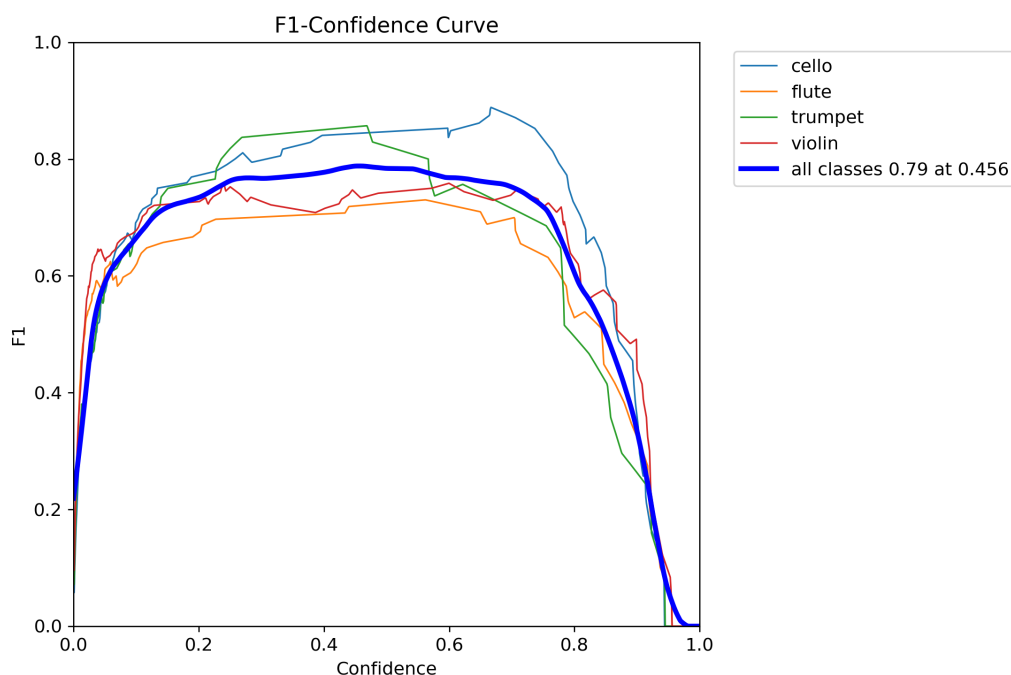
Mjere preciznosti i odziva često djeluju suprotno jedna drugoj. Povećanje preciznosti povezano je sa strožim pragom pouzdanosti, čime se direktno smanjuje broj objekata koje model prepoznaje. S druge strane, povećanje odziva dovodi do većeg broja negativnih objekata koji se klasificiraju kao pozitivni, čime se smanjuje preciznost.

Kako bi se pronašla ravnoteža između preciznosti i odziva, koristi se F1-mjera. F1-mjera predstavlja harmonijsku sredinu između preciznosti i odziva te se izračunava prema formuli[8]:

$$F_1 = \frac{2PR}{P + R} \quad (5.3)$$

Maksimiziranjem F1-mjere postiže se uravnoteženo povećanje i preciznosti i odziva, što je ključno za optimalnu izvedbu modela.

F1-mjeru moguće je grafički prikazati u ovisnosti o pragu pouzdanosti, kao što je prikazano na slici 5.7. Na grafu, X-os predstavlja prag pouzdanosti, dok Y-os prikazuje F1-mjeru. Ovaj prikaz omogućava vizualnu analizu kako se F1-mjera mijenja s variranjem praga pouzdanosti, pomažući pri odabiru optimalnog praga za maksimalnu uravnoteženost između preciznosti i odziva.



Slika 5.7. F1 krivulja



## 5.1.6. Srednja prosječna preciznost

Srednja prosječna preciznost (mAP)[9] mjeri performanse natreniranih modela te se računa kao srednja vrijednost preciznosti za svaku klasu. Vrijednost mAP-a se kreće između 0 i 1.

Postupak računanja započinje izborom IoU praga pouzdanosti, najčešće 0.5 (mAP@0.5). Za svaku klasu potrebno je izračunati srednju preciznost (AP). Srednja preciznost izračunava se kao površina ispod krivulje preciznosti i odziva (engl. *precision-recall curve*), što predstavlja integral te krivulje.

$$AP = \int_{r=0}^1 p(r) dr \quad (5.4)$$

Nakon što su srednje preciznosti izračunate za svaku klasu, mAP se dobiva računajući njihovu srednju vrijednost. Ovaj postupak omogućava objektivnu procjenu točnosti modela u prepoznavanju različitih klasa objekata.

$$mAP = \frac{1}{k} \sum_i^k AP_i \quad (5.5)$$

## 5.2. Eksperimenti nad Yolov5 modelu

### 5.2.1. Utjecaj broja epoha treninga

Prvi eksperiment proveden je s ciljem učenja modela YOLOv5 kroz različiti broj epoha. Cilj je bio utvrditi koliko broj epoha treniranja utječe na performanse modela te optimizirati sam proces učenja. Pretpostavka eksperimenta je da će povećanjem broja epoha rasti i performanse modela.

Eksperiment je proveden kroz tri testna primjera. Odabran je treći skup podataka, sastavljen od 295 različitih fotografija. Ulazni skup podataka podijeljen je na tri podskupa: skup za treniranje (70%), validacijski skup (20%) i skup za testiranje (10%). Veličina serije (engl. *batch size*) postavljena je na 16, a broj epoha treniranja redom je postavljen na 100, 200 i 400, pri čemu se svakim sljedećim treningom broj epoha udvostručio.

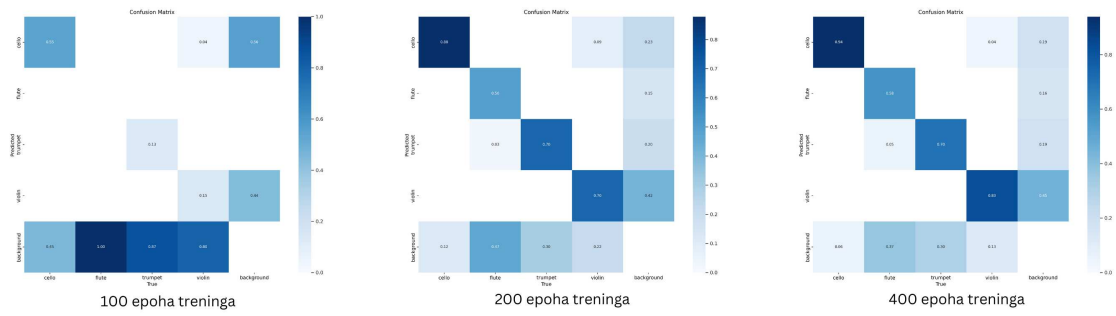
Nakon 100 epoha treniranja dobiveni su sljedeći rezultati: slika 5.8. prikazuje dobitu matricu zabune, a slika 5.9. prikazuje F1 krivulju. Analizom matrice zabune možemo uočiti da većina primjera ne pada na glavnu dijagonalu matrice, već su većinom zamijenjeni s pozadinom. Drugim riječima, postoji veliki udio lažno negativnih primjera, što ukazuje na slabost modela u identificiranju prisutnosti objekata u slici. Ovaj problem je vidljiv i u metrici `obj_loss`, koja se tijekom treninga drži na stalnoj razini umjesto da pokazuje trend pada. Ako pogledamo performanse modela nad fotografijama iz seta za testiranje, vidljivo je da na većini fotografija model ne daje nikakve predikcije, a na nekoliko fotografija s danim predikcijama njihova je pouzdanost maksimalno 0.51.

Već s 200 epoha treninga rezultati su znatno bolji. Matrica konfuzije pokazuje veći broj ispravno detektiranih i klasificiranih objekata kao i manji broj lažno negativnih predikcija. F1 krivulja viša je u odnosu na prethodnu i pokazuje bolji odnos preciznosti i odziva modela. U setu za testiranje smanjen je udio fotografija bez predikcija, a pouzdanost predikcije klasa ovog modela veća je u odnosu na prethodni.

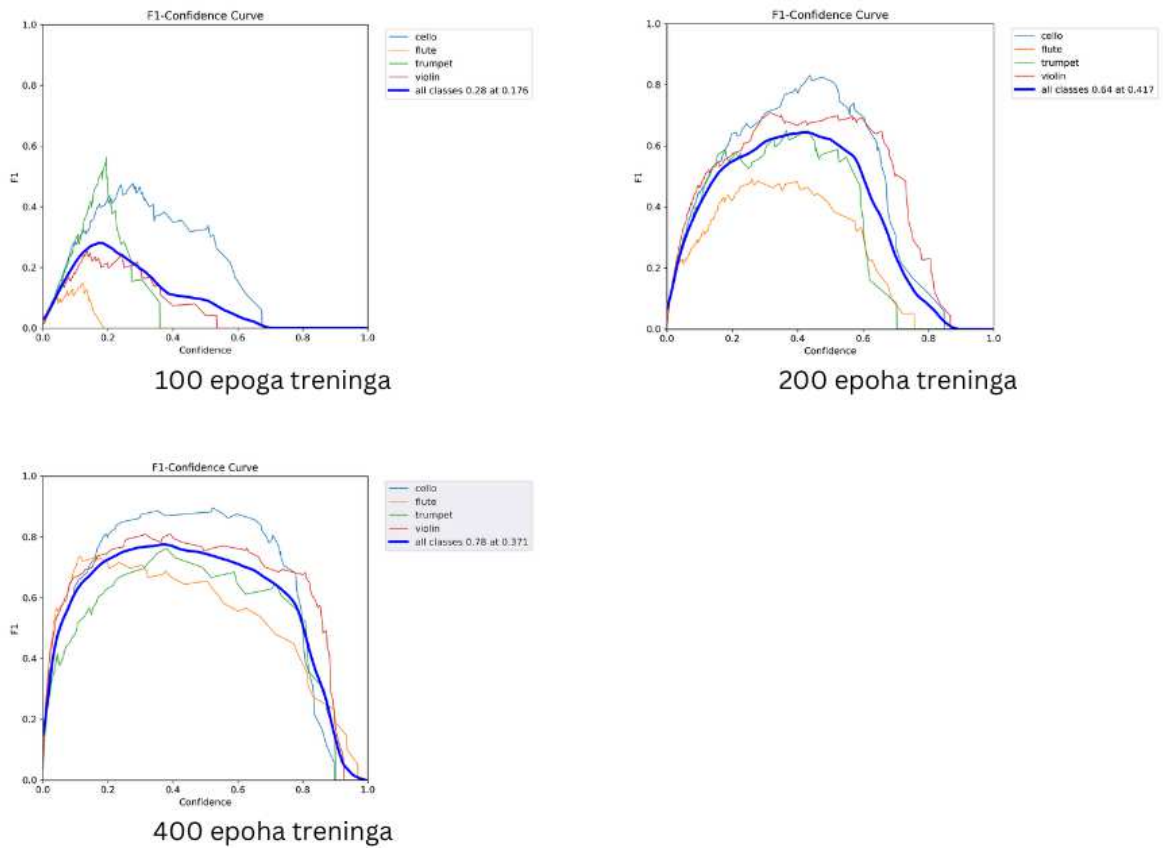
Prolaskom kroz 400 epoha treninga uočava se manji skok u performansama u odnosu na skok između 100 i 200 epoha. Model je nešto precizniji od prethodnog, a najveći je pomak u odnosu na prethodni vidljiv u srednjoj prosječnoj preciznosti predikcije flaute koja je skočila s 0.403 na 0.723.

Tablica 5.1. Srednja prosječna preciznost detekcije klasa

klasa	mAP50 (100 epoha)	mAP50 (200 epoha)	mAP50 (400 epoha)
sve klase	0.283	0.673	0.795
violončelo	0.456	0.857	0.938
flauta	0.0656	0.403	0.723
truba	0.45	0.707	0.722
violina	0.16	0.726	0.797



Slika 5.8. Usporedba matrica zabune



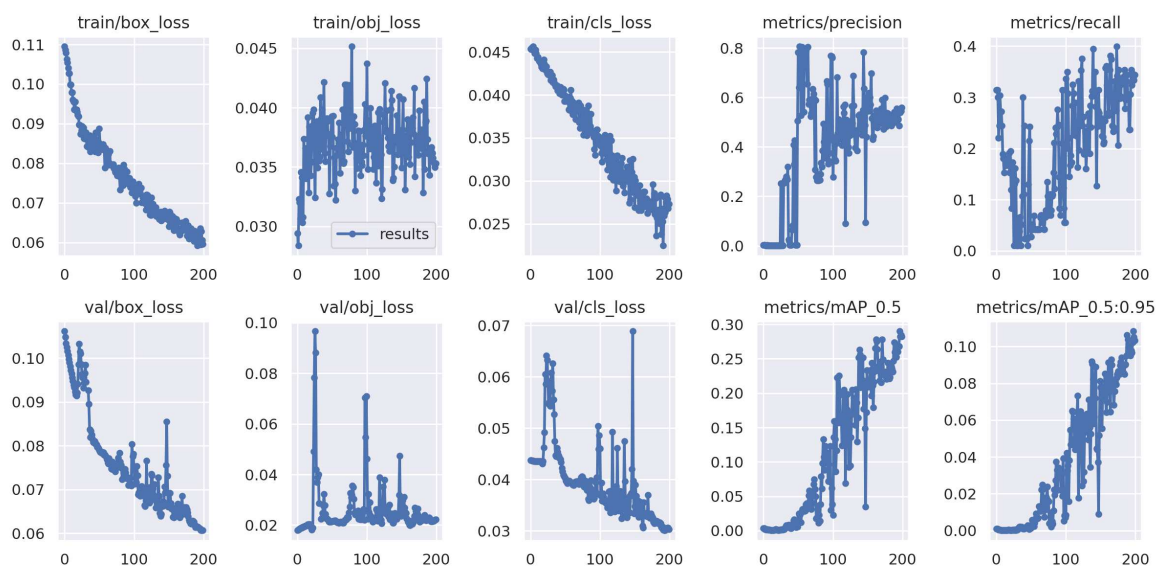
Slika 5.9. Usporedba f1 krivulja

## 5.2.2. Utjecaj veličine ulaznog skupa podataka

U sljedećem eksperimentu ispituje se utjecaj veličine ulaznog skupa podataka na performanse modela tijekom 200 epoha treniranja. Pretpostavka je da će veći ulazni skup rezultirati boljim performansama modela, jer će veći broj primjera omogućiti modelu bolje generaliziranje na nove, dosad neviđene primjere. Također, dodatno ćemo ispitati utjecaj izvora fotografija na performanse modela uspoređujući performanse skupa dobivenog prikupljanjem različitih fotografija (skup 3) s performansama skupa dobivenog augmentacijom manjeg skupa (skup 2).

Provedeno je testiranje prvog skupa podataka, sastavljenog od 126 primjera. Rezultati su pokazali prilično slabe performanse modela. Analizom matrice zabune uočava se veliki udio lažno negativnih primjera jer model loše detektira prisutnost objekta na fotografijama. Osim toga, značajna je zastupljenost lažno pozitivnih predikcija, jer model pozadinu često prepoznaje kao objekt određene klase.

Ovi rezultati jasno su vidljivi i na krivulji treninga (5.10.), koja pokazuje slab trend smanjenja gubitka kod treniranja objekta (engl. *train-obj\_loss*), kao i prilično nestabilan trend rasta preciznosti i odziva.



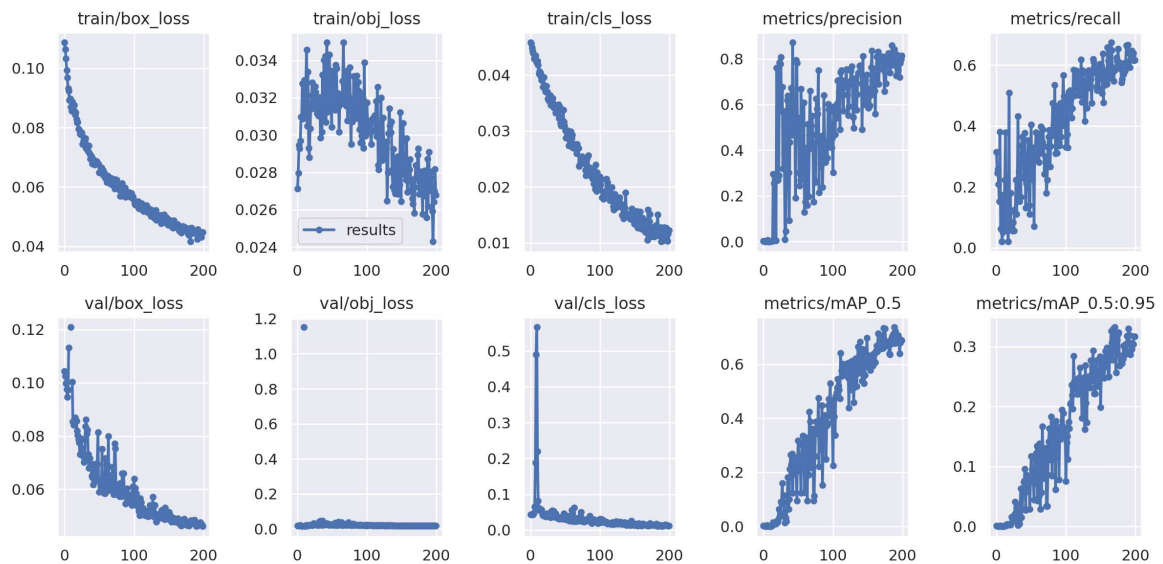
Slika 5.10. Trening prvog ulaznog seta

Proces učenja ponovljen je s drugim skupom podataka, koji je dobiven primjenom augmentacija na prethodni skup. Rezultati potvrđuju pretpostavku te pokazuju značajno poboljšanje performansi modela u odnosu na prethodne rezultate. Veći broj primjera

klasificiran je ispravno, što se očituje u matrici zabune koja je pretežno ispunjena na glavnoj dijagonali. Ipak, postoji prostor za napredak, jer model i dalje ne pronalazi sve objekte u slici ili zamjenjuje pozadinu za objekte.

Analizom grafova treninga (5.11.) vidljiv je povoljniji trend smanjenja gubitka kod treniranja objekata, iako prisutne fluktuacije ukazuju na probleme s konzistentnim prepoznavanjem objekata. Graf gubitka kod treniranja klasa pokazuje pravilan pad, što upućuje na to da model dobro uči klasificirati objekte.

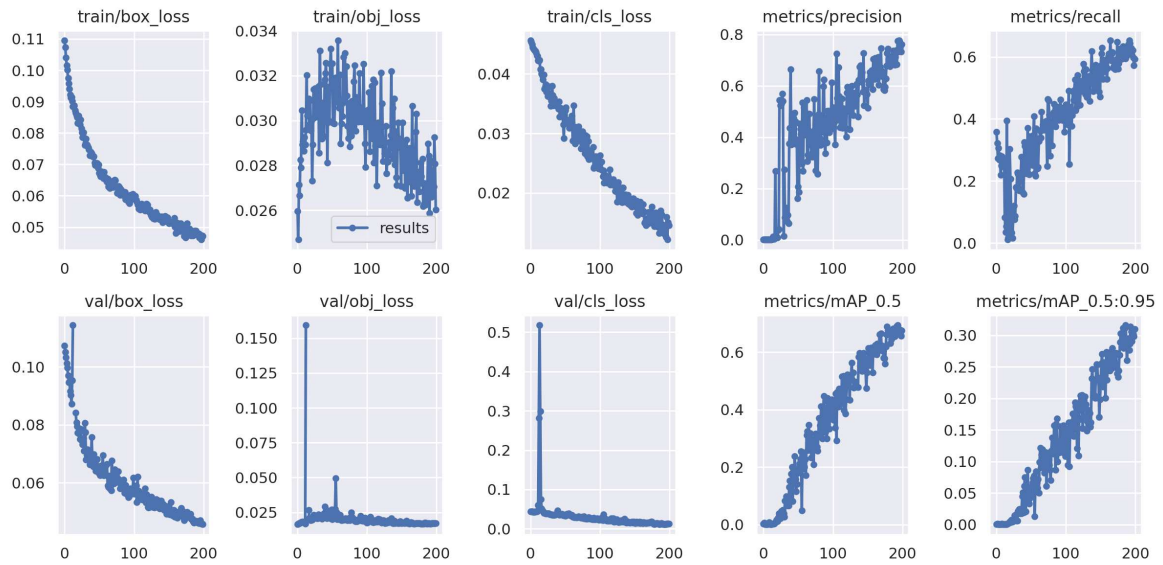
Preciznost i odziv ravnomjernije rastu, no prisutne oscilacije sugeriraju potrebu za dodatnim prilagodbama kako bi se postigla stabilnija i konzistentnija performansa modela.



**Slika 5.11.** Trening drugog ulaznog seta

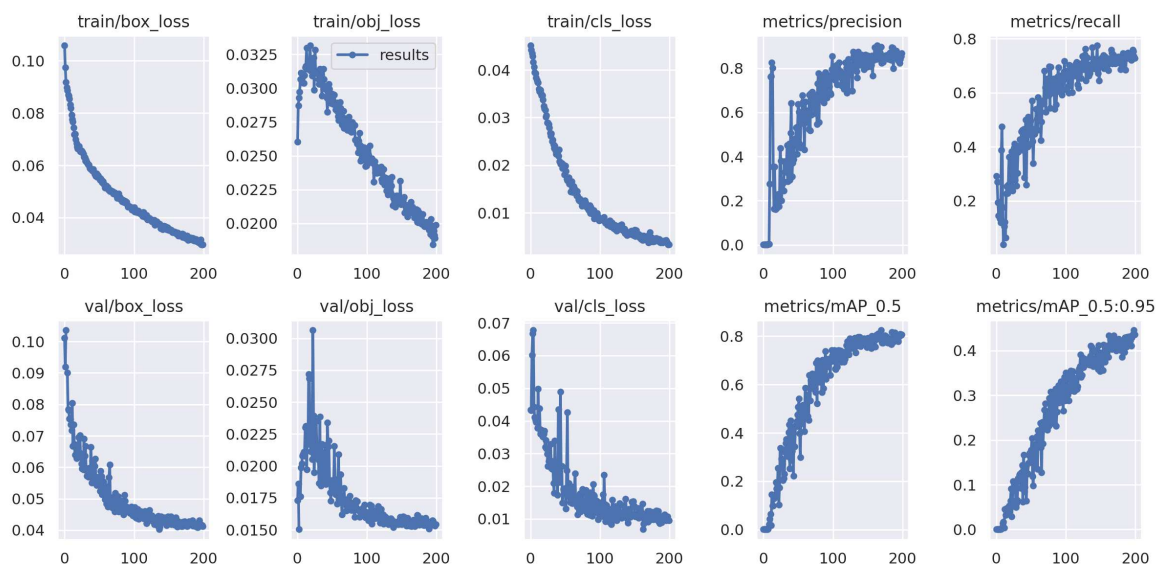
Sljedeći skup podataka sadrži jednak broj primjera kao prethodni, ali su svi ulazni primjeri odvojeno prikupljeni i međusobno različiti. Cilj ovog eksperimenta bio je dokazati da način prikupljanja fotografija ne utječe značajno na proces učenja modela ako je skup podataka jednake veličine. Rezultati podržavaju ovu pretpostavku.

Većina klasificiranih primjera nalazi se na glavnoj dijagonali matrice zabune, iako su i dalje prisutne lažne pozitivne i lažne negativne klasifikacije. U usporedbi s prethodnim treningom, dobiveni grafovi pokazuju gotovo identične rezultate, što dodatno potvrđuje zaključak da način generiranja fotografija nema značajan utjecaj na performanse modela kada je veličina skupa podataka ista.

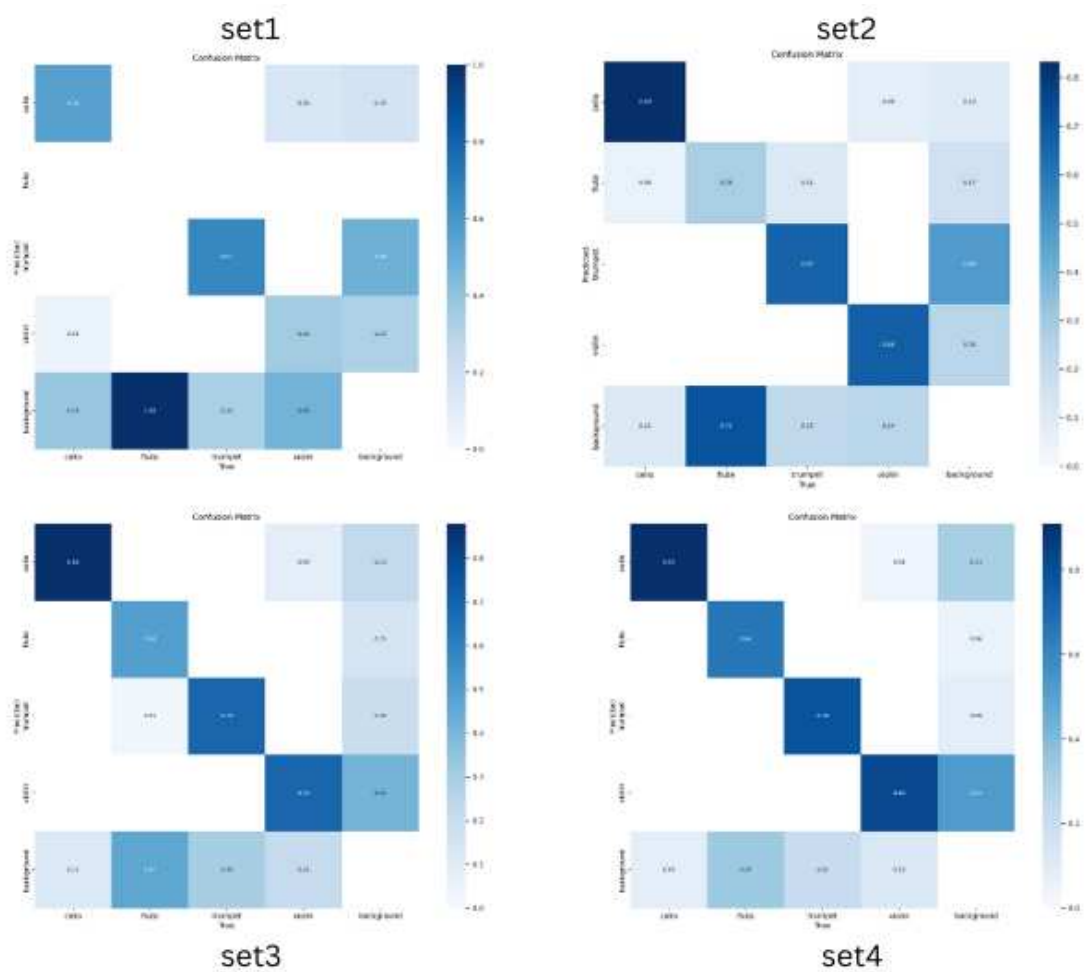


**Slika 5.12.** Trening trećeg ulaznog seta

Posljednji trening proveden je na najvećem skupu podataka, koji je 2,8 puta veći od prethodna dva skupa. Rezultati pokazuju pozitivan trend poboljšanja performansi modela, s većom preciznošću u predviđanju klase objekta i smanjenim udjelom lažno pozitivnih i lažno negativnih klasifikacija. Trendovi pada gubitka za objekte, klase i okvire pokazuju značajno smanjenje oscilacija u odnosu na prethodne eksperimente (5.13.), što ukazuje na stabilniji i točniji model.



**Slika 5.13.** Trening četvrtog ulaznog seta



Slika 5.14. Usporedba matrica zabuna za četiri ulazna seta

Tablica 5.2. Srednja prosječna preciznost detekcije klasa

klasa	mAP50 (set 1)	mAP50 (set 2)	mAP50 (set 3)	mAP50 (set 4)
sve klase	0.281	0.57	0.673	0.805
violončelo	0.521	0.784	0.857	0.878
flauta	0.0667	0.237	0.403	0.696
truba	0.255	0.562	0.707	0.828
violina	0.282	0.698	0.726	0.819

### 5.2.3. Diskusija rezultata

Ideja prethodnih eksperimenata bila je demonstrirati optimalne prakse za budući rad i treniranje modela. Rezultati prvog eksperimenta ukazuju na potrebu za većim brojem epoha tijekom treninga, jer kraći period učenja ne daje zadovoljavajuće rezultate. Pri odabiru broja epoha važno je pronaći optimalnu ravnotežu i izbjegavati prevelik broj epoha kako bi se spriječila pretreniranost modela.

Drugi eksperiment istraživao je važnost ulaznog skupa podataka za model. Rezultati jasno pokazuju da veći broj fotografija i instanci klase značajno poboljšava performanse modela. Stoga je preporučljivo za buduće treninge prikupiti što veći broj instanci iz svake klase, vodeći računa o varijabilnosti primjera kako bi model bolje naučio prepoznavati objekte. Jednostavan način za povećanje ulaznog skupa je primjena augmentacija nad prikupljenim fotografijama, budući da je pokazano da trening nad takvim setom dovodi do jednako dobrih rezultata.

Matrice zabune reflektiraju činjenicu da je, primjerice, broj instanci flaute manji od broja instanci violončela, što je vidljivo u rezultatima svih primjera. Model je postigao visoku pouzdanost u prepoznavanju violončela, dok su rezultati za flautu uvijek relativno niski. To upućuje na zaključak da je za budući rad potrebno generirati veći broj instanci te klase, kao i uvesti veći broj perspektiva na objekt kako bi ga model bolje naučio prepoznavati.



## 6. Zaključak

Ovaj rad daje uvod u jedan od temeljnih problema računalnog vida, detekcija objekata, čiji je cilj lokalizirati i prepoznati objekte u slici. Naglasak rada je na YOLO modelu koji pripada detektorima u jednoj fazi. Dan je pregled algoritma rada i postupak treniranja. Detaljno su analizirane metrike kojima se opisuju performanse modela i samog treninga, a sve u svrhu boljeg razumijevanja provedenih pokusa.

U sklopu rada, izrađen je vlastiti skup podataka korišten za učenje te su osmišljeni i provedeni pokusi koji ilustriraju dobre prakse prilikom treniranja modela. Svaki pokus analiziran je preko izlaznih parametara i popratnih grafova.

Provedeni pokusi jasno demonstriraju važnost nekoliko ključnih aspekata u procesu treniranja modela za prepoznavanje i klasifikaciju objekata. Rezultati prvog eksperimenta upućuju na to da je korištenje većeg broja epoha ključno u procesu učenja. Prekratak period učenja dovodi do suboptimalnih rezultata, dok predugačak period može rezultirati smanjenom sposobnošću generalizacije.

Kako bi promjena broja epoha treniranja i dovela do ikakvih rezultata temeljni je zadatak priprema dobrog ulaznog seta podataka. Ravnomjerna i velika zastupljenost svake klase, kao i precizno i dosljedno označavanje primjera temelj su svakog treniranja. Jednom označeni skup možemo provesti kroz korak augmentacije te tako na jednostavan način dodatno povećati skup podataka s kojim radimo.

## Literatura

- [1] A. Sharma, "Introduction to the YOLO family", u *PyImageSearch*, D. Chakraborty, P. Chugh, A. R. Gosthipaty, S. Huot, K. Kidriavsteva, R. Raha, i A. Thanki, Ur., 2022., <https://pyimg.co/dgbvi>.
- [2] L. YU i S. LIU, "A single-stage deep learning-based approach for real-time license plate recognition in smart parking system", *International Journal of Advanced Computer Science and Applications*, sv. 14, 01 2023. <https://doi.org/10.14569/IJACSA.2023.01409119>
- [3] J. Redmon, S. Divvala, R. Girshick, i A. Farhadi, "You only look once: Unified, real-time object detection", u *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [4] J. Solawetz i J. Nelson. (2020., jun) How to train a yolov5 model on a custom dataset. Accessed: 2024-06-04. [Mrežno]. Adresa: <https://blog.roboflow.com/how-to-train-yolov5-on-a-custom-dataset/>
- [5] Roboflow. (2024., jun) Image augmentation. Accessed: 2024-06-04. [Mrežno]. Adresa: <https://docs.roboflow.com/datasets/image-augmentation>
- [6] H. Vedoveli. (2023., jun) Metrics matter: A deep dive into object detection evaluation. Accessed: 2024-06-07. [Mrežno]. Adresa: <https://medium.com/@henriquevedoveli/metrics-matter-a-deep-dive-into-object-detection-evaluation-ef01385ec62>
- [7] J. Lowe. (2022., mar) Precision and recall in machine learning. Accessed: 2024-06-07. [Mrežno]. Adresa: <https://blog.roboflow.com/precision-and-recall/>

- [8] V7 Labs. (2023) F1 score guide: Formula, calculation, and interpretation. Accessed: 2024-06-07. [Mrežno]. Adresa: <https://www.v7labs.com/blog/f1-score-guide>
- [9] J. Solawetz. (2020., may) What is mean average precision (map) in object detection? Accessed: 2024-06-06. [Mrežno]. Adresa: <https://blog.roboflow.com/mean-average-precision/>

# Sažetak

## Prepoznavanje objekata primjenom dubokog učenja

Tara Baće

U ovom istraživanju objašnjen je problem detekcije objekata u slikama te princip rada jednoprolaznih detektora. Proveden je trening modela YOLOv5 na vlastitom skupu podataka s ciljem razlikovanja glazbenih instrumenata. Uspoređeni su rezultati dobiveni nad različitim ulaznim skupovima podataka te je detaljno opisan postupak vrednovanja tih rezultata.

**Ključne riječi:** YOLO, detekcija objekata, duboko učenje

# **Abstract**

## **Uncomputable Computability**

Tara Baće

This research explains the problem of object detection in images and the working principle of single-shot detectors. The YOLOv5 model was trained on a custom dataset to distinguish musical instruments. The results obtained from different input datasets were compared, and the evaluation process of these results was described in detail.

**Keywords:** Yolo, object detection, deep learning