

# Detekcija registarskih pločica i prepoznavanje znakova koristeći modele dubokog učenja

---

Grgić, Anto

Master's thesis / Diplomski rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:122:802485>

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

Download date / Datum preuzimanja: **2025-01-29**

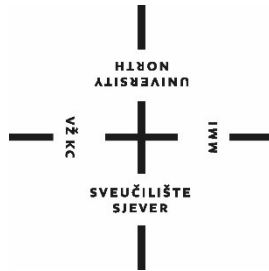


Repository / Repozitorij:

[University North Digital Repository](#)



**SVEUČILIŠTE SJEVER**  
**SVEUČILIŠNI CENTAR VARAŽDIN**



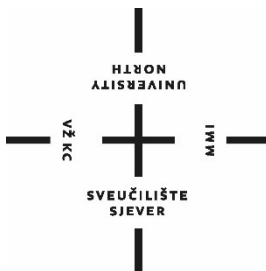
DIPLOMSKI RAD br. 091-MMD-2023

**DETEKCIJA REGISTARSKIH PLOČICA I  
PREPOZNAVANJE ZNAKOVA KORISTEĆI  
MODELE DUBOKOG UČENJA**

Anto Grgić

Varaždin, rujan 2023.

**SVEUČILIŠTE SJEVER**  
**SVEUČILIŠNI CENTAR VARAŽDIN**  
**Studij Multimedija**



DIPLOMSKI RAD br. 091-MMD-2023

**DETEKCIJA REGISTARSKIH PLOČICA I  
PREPOZNAVANJE ZNAKOVA KORISTEĆI  
MODELE DUBOKOG UČENJA**

Student:

Anto Grgić, 0016119941

Mentor:

izv. prof. dr. sc. Emil Dumić

Varaždin, rujan 2023.



## Prijava diplomskog rada

### Definiranje teme diplomskog rada i povjerenstva

ODJEL	Odjel za multimediju		
STUDIJ	diplomski sveučilišni studij Multimedija		
PRISTUPNIK	Grgić Anto	JMBAG	0016119941
DATUM	26.06.2023.	KOLEGIJ	Računalni vid
NASLOV RADA	Detekcija registarskih pločica i prepoznavanje znakova koristeći modele dubokog učenja		

NASLOV RADA NA ENGL. JEZIKU	License plate detection and character recognition using deep learning models
-----------------------------	--

MENTOR	Emil Dumić	ZVANJE	izv.prof.dr.sc.
--------	------------	--------	-----------------

ČLANOVI POVJERENSTVA	1.	doc. art. dr. sc. Mario Periša - predsjednik
	2.	izv. prof. dr. sc. Dean Valdec - član
	3.	izv. prof. dr. sc. Emil Dumić - mentor
	4.	doc. dr. sc. Andrija Bernik - zamjenski član
	5.	

### Zadatak diplomskog rada

BROJ	091-MMD-2023
------	--------------

OPIS  
U ovom radu će biti opisani i ispitani različiti modeli dubokog učenja za detekciju objekata - registarskih pločica i prepoznavanja znakova na njima.

Detekcija objekata je bitan korak u analizi slika i videozapisa, a općenito uključuje njihovu klasifikaciju i lokalizaciju. U teoretskom dijelu diplomskog rada bit će opisane različite metode za detekciju objekata s naglaskom na modele dubokog učenja: R-CNN (i poboljšanja), različite verzije SSD detektora modela (poput YOLO modela). Detaljnije će se opisati YOLO modeli zbog kasnijeg korištenja u praktičnom dijelu zadatka, s naglaskom na YOLOv7. Opisat će se mjere za usporedbu metoda: za klasifikaciju (preciznost, odziv, točnost, F1 mjera) i za detekciju (IoU, AP i mAP). Opisat će se neke od postojećih baza za učenje (PASCAL VOC 12, MS-COCO, ILSVRC, OpenImage). Također će se dati primjeri usporedbe opisanih modela nad nekom postojećom bazom poput MS-COCO baze, koristeći veličinu modela, brzinu izvođenja i mAP mjere.

U praktičnom dijelu rada će se koristiti YOLOv7 model za detekciju hrvatskih registarskih pločica, te EasyOCR model za prepoznavanje znakova. Bit će korištena neka od postojećih baza registarskih pločica za trening ili će biti izrađena vlastita baza. Koristi će se YOLOv7 model prilagođen za detekciju registarskih pločica i uspoređeni koristeći bazu za test po veličini modela i te mjerama za evaluaciju detekcije objekata.

ZADATAK URUČEN 01.09.2023.



POTPIS MENTORA  
Emil Dumić

Sveučilište  
Sjever

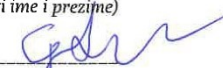


IZJAVA O AUTORSTVU  
I  
SUGLASNOST ZA JAVNU OBJAVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, ANTO GRGIĆ (ime i prezime) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog (obrisati nepotrebno) rada pod naslovom DETEKCIJA REGISTRARSKIH PLOČICA I PREPOZNAVANJE ZNAKOVA KORISTEĆI MODELE DUBOKOG UČENJA (upisati naslov) te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.


Student/ica:  
(upisati ime i prezime)

ANTO GRGIĆ,   
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnosti i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, ANTO GRGIĆ (ime i prezime) neopozivo izjavljujem da sam suglasan/na s javnom objavom završnog/diplomskog (obrisati nepotrebno) rada pod naslovom DETEKCIJA REGISTRARSKIH PLOČICA I PREPOZNAVANJE ZNAKOVA KORISTEĆI MODELE DUBOKOG UČENJA (upisati naslov) čiji sam autor/ica.

Student/ica:  
(upisati ime i prezime)

ANTO GRGIĆ,   
(vlastoručni potpis)

## ***Predgovor***

Računalni vid je postao izuzetno zanimljivo područje posljednjih godina zbog svoje široke primjene u različitim područjima poput prometa (autonomna vozila), robotike, medicinske dijagnostike, prepoznavanja lica, nadzora, te prepoznavanja registarskih pločica automobila. Detekcija i prepoznavanje objekata su ključni zadaci računalnog vida koji omogućuju strojevima da percipiraju i analiziraju svoju okolinu putem ulaznih signala u obliku slika ili videa. Posebno je važno imati sposobnost prepoznavanja i detekcije objekata u stvarnom vremenu u mnogim aplikacijama, kao što je autonomna vožnja, gdje pravovremeno otkrivanje objekata i njihova klasifikacija igraju ključnu ulogu u osiguravanju sigurnosti vozila i putnika.

Postoje razne vrste algoritama za praćenje objekata, no ovaj diplomski rad posvećen je istraživanju i implementaciji metoda za detekciju registarskih pločica i prepoznavanje znakova koristeći modele dubokog učenja, s posebnim fokusom na YOLO (You Only Look Once). Cilj ovog istraživanja je prikazati efikasne algoritme i sustave koji mogu precizno identificirati i izdvojiti registarske pločice s fotografija ili video snimaka, te kasnije automatski prepoznati znakove na njima koristeći OCR (Optical Character Recognition) algoritam.

Uvodni dio rada pruža pregled računalnog vida kao područja računalne znanosti trenutnog stanja istraživanja u području detekcije registarskih pločica i prepoznavanja znakova. Naposljetku, obuhvaćene su različite metode i tehnike dubokog učenja koje su dosad korištene, kao i njihove prednosti i ograničenja. Posebna pažnja posvećena je YOLO algoritmu, koji se istaknuo kao izuzetno moćan i brz alat za detekciju objekata u slikama. Nakon toga detaljno će se opisati cjelokupni proces korištenja YOLO algoritma i skupa algoritama potrebnih za odrađivanje detektiranih registarskih pločica i prepoznavanje znakova na istima u programskom jeziku Python.

Konačno, zaključak rada će sažeti postignute rezultate istraživanja i izvući važne zaključke o primjeni YOLO algoritma u detekciji registarskih pločica i prepoznavanju znakova.

Nadam se da će ovaj rad pružiti korisne informacije o primjeni, učinkovitosti i mogućnostima korištenih tehnika i algoritama u različite svrhe, te da će poslužiti kao smjernica za daljnje istraživanje i primjenu računalnog vida u području autonomne vožnje.



## **Sažetak**

U ovom radu istražena je tema detekcije registarskih oznaka na motornim vozilima koristeći neuronsku mrežu YOLO v7, kao i prepoznavanje znakova na istoj pločici uz pomoć EasyOCR-a. Razmotreni su različiti parametri učenja koji mogu utjecati na konačni rezultat. Proveden je proces treniranja, validacije i testiranja te su iznijeti rezultati evaluirani. Također, definirani su ključni pojmovi računalnog vida. Ovaj rad pruži i pregled ostalih sustava za detekciju objekata kako bi se identificirale prednosti i nedostaci te se navode ključne razlike, s ciljem stjecanja uvida u način rada detekcijskih sustava i metode koje koriste.

Ključne riječi: računalni vid, klasifikacija, lokalizacija, sustavi za detekciju, duboko učenje, konvolucijske neuronske mreže.

## **Summary**

This paper explores the topic of license plate detection on motor vehicles using the YOLO v7 neural network, as well as character recognition on the same plate with the assistance of EasyOCR. Various learning parameters that may influence the final outcome have been examined. The process of training, validation, and testing was carried out, and the results were presented and evaluated. Additionally, key concepts of computer vision were defined. This paper also provides an overview of other object detection systems to identify strengths and weaknesses, and highlights key differences, with the aim of gaining insight into the functioning of detection systems and the methods they employ.

Keywords: computer vision, classification, localization, detection systems, deep learning, convolutional neural networks.



## Sadržaj

Uvod.....	1
1. Računalni vid – detekcija objekata .....	3
2. Duboko učenje – neuronske mreže .....	6
2.1.1. Nadzirano duboko učenje.....	6
2.1.2. Djelomično nadzirano učenje.....	7
2.1.3. Nenadzirano učenje.....	8
3. Obrada slike .....	10
4. ANPR sustavi.....	11
4.1. Gdje se koristi ANPR tehnologija? .....	11
4.2. Prednosti ANPR-a.....	14
4.3. Vrste sustava ANPR-a .....	15
4.4. Kako funkcionira ANPR? .....	17
4.5. ANPR tehnike .....	18
4.5.1. Detekcija objekata u stvarnom vremenu .....	18
4.5.2. Obrada slike .....	19
4.5.3. Optičko prepoznavanje znakova (OCR) .....	19
4.5.4. Podudaranje uzoraka .....	21
5. OpenCV .....	22
5.1. Primjena i korist OpenCV-a.....	22
6. Mjere za usporedbu metoda za detekciju objekata .....	25
6.1. Klasifikacijske metrike.....	25
6.2. Mjere detekcije objekata .....	28
6.3. Baze za učenje.....	32

6.3.1. Pascal VOC 07/12 .....	32
6.3.2. MS-COCO .....	32
6.3.3. ILSVRC .....	33
6.3.4. Open Image .....	34
6.4. Primjer usporedbe modela .....	35
7. YOLOv7.....	41
8. Praktični zadatak .....	43
8.1. Prikupljanje skupa podataka i anotacija.....	43
8.2. Arhitektura modela i treniranje .....	44
8.3. Testiranje rezultata treninga.....	45
8.4. Evaluacija performansi treniranog modela .....	47
8.5. Prepoznavanje znakova na registarskoj oznaci .....	49
9. Zaključak.....	52
10. Literatura .....	53
11. Popis slika .....	56
12. Popis tablica .....	59

## Uvod

Zbog sve veće integracije informacijske tehnologije u mnoge aspekte modernog života, postoji rastuća potražnja za sustavima za obradu podataka o vozilima. Takvi sustavi zahtijevaju obradu i arhiviranje podataka ili od strane čovjeka (što je neefikasno) ili posebnog algoritma/sustava koji je u stanju prepoznati vozila po njihovim registarskim pločicama. Kao rezultat toga, razvijeno je više takvih sustava.

Vozila na cesti se obično prepoznaju po brojevima registarskih pločica, koje su jednostavne za čitanje ljudima, ali teške za strojeve. Registarska pločica, za računalo nije ništa više od skupina piksela unutar područja slike s određenim intenzitetom i svjetlinom. Zbog toga je bitno stvoriti snažan matematički sustav sposoban razumjeti i izvući željene informacije iz prikupljene slike [1].

Ovakve sustave obrade digitalne slike, u užem smislu prepoznavanja automobilskih registarskih pločica nazivamo "ANPR sustavi", engl. *Automatic Number Plate Recognition* ili također „LNPR“, engl. *Licence Number Plate Recognition* i obuhvaća cjelokupni proces transformacije između stvarnog okruženja i informacijskih sustava koji trebaju pohranjivati i upravljati prikupljenim informacijama.

Ovi sustavi su temelj istraživanja u nekim područjima kao što su umjetna inteligencija, računalni vid, prepoznavanje uzoraka i neuronskih mreža. Sustavi automatskog prepoznavanja registarskih pločica su kombinacija hardvera i softvera za obradu signala iz kamere, koji se pretvara u grafički prikaz kao statične slike ili njihove sekvence (videozapisi). Temeljni hardver ovih sustava sastoji se od kamere, slikovnog procesora, memorije za evidenciju događaja i jedinice za pohranu [2].

Kvaliteta algoritama koji se koriste za prepoznavanje registarskih pločica i kvaliteta slikovne tehnologije, kao što su kamera i osvjetljenje, dvije su ključne komponente ovih sustava. Što je lošija kakvoća slike, to će računalo teže odraditi zadatke koje mu zadamo nad istom slikom. Kod ovakvih sustava, treba uzeti u obzir sljedeće čimbenike: maksimalnu točnost prepoznavanja, što kraće vrijeme obrade, rukovanje sa što je moguće više različitih tipova registarskih pločica, upravljanje najrazličitijim kvalitetama slike i postizanje maksimalne tolerancije na izobličenje ulaznih podataka. Sve su to komponente

koje su važne, kako bi rezultat, odnosno točnost prepoznavanja registarskih pločica bio na najvišoj mogućoj razini.

Napredne računalne aplikacije obrađuju više od tekstualnih podataka rješavajući svakodnevne probleme. Ulazi s optičkih uređaja koriste se u primjenama domene kao što su medicina, sigurnost, nadzor te inženjering. Sposobnost računala da obradi sliku i prevede je u nešto smisleno postala je sve popularnija. Stoga je tehnologija obrade slike usvojena u upravljanju sustavom parkiranja vozila, pristupu vozila ograničenom području, sustavu praćenja prometa i elektroničkoj naplati cestarine na autocestama. U tu svrhu računalo treba uhvatiti registarsku pločicu vozila i obraditi je u računalu [3].

# 1. Računalni vid – detekcija objekata

Računalni vid se odnosi na područje računalne znanosti koje proučava načine na koje računala mogu analizirati i interpretirati slike i videozapise. Cilj računalnog vida jest razviti algoritme i sustave koji omogućuju računalima da "vide" i razumiju svijet oko sebe.

Jedna od najznačajnijih primjena računalnog vida jest detekcija i prepoznavanje objekata. To omogućuje računalima da analiziraju okolinu i prepoznaju objekte poput ljudi, životinja, vozila, znakova i drugih predmeta. Detekcija i prepoznavanje objekata su ključne funkcije u mnogim područjima poput virtualne i proširene stvarnosti, robotike, medicinske dijagnostike, sigurnosnih sustava, daljinskih istraživanja i autonomnih vozila.

Napredak u računalnom vidu posljednjih godina doveo je do razvoja novih i poboljšanih algoritama za detekciju i prepoznavanje objekata. U ranijim fazama detekcija objekata građena je na ručno izrađenim algoritmima prepoznavanja značajki poput Viola-Jones detektora, histograma orijentiranih gradijenata (HOG) i slično. Mada su oni uspijevali rješavati zadatke detekcije objekata, to su radili sporo, neprecizno i loše su se nosili s nepoznatim skupovima podataka [4]. Zbog toga se u zadnjih desetak godina uvode konvolucijske neuronske mreže (CNN, engl. *Convolutional neural network*) – odnosno duboko učenje za klasifikaciju slika. Također, dostupnost senzora i kamera visoke kvalitete te povećanje računalne snage danas omogućuju brže i preciznije otkrivanje i prepoznavanje objekata u stvarnom vremenu. Sve to otvara nove mogućnosti u različitim industrijama i omogućuje razvoj inovativnih rješenja koja se temelje na računalnom vidu.

## 1.1. Detekcija objekata - postavljanje zadatka (problema)

Detekcija objekata je logičan nastavak na klasifikaciju objekata, koja se usredotočuje samo na prepoznavanje objekata na slici. Glavni cilj detekcije objekata je otkrivanje svih instanci predefiniраниh klasa (npr. „auto“, „avion“, „mačka“ itd.) i pružanje njihove približne lokacije na slici pomoću okomitih pravokutnika. Detektor treba prepoznati sve instance objekata i označiti ih okvirom. Ovaj problem se obično tretira kao problem nadziranog učenja, gdje moderni modeli detekcije objekata imaju pristup velikim skupovima označenih slika za trening i ocjenjuju se na raznim referentnim skupovima podataka [4].

## 1.2. Računalni vid i ljudski vid

Računalni vid i ljudski vizualni sustav imaju neke sličnosti, ali i mnogo razlika. Dok ljudski vizualni sustav ima sposobnost prepoznavanja objekata u realnom vremenu, razlikovanja boja i oblika, računalni vid koristi algoritme i matematičke modele za obradu slike i prepoznavanje oblika.

Ljudski vizualni sustav također ima mogućnost zaključivanja iz konteksta i znanja o svijetu oko nas, dok računalni vid koristi strogo definirana pravila za prepoznavanje i analizu slike. Računalni vid također ima sposobnost obraditi velike količine podataka brzo i točno, što ljudski vizualni sustav ne može.

Unatoč razlikama, računalni vid ima sve više primjena u različitim područjima, kao što su medicina, robotika, sigurnost i promet. Uz napredak u tehnologiji računalni vid postaje sve sofisticiraniji i bliži sposobnostima ljudskog vizualnog sustava.

Ljudski vizualni sustav je vrlo složen i uključuje obradu vizualnih informacija u mozgu. On koristi obje oči da bi dobili binokularni vid i stvorili 3D prikaz svijeta. Ljudski vid ima sposobnost prilagodbe različitim uvjetima osvjetljenja i kontrastu, te ima sposobnost razlikovanja milijuna različitih boja.

S druge strane, računalni vid se temelji na obradi digitalnih slika koje prikupljaju senzori kamere. Slike se obrađuju korištenjem matematičkih algoritama za otkrivanje oblika, boja, i drugih značajki slike. Za razliku od ljudskog vida, računalni vid ne može prilagoditi se različitim uvjetima osvjetljenja i kontrastu, te nije u stanju razlikovati toliko nijansi boja.

Iako računalni vid nije još uvijek na razini ljudskog vida u pogledu složenosti i sposobnosti, on se ubrzano razvija i sve više se koristi u različitim industrijskim i znanstvenim aplikacijama. U budućnosti, računalni vid će vjerojatno postati još napredniji i sposoban za složenije zadatke koji zahtijevaju visok stupanj preciznosti i brzine.

Ovaj rad ima za cilj istražiti uporabu detekcije i prepoznavanja objekata u stvarnom vremenu korištenjem OpenCV-a, s naglaskom na primjene u autonomnoj vožnji. Rad će obuhvatiti različite aspekte detekcije i prepoznavanja objekata u stvarnom vremenu, uključujući tehnike za detekciju i praćenje objekata, metode za prepoznavanje objekata te integraciju sustava za detekciju i prepoznavanje objekata s autonomnim sustavom vožnje.



Računalni vid i ljudski vizualni sustav imaju neke sličnosti, ali i mnogo razlika. Dok ljudski vizualni sustav ima sposobnost prepoznavanja objekata u realnom vremenu, razlikovanja boja i oblika, računalni vid koristi algoritme i matematičke modele za obradu slike i prepoznavanje oblika.

Ljudski vizualni sustav također ima mogućnost zaključivanja iz konteksta i znanja o svijetu oko nas, dok računalni vid koristi strogo definirana pravila za prepoznavanje i analizu slike. Računalni vid također ima sposobnost obraditi velike količine podataka brzo i točno, što ljudski vizualni sustav ne može. Unatoč razlikama, računalni vid ima sve više primjena u različitim područjima, kao što su medicina, robotika, sigurnost i promet. Uz napredak u tehnologiji računalni vid postaje sve sofisticiraniji i bliži sposobnostima ljudskog vizualnog sustava, no već sad u mnogočemu prednjači nad ljudskim vidom.

## 2. Duboko učenje – neuronske mreže

Duboko učenje (engl. *Deep learning*) područje je strojnog učenja koje se bazira na treningu složenih modela umjetne inteligencije, zvanih duboke neuronske mreže (engl. *Deep neural network*) da kroz mnogo iteracija i ulaznih podataka (u ovom slučaju slika/videozapisa) nauče specifične karakteristike određenog skupa podataka [5].

Duboko učenje kao ulazne parametre koristi velike skupove podataka i složene algoritme optimizacije kako bi treniralo duboke neuronske mreže. Tijekom treninga, modeli se prilagođavaju podacima tako da automatski otkrivaju predodređene relevantne značajke i obrasce iz podataka, bez potrebe za ručnim određivanjem značajki. Glavna prednost dubokog učenja je sposobnost da iz velikih količina neobrađenih podataka nauči prepoznati ono što mu je zadano – bilo to prepoznavanje objekata, klasifikacija slika, prepoznavanje uzoraka, generiranje sadržaja i sl [6].

Tradicionalna metoda detekcije objekata se temeljila na ručno izrađenim karakteristikama, ali nije bila otporna na promjene osvjetljenja i nije pokazivala dobru sposobnost generalizacije. Također je prepoznato da je napredak u detekciji objekata bio vrlo spor tijekom 2010-2012. godine u PASCAL VOC izazovu [6].

Duboko učenje se može klasificirati u 3 pristupa: nadzirani, polu-nadzirani ili djelomično nadzirani i nenadzirani pristup, i duboki modeli pojačanog učenja kao spoj dubokih modela i pojačanog učenja.

### 2.1 Vrste dubokog učenja

#### 2.1.1. Nadzirano duboko učenje

Nadzirano duboko učenje (engl. *Deep supervised learning* – DSL) je metoda dubokog učenja gdje se koriste označeni podaci (od strane čovjeka) – ovaj pristup je također korišten kasnije u ovom radu. U ovakvom modelu učenja, okruženje ima skup ulaza (ulazna slika + označeno područje interesa, npr. čovjek na slici) i odgovarajućih izlaza. Zatim osoba koja nadzire učenje može modificirati mrežne parametre za bolju aproksimaciju željenih izlaza. Nakon uspješnog treniranja, možemo dobiti točne rezultate za određenu ulaznu sliku bez označenog područja, npr. naučeni algoritam može prepoznati čovjeka na slici [7].

Postoje različiti pristupi nadziranog učenja za duboko učenje, uključujući duboke neuronske mreže (DNN, engl. Deep neural network), konvolucijske neuronske mreže (CNN), ponavljajuće neuronske mreže (RNN, engl. Recurrent neural network).

### 2.1.2. Djelomično nadzirano učenje

Djelomično nadzirano učenje (SSL, engl. *Semi-supervised learning*) je vrsta dubokog učenja gdje kao ulaz imamo samo nekoliko označenih primjera dostupno (često između 1% i 10% cijelog skupa podataka), zajedno s velikim brojem neoznačenih uzoraka. SSL metode su primjenjive na aplikacije u stvarnom svijetu gdje su neoznačeni podaci lako dostupni i lako ih je nabaviti (npr. slike na kojima se nalaze ljudi/auti/semafori i slično), dok je označene uzorke često teško nabaviti, te su skupi i vrijeme za prikupljanje podataka je često nedostupno. SSL je kroz učenje sposoban izgraditi bolje klasifikatore koji će naposljetku nadoknaditi nedostatak označenih podataka u treningu. Međutim, jasno je da moramo biti svjesni primjene određenog modela – u SSL modelu dobivamo na brzini pokretanja cijelog sustava (vrijeme prikupljanja podataka je značajno smanjeno), troškovi su niži, no model često daje lošije rezultate od prethodno navedenog nadziranog modela dubokog učenja [8].

Neke metode i algoritmi za SSL su:

- Trening dosljednosti – temelji se na metodi *pretpostavke glatkoće* (engl. *smoothness assumption*) gdje se pretpostavlja da podatkovne točke (pikseli) koje su blizu jedna drugoj vjerojatno će imati istu oznaku (engl. *label*)
- Metode zamjenskih oznaka (engl. *proxy*) - takve metode koriste već uvježbani model nad označenim skupom za proizvodnju dodatnih primjera za trening označavanjem instanci neoznačenog skupa na temelju nekog predoznačenog modela.
- Metode temeljene na grafovima – Označene i neoznačene podatkovne točke mogu se smatrati čvorovima grafa, a cilj je približiti oznake od označenih čvorova do neoznačenih koristeći sličnost dvaju čvorova na grafu. Često se odstupanja na grafu reguliraju Laplaceovim regulatorom grafa.

### 2.1.3. Nenadzirano učenje

Sustavi za duboko učenje bez nadzora su sustavi koji se mogu učiti bez prisustva predodređenih značajki. Sustav pokušava ulazne podatke analizirati i grupirati tako da otkriva određeni uzorke ili grupacije sličnih podataka, odnosno zajedničke karakteristike [9].

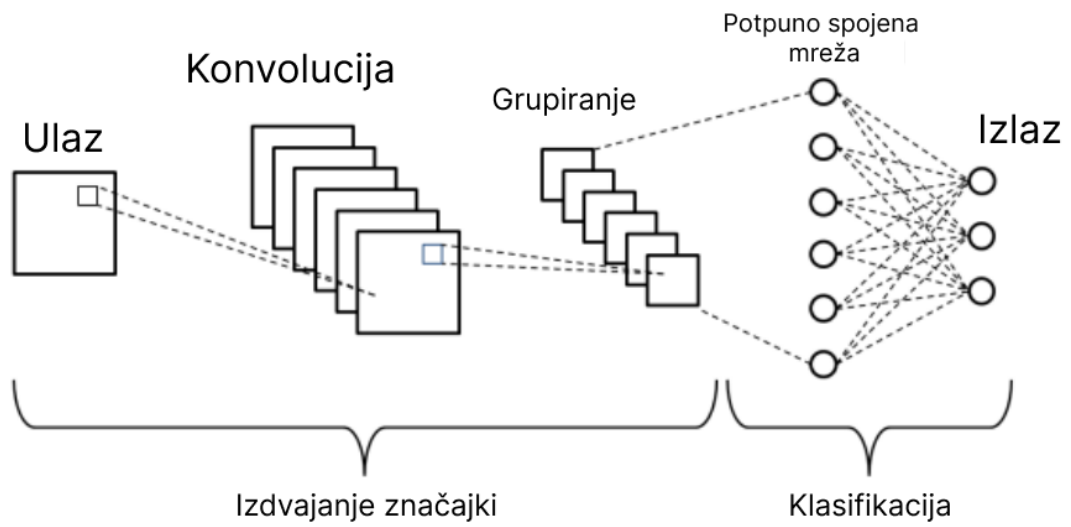
## 2.2. Konvolucijska neuronska mreža - CNN

Konvolucijska neuronska mreža (engl. *Convolutional neural network*) pruža sveobuhvatni model učenja u kojem se model može trenirati pomoću metode gradijentnog spusta. U dubokom učenju, CNN-ovi su najčešće mreže koje se koriste za klasifikaciju slika. Značajke svakog sloja (engl. *layer*) na slici generiraju se iz lokalnog područja prethodnog sloja (prijemnog polja) dijeljenjem težine konvolucijske jezgre [6].

Treniranje duboke neuronske mreže zahtijeva veliki skup podataka specifičan za praćenje značajki. Konvolucijskim neuronskim mrežama smanjujemo iteracije treninga i kompleksnost čitavog modela. CNN modeli mogu izvući značajke iz slike, tako da nema potrebe za metodom izdvajanja značajki.

Slika 2.1. prikazuje jednostavnu shematsku reprezentaciju nekog osnovnog CNN-a. Mreža na slici se sastoji od 5 slojeva: ulaznog sloja, konvolucijskog sloja, grupacijskog sloja, sloja potpuno spojene mreže i izlaznog sloja. Slojevi su podijeljeni u dva skupa: Izdvajanje značajki i klasifikacija.

Ulazni sloj određuje fiksnu veličinu za ulazne slike, čija se veličina može promijeniti ako je potrebno. U dijelu izdvajanja značajki kroz slojeve konvolucije i grupiranja prolazi se više puta prije no što se dođe do dijela klasifikacije. U grupacijskom sloju se smanjuje veličina slike, uz zadržavanje sadržanih informacija. Nakon što se prođe konvolucija i grupiranje, izdvojene značajke se spajaju u potpuno spojenu mrežu, iz koje izlazi rezultat klasifikacije slike.



**Slika 2.1.** Shematski prikaz arhitekture jednostavne konvolucijske neuronske mreže (Phung & Rhee, 2019)

### 3. Obrada slike

Nakon što iz kamere sliku učitamo na računalo, nad njom se najčešće, u svrhu aplikacije računalnog vida vrše razne operacije, takozvana obrada/predobrada slike (engl. *image processing/preprocessing*). Koraci koji se vrše nad ulaznom slikom omogućuju računalu lakšu analizu podataka – recimo, u mnogo slučajeva nije nam od važnosti boja na slici, pa ćemo određenim koracima pretvoriti sliku u crno-bijelu, što je računalu pogodno, zato što radi s manje podataka. Pokušavamo izvući podatke iz ulazne slike koji su potrebni računalu za određen zadatak u računalnom vidu, a otkloniti one suvišne. Predobrada slike je izuzetno važna u aplikacijama računalnog vida koji zahtijevaju obradu u pravom vremenu (ili što bliže pravom vremenu). Budući da su računala određena, pa i ograničena svojom računalnom snagom, potrebno je sliku najprije predobraditi kako bi postigli što veću ekonomičnost u smislu resursa koje računalo koristi (memorija i procesorska snaga) [10].

U većini aplikacija računalnog vida, koriste se neke zajedničke bazne operacije obrade slike. Neke od najčešće korištenih operacija za obradu slike su:

- Promjena veličine slike (engl. *resizing*) – često je potrebno promijeniti veličinu slike da odgovara potrebnoj veličini za određene daljnje zahtjeve obrade i kako bi se ujednačila dimenzija svih slika u skupu za određeni zadatak.
- Obrezivanje slike (engl. *crop*) – koristi se za uklanjanje nepotrebnih dijelova slike ili za fokusiranje na određeno željeno područje.
- Uklanjanje šuma (engl. *noise reduction*) – šum u fotografskom smislu označava pojam za neželjenu pojavu slučajne varijacije svjetlosti ili boje na slici. Ova pojava najčešće se javlja zbog nesavršenosti samih fotografskih uređaja – previsok ISO (koji se možda koristio zbog nedostatka svjetlosti na slici), loša kvaliteta senzora, duga ekspozicija i sl. Za uklanjanje šuma može se koristiti niskopropusni filter kao što je Gaussovo zamućivanje slike (engl. *Gaussian blur*).
- Binarna segmentacija: segmentacija pozadine od prednjeg dijela slike (subjekta ili objekta u fokusu).

## 4. ANPR sustavi

Automatsko prepoznavanje automobilskih pločica ili ANPR je vrlo precizna tehnologija koja može čitati registracije na automobilu bez direktne ljudske pomoći. Ovaj sustav koristi optičko prepoznavanje znakova (engl. „*Optical Character Recognition*“) nad slikama vozila za čitanje registarskog broja vozila. Sustav primjenjuje različite tehnike obrade slike za brzu i automatsku identifikaciju vozila na slikama ili videozapisu u stvarnom vremenu s jednom ili više kamera [11].

### 4.1. Gdje se koristi ANPR tehnologija?

Sustav automatskog prepoznavanja registarskih pločica može se koristiti u različite svrhe, kao što je praćenje kretanja vozila, identifikacija određenih automobila, automatizirano parkiranje itd. Upotreba ANPR sustava postaje sve popularnija kako tehnologija brzo napreduje s pojavom strojnog i dubinskog učenja; troškovi računanja se smanjuju, a točnost primijenjenih tehnika obrade slike raste [2].

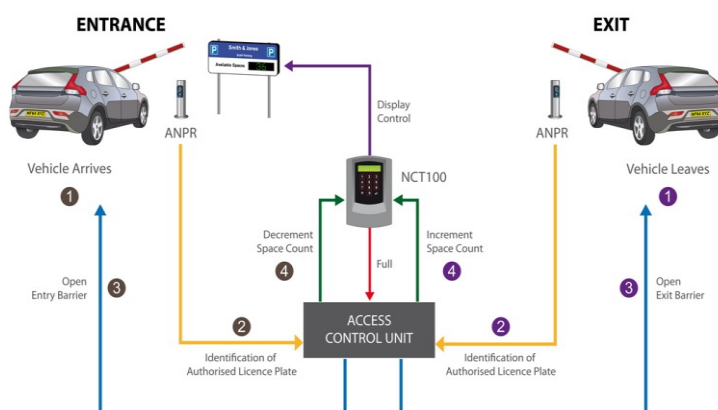
- **Policija:** Policijske uprave koriste ANPR za identifikaciju vozila koja sudjeluju u prometnim prekršajima, za provjeru je li vozilo registrirano, te za provedbu zakona. Vlasti mogu identificirati vozila i odrediti njihovu lokaciju zahvaljujući sposobnosti otkrivanja i prepoznavanja registarskih pločica u stvarnom vremenu. Kako bismo pomogli u učinkovitosti u našem radu s prijevozom i infrastrukturom koja se temelji na vozilima, kao i za podršku sigurnosti i sigurnosti opće javnosti, ANPR se koristi na različite načine. ANPR je tehnologija koja pomaže u provođenju zakona, a istodobno osigurava stalnu sigurnost svih sudionika u prometu (naravno, ovisno o tome kako određena vlast koristi svoje ovlasti u sklopu ovakvih sustava). Sustav napreduje u poboljšanju učinkovitog kretanja ljudi u stvarnom vremenu, pomažući u sigurnosti pojedinaca diljem svijeta [12]. Primjer sustava za prepoznavanje registarskih pločica automobila je vidljiv na Slici 4.1.



**Slika 4.1.** Prepoznavanje registarskih pločica automobila, izvor:

<https://www.skyfilabs.com/project-ideas/number-plate-scanner-using-matlab>

- Pametno upravljanje parkingom:** Za prepoznavanje određenih automobila, upravljanje parkiralištem zahtijeva integrirani sustav. Automatsko upravljanje parkingom u garaži omogućuje ANPR jer je svako vozilo identificirano brojem registarske pločice. Ovi sofisticirani sustavi upravljanja parkiranjem prate svako vozilo unutar zgrade i pružaju potpunu kontrolu. Brigu o održavanju vlastitih karata, evidenciji utrošenog vremena te opasnosti od kažnjavanja zbog netočne uplate karte ili zagubljenog mjesta korisnici garaže stoga izbjegavaju. U slučaju spora, može se čak vratiti na automatizirani nadzor. Primjer pametnog upravljanja parkingom prikazan je na Slici 4.2.





**Slika 4.2.** Način rada ANPR-a na parkiralištu, izvor:

<https://www.nortechcontrol.com/solutions/vehicle/vehicle-access-with-counting-using-anpr>

- **JTA (engl. „Journey time analysis“):** Program koji pomaže vlastima u praćenju prolaska automobila i vremena koje im je potrebno da prijeđu između dva čvora. Uvidi pomažu upraviteljima prometa da učinkovitije dizajniraju rute.
- **Inteligentno upravljanje prometom:** Automatsko otkrivanje vozila može se koristiti u gradovima za prepoznavanje vozila koja voze prebrzo, neoprezno ili na bilo koji drugi način. Podaci o prometu područja ili cijelog grada mogu se mjeriti i analizirati pomoću ANPR sustava. Sposobnost razumijevanja prometnih zagušenja u većoj mjeri omogućuje poboljšano planiranje prometa (Slika 4.3.).



**Slika 4.3.** Inteligentno upravljanje prometom, izvor: <https://viso.ai/computer-vision/automatic-number-plate-recognition-anpr/>

- **Sigurnost maloprodajnog parka:** Neovlašteno parkiranje često je problem u maloprodajnim parkovima, što uzrokuje probleme s legitimnim parkirnim mjestima ili povremeno sumnjivo ponašanje. Uz ANPR tehnologiju, ti se sigurnosni rizici mogu smanjiti jamstvom da samo ovlaštena vozila pristupaju parkirnim mjestima.

- **Evidencija naplatnih kućica:** U nekoliko regija svijeta ručno upravljanje naplatnim kućicama još uvijek je uobičajena praksa na autocestama, recimo u Hrvatskoj dok pišem ovaj rad, još uvijek se većinski manualno naplaćuju cestarine, iako je dijelom moguće platiti karticom ili ENC sustavom, što je poduprieto prepoznavanjem tablica. Automatizirane naplatne kućice često koriste različite tehnologije za autonomno upravljanje naplatnim kućicama. Na primjer, ANPR omogućuje vlastima prikupljanje brojeva registarskih pločica za plaćanje cestarine putem pošte ili automatski na većim autocestama umjesto zaustavljanja i plaćanja na naplatnim kućicama kojima upravljaju ljudi. Kao rezultat toga, ANPR omogućuje učinkovitu administraciju naplatnih kućica, dinamičko određivanje cijena cesta i smanjenje operativnog vremena, što sve povećava produktivnost [13].

## 4.2. Prednosti ANPR-a

Brojne prednosti ANPR-a služe kao temelj za praktične primjene. Automatizacija ručnih poslova, iznimno učinkovito upravljanje prostorom, poboljšano upravljanje i poboljšano korisničko iskustvo glavne su prednosti ANPR-a.

- **Automatizacija:** Automatizirano prepoznavanje registarskih pločica omogućuje automatizirana upozorenja i kontrole objekata. Stoga je ANPR ključna tehnologija za pametne gradove ili naplatne postaje.
- **Preciznost:** ANPR sustavi mogu postići vrlo visoku točnost i mogu brzo i jednostavno identificirati vozila prema njihovim registarskim pločicama.
- **Analitika:** Generirani podaci mogu se koristiti za analizu protoka prometa. Ovo je posebno važno za upravljanje inteligentnim transportnim sustavima, gdje se tehnologije obrade podataka koriste za poboljšanje mobilnosti ljudi i dobara, upravljanje potražnjom, povećanje sigurnosti, smanjenje prometnih gužvi i učinkovito upravljanje incidentima.
- **Identifikacija:** Brzo prepoznavanje registarske pločice vozila osnova je za brzu i besprijekornu identifikaciju vozila. Identifikacija se može koristiti za odobravanje pristupa vozilima ili pronalaženje i praćenje određenih vozila.

- **Isplativost:** Precizno i brzo prepoznavanje registarskih pločica ne oslanja se na ljudski unos. Velik broj registarskih pločica može se analizirati vrlo brzo, što ga čini vrlo učinkovitim metodom za identifikaciju vozila. Stoga pokreće troškovno učinkovito upravljanje i smanjuje vrijeme čekanja.
- **Praktičnost:** ANPR je obično integriran s drugim IT sustavima i radi u ekosustavu kako bi krajnjim korisnicima pružio besprijekorno iskustvo bez muke. Stoga se tehnologija koristi za poboljšanje korisničkog iskustva i ponudu novih usluga i proizvoda, poput automatiziranog plaćanja parkiranja.
- **Raznovrsnost:** automatizirana identifikacija vozila može se primijeniti za različite primjene, od upravljanja parkiranjem do sigurnosti, nadzora prometa, optimizacije logistike u proizvodnji itd.
- **Sigurnost:** ANPR rješenja su od velike važnosti za razne sigurnosne i nadzorne primjene računalnog vida. Takvi sustavi pomažu u poboljšanju sigurnosti pružajući metodu za autonomnu identifikaciju i praćenje više vozila.
- **Prednosti za okoliš:** korištenje ANPR sustava može pomoći u smanjenju prometnih gužvi i zagađenja obeshrabrujući nepotrebnu vožnju. Smanjenje vremena čekanja, odnosno vremena traženja parkirnih mjesta u gradovima pomaže u smanjenju zagađenja okoliša [14].

### **4.3. Vrste sustava ANPR-a**

Postoje dvije osnovne vrste ANPR sustava:

- **Fiksni sustavi:** Fiksni ANPR sustavi su oni koji su instalirani na određenoj lokaciji i u određenom trenutku. Na primjer, ti se sustavi mogu instalirati na naplatnim kućicama, graničnim prijelazima ili drugim strateškim mjestima. Kamere se postavljaju na fiksno mjesto, poput semafora, telefonskog stupa, ulaza u zgradu ili izlazne rampe s autoceste. Ove kamere obično snimaju samo pokretne objekte koji prolaze u njihovom vidnom polju. Lokacija i brzina automobila mogu se odrediti postavljanjem brojnih stacionarnih ANPR kamera duž jedne ceste. Vlada može utvrditi da vozač vjerojatno živi ili radi u blizini ako se podaci čuvaju tijekom vremena i mogu se koristiti za određivanje svaki put kad je određena registarska pločica prošla određeno mjesto.

Povremeno je moguće pomicanje stacionarnih kamera. Na primjer, ANPR uređaji mogu se instalirati u kombije za nadzor ili prikolice kamiona, koji se zatim mogu postaviti na ključna mjesta kao što su oružarske predstave ili političke demonstracije [9]. ANPR kamere često se koriste s automatiziranim sustavima za kontrolu brzine i crvenog svjetla, kao i za naplatu cestarina na cestama i mostovima, prikazano na Slici 4.4.



**Slika 4.4.** Stacionarne ANPR kamere, izvor:

<https://pailypost.com/2022/10/24/council-oks-cameras-that-read-license-plates-to-stop-crime/>

- Mobilni sustavi: Mobilni ANPR sustavi su oni koji se nalaze u vozilu i mogu se koristiti za skeniranje registarskih drugih vozila. Na primjer, ove sustave mogu koristiti policijski automobili za pronalaženje traženih vozila. Kamere se često postavljaju na policijska patrolna vozila, omogućujući policajcima prikupljanje informacija s registarskih tablica dok se kreću gradom tijekom svojih smjena. Te se kamere obično uključuju na početku smjene i ostavljaju uključene do kraja smjene. Osim toga, privatne tvrtke poput Vigilant Solutions koriste mobilne ANPR-ove za uzimanje podataka o tablicama i prodaju ih organizacijama za provođenje zakona i drugim stranama. Mobilne ANPR kamere učinkovite su u snimanju registarskih tablica parkiranih automobila osim snimanja vozila u prolazu [15]. Na primjer,

patrolno vozilo može krstariti oko javnog parkirališta i brzo zabilježiti registarske pločice stotina automobila, kao što je prikazano na Slici 4.5.



**Slika 4.5.** Mobilne ANPR kamere, izvor: <https://www.quora.com/Do-police-ever-scan-number-plates-of-cars-going-into-McDonalds-drive-throughs-to-locate-criminals>

#### **4.4. Kako funkcionira ANPR?**

ANPR je tehnologija koja automatski očitava registracijske pločice automobila, omogućujući usporedbu tih podataka sa zapisima baze podataka.

ANPR kamera snima sliku registarske pločice vozila; slika se prosljeđuje čitaču koji locira automobilsku pločicu i čita slova i znakove tako da se mogu identificirati pomoću OCR softvera. Slika pogleda registarske pločice i/ili kamere za pregled se zatim sprema i provjerava u bazi podataka.

ANPR podaci sastoje se „read“ i „hit“ podataka. Očitavanje je snimanje automobilske pločice i slike vozila dok ono prolazi kroz prikaz kamere; očitani podaci su izraz koji se koristi za opisivanje svih podataka prikupljenih dok vozila prolaze kroz ANPR čitač.

Pogodak je podudaranje s automobilskom pločicom unutar baze podataka koja se pretražuje [16].

## **4.5. ANPR tehnike**

Popularne ANPR metode uključuju nekoliko različitih pristupa koji se kombiniraju kao dio računalnog vida.

### **4.5.1. Detekcija objekata u stvarnom vremenu**

Koristi se duboko učenje za identifikaciju vozila i drugih klasa vozila (autobus, kamion, automobil, kombi, motocikl itd.). Suvremeni algoritmi za otkrivanje objekata poput YOLOv3 i YOLOv7 koriste neuronske mreže koje su usavršene korištenjem zbirke fotografija.

Jedan od ključnih problema koji se rješava računalnim vidom je detekcija objekata. Služi kao temelj za mnoge druge zadatke računalnog vida koji dolaze kasnije, uključujući praćenje objekata, opise slika, segmentaciju slika i još mnogo toga. Primjene za detekciju specifičnih objekata uključuju detekciju pješaka, identifikaciju životinja, detekciju vozila, detekciju teksta, detekciju poze, prepoznavanje registarskih oznaka i dr.

Zamah tehnologije detekcije objekata dramatično se ubrzao posljednjih godina zbog ogromnog napretka u tehnikama dubokog učenja. Mreže dubokog učenja i računalni kapacitet grafičke kartice - GPU-a (engl. *graphic processing unit*), značajno su povećali performanse detektora i praćenja objekata, što je dovelo do važnih napretka u detekciji objekata (Slika 4.6.).



**Slika 4.6.** Detekcija objekata za vozila temeljena na dubokom učenju, izvor:

<https://viso.ai/edge-ai/google-coral/>

#### **4.5.2. Obrada slike**

Tradicionalne tehnike računalnog vida koriste se u obradi slike kako bi se slike standardizirale i pripremile za OCR algoritam. Funkcije obrade slike koriste se za izoštravanje, ispravljanje boja ili obrezivanje slika kako bi se značajno poboljšali rezultati i izlaz naknadnih algoritama jer se ANPR aplikacije obično koriste u teškim okruženjima stvarnog svijeta s različitim osvjetljenjem, vremenskim uvjetima i nedosljednim postavkama. Jedna od najpopularnijih biblioteka za poslove obrade slika u ANPR sustavima je OpenCV.

#### **4.5.3. Optičko prepoznavanje znakova (OCR)**

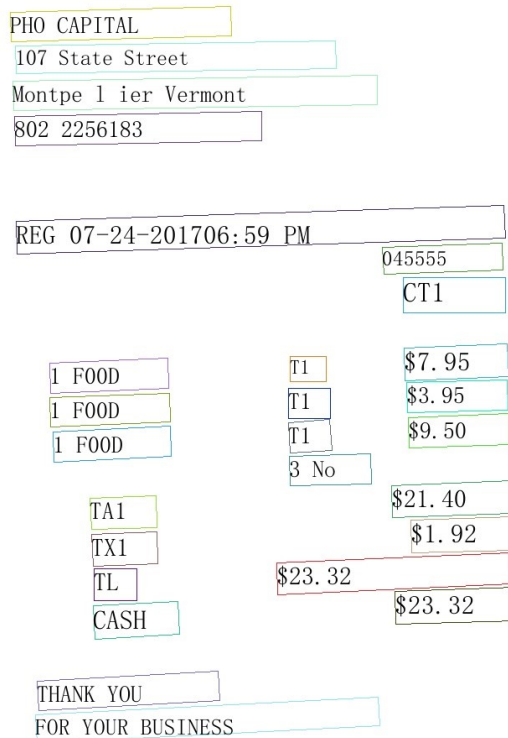
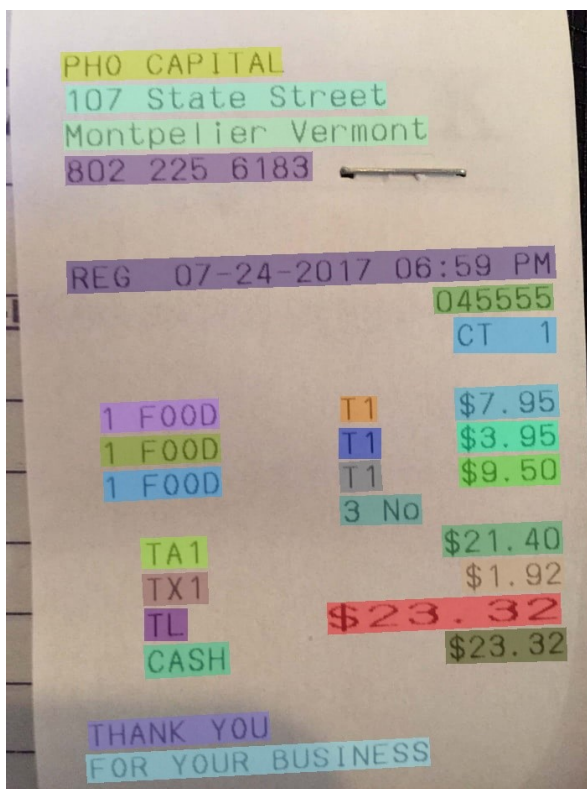
OCR je kratica za „Optical Character Recognition“ i odnosi se na tehnologiju koja se koristi za pretvaranje skeniranih slika u tekstualne datoteke, odnosno prepoznavanje i ispisivanje prepoznatog teksta. OCR je ključna metoda računalnog vida za izdvajanje teksta iz slika. Uz pomoć OCR-a, ANPR sustav može prepoznati registarske pločice. Za identifikaciju pojedinačnih znakova, potvrdu njihovog redoslijeda i prevođenje slike registarske pločice u tekst, koriste se metode obrade slike kao što su „Tesseract OCR“ ili „EasyOCR“ algoritmi, no postoje i mnogi drugi takvi algoritmi – oni koji se plaćaju, i oni besplatni slobodni za korištenje kao što je EasyOCR (korišten u radu).

Godine 1913., dr. Edmund Fournier d'Albe stvorio je „Optophone“ za skeniranje i transkripciju teksta za slijepe osobe. Optofon je uređaj koji je zamišljen za slijepe osobe,

koji skenira tekst i generira vremenski promjenjive akorde tonova za prepoznavanje slova. Taj proces sonifikacije može se nazvati jednim od prvih oblika OCR-a. Od tada je OCR tehnologija prošla kroz nekoliko faza razvoja. Tehnologija je postala poznata u 1990-ima kada su stare novine digitalno sačuvane. Osim toga, razvoj pametnih telefona i elektroničkih dokumenata pomogao je još više unaprijediti OCR tehnologiju [17].

Način na koji OCR funkcioniра je prilično jednostavan. No, njegova implementacija može biti zahtjevna zbog nekoliko čimbenika, poput raznolikosti fontova ili metoda korištenih za oblikovanje slova. Na primjer, implementacija OCR-a može postati eksponencijalno složenija kada se kao unos koriste uzorci nedigitalnog rukopisa umjesto tipkanog pisma.

Cijeli proces OCR-a uključuje niz koraka koji uglavnom sadrže tri cilja: prethodnu obradu slike, prepoznavanje znakova i naknadnu obradu specifičnog izlaza. Još neki zadaci OCR-a uključuju obradu prirodnog jezika NLP (engl. *Natural Language Processing*) ne samo za čitanje, već i za analizu i razumijevanje značenja teksta i govora [18]. Primjer OCR-a je prikazan na Slici 4.7.





**Slika 4.7.** Primjer OCR-a, izvor:

[https://github.com/PaddlePaddle/PaddleOCR/blob/release/2.5/doc/doc\\_en/quickstart\\_en.md](https://github.com/PaddlePaddle/PaddleOCR/blob/release/2.5/doc/doc_en/quickstart_en.md)

#### **4.5.4. Podudaranje uzoraka**

Kako bi se identificiralo podudaranje, tehnika koja se naziva „Template Matching“ uspoređuje sliku registarske pločice sa zbirkom slika registarske pločice. Tehnika podudaranja uzoraka služi za pronalaženje određenog uzorka u tekstualnom nizu. U slučaju ANPR-a, skup znakova su registarske pločice vozila [18].

## 5. OpenCV

OpenCV (Open Source Computer Vision Library) je popularna knjižnica otvorenog koda za primjenu računalnog vida i strojnog učenja, koja nudi razne algoritme i alate za detekciju i prepoznavanje objekata. Izvorno ju je 1999. godine razvio Intel, a otkad je održava i razvija velika zajednica programera - licenca Apache 2 čini ga slobodno dostupnim. OpenCV je i dalje vrlo popularan u 2023. godini, s više od 29.000 preuzimanja tjedno.

OpenCV uključuje širok raspon algoritama za obradu slike i računalni vid, poput filtriranja slika, otkrivanja značajki, detekcije i praćenja objekata, kalibracije kamere te algoritama temeljenih na strojnom učenju za analizu slika i videozapisa. Također podržava različite programske jezike, uključujući C++, Python, Java i MATLAB. Knjižnica OpenCV-a ima više od 2500 algoritama, znatnu dokumentaciju i primjer koda za računalni vid u stvarnom vremenu [19].

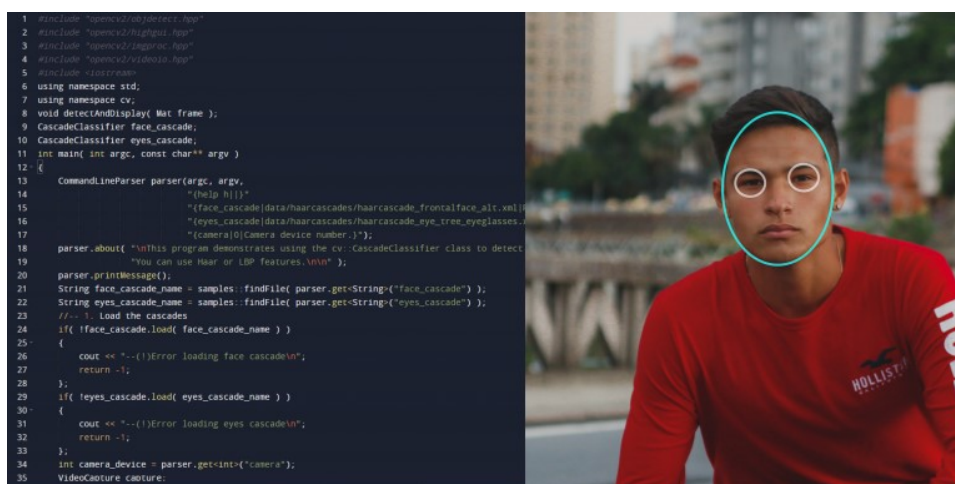
### 5.1. Primjena i korist OpenCV-a

Velike tvrtke i vladine organizacije, kao što su Google, Toyota, IBM, Microsoft, SONY, Siemens i Facebook, koriste OpenCV. Osim njih, poznate tvrtke za računalni vid, kao što je viso.ai, koriste OpenCV za stvaranje moćnih proizvoda i AI rješenja. Isto tako, koriste ga mnoge istraživačke institucije, uključujući Stanford, MIT, INRIA, Cambridge i CMU.

OpenCV se koristio u brojnim aplikacijama, proizvodima i istraživačkim projektima od svog prvog izdanja 2000. godine pod BSD licencom, a zatim pod Apache 2 licencom. Osim kalibracije kamere, ove aplikacije također uključuju spajanje slika kamere za satelitske ili web-karte, poravnanje skeniranih slika, smanjenje šuma u medicinskim slikama, analizu objekata, sigurnost, nadzor i sustave za otkrivanje upada, automatski nadzor i sigurnosne sustave, proizvodnja AI inspekcije i bespilotnih letjelica, zemaljskih i podvodnih vozila itd.

OpenCV je stvoren za izvršavanje računalno intenzivnih zadataka vida što je moguće učinkovitije i brže. Kao rezultat toga, stavlja velik naglasak na aplikacije AI vida u stvarnom vremenu. Softver je višenišni (engl. *multithread*) i napisan u C-u koji je optimiziran za višejezgrene procesore. S preko 500 funkcija koje pokrivaju širok raspon tema povezanih s vidom, OpenCV ima za cilj pružiti intuitivnu infrastrukturu računalnog vida koja korisnicima olakšava brzo stvaranje složenih aplikacija za vid. OpenCV se često koristi u robotskom vidu, 3D vidu, sigurnosnoj analizi, sučelju čovjek-stroj i inspekciji tvorničkih proizvoda.

Primjene računalnog vida su bezbrojne. Manji broj pojedinaca može svjedočiti slučajevima upotrebe u određenim industrijama poput industrijske proizvodnje ili maloprodajne analitike, unatoč činjenici da je većina ljudi svjesna vrlo dobro poznatih slučajeva upotrebe u sigurnosti i videonadzoru ili samovozećim automobilima. Tvrtke iz raznih industrija uspjele su stvoriti aplikacije računalnog vida s određenom svrhom koje se bave vrlo specifičnim problemima (identificiraju greške proizvoda, broje stvari itd.) zahvaljujući nedavnom brzom napretku u tom području (Slika 5.1.).



Slika 5.1. OpenCV s demo kodom za prepoznavanje lica, izvor: <https://viso.ai/deep-learning/face-detection-overview/>

OpenCV, alat koji se koristi za razne aplikacije računalnog vida, uključujući identifikaciju slika, 2D ili 3D analizu, praćenje pokreta, prepoznavanje lica i više. Neke od najvažnijih mogućnosti programa su sljedeće:

- Detekcija objekata u stvarnom vremenu: Kako bi se locirali određeni predmeti u videopodacima ili fotografijama, kao što su automobili, ljudi, životinje i određeni

dijelovi strojeva ili opreme koji se koriste u industrijskoj proizvodnji, koristi se tehnologija detekcije objekata za primjenu prepoznavanja objekata na slici.

- Segmentacija slike: Primjenom metoda obrade slike segmentacija slike razdvaja sliku na različite segmente. Segmentacija se obično koristi u kombinaciji s drugim aktivnostima računalnog vida za poboljšanje, modificiranje ili pojednostavljenje slike. Primjer: u autonomnoj vožnji gdje se segmentacija slike koristi za prepoznavanje ceste.
- Prepoznavanje pokreta i gesta: Kroz videoanalizu, ljudske se geste tumače korištenjem ljudskog položaja i prepoznavanja gesti. Da bi se odredila unaprijed definirana kategorija, pokreti tijela, ruku ili lica mogu se identificirati i kategorizirati. Kako bi se ispitali pokreti tijela s referentnim ključnim točkama (zglobovi, udovi), analiza pokreta često je komponenta procjene poze. Razumijevanje načina na koji je objekt postavljen u 3D prostoru, na primjer, kako se rotira, može se postići izračunavanjem položaja objekta.
- Prepoznavanje lica: Prepoznavanjem ljudskog lica i uspoređivanjem s bazom podataka na temelju otkrivenih crta lica, automatsko prepoznavanje lica koristi se za identifikaciju ljudi. Odabir popularnih algoritama za prepoznavanje lica nudi OpenCV-ov FaceRecognizer za korištenje u praktičnim primjenama.
- Proširena stvarnost: Stvarni svijet i virtualni svijet mogu komunicirati u stvarnom vremenu zahvaljujući proširenoj stvarnosti. Kao rezultat toga, cilj proširene stvarnosti je dodavanje računalno generiranih perceptivnih podataka stvarnom svijetu [20].

## 6. Mjere za usporedbu metoda za detekciju objekata

Tehnike dubokog učenja kao što su klasifikacija slika, segmentacija, detekcija objekata koriste se vrlo često. Odabir prave metrike evaluacije vrlo je bitan za odluku koji model koristiti, kako podesiti hiperparametre, potrebu za tehnikama regulacije itd.

### 6.1. Klasifikacijske metrike

Prvo će se razmotriti metrika klasifikacije za klasifikaciju slika. Problemi klasifikacije slika mogu biti binarne ili višestruke klasifikacije. Primjer binarne klasifikacije uključuje npr. otkrivanje raka. Neki primjeri klasifikacije s više oznaka uključuju MNIST, CIFAR, baze slika s 10 klasa itd.

Prva mjera na koju se obično pomisli je točnost (engl. *Accuracy*). To je jednostavna mjera koja izračunava omjer točnih i pogrešnih predviđanja. Za primjer će se uzeti slučaj neuravnoteženog skupa podataka pacijenata oboljelih od raka. Ovdje će većina podatkovnih točaka pripadati negativnoj klasi, a vrlo malo pozitivnoj klasi. Dakle, samo klasificiranjem svih pacijenata kao "negativnih", model bi postigao veliku točnost. Zbog toga je bitno da se koristi balansirani skup podataka, odnosno, ako skup nije balansiran, može se koristiti težinska točnost (engl. *weighted accuracy*). Sljedeći korak je iscrtavanje matrice zabune (engl. *confusion matrix*). Ima 4 kategorije: istinito pozitivni (TP, engl. *true positive*), istinito negativni (TN, engl. *true negative*), lažno pozitivni (FP, engl. *false positive*), lažno negativni (FN, engl. *false negative*). Pomoću ove matrice mogu se izračunati razne korisne mjere [21]. Matrica zabune je prikazana na Slici 6.1. Formula za točnost (accuracy) prikazana je u formuli 6.1.

$$\text{Točnost} = (TP + TN) / (TP + TN + FP + FN) \quad (6.1.)$$

- **Odziv, osjetljivost, istinito pozitivna stopa (engl. *recall, sensitivity, true positive rate - TPR*)-** Označava postotak/udio pozitivne klase koja je ispravno predviđena i klasificirana. Formula za odziv prikazana je u formuli (6.2.)

$$\text{Odziv} = TP / (TP + FN) \quad (6.2.)$$

- **Specifičnost, istinito negativna stopa (engl. *specificity, true negative rate - TNR*)-** Iako je bitno točno predvidjeti pozitivnu klasu, što bi se dogodilo da se pacijentu s negativnim karcinomom netočno kaže da je u opasnosti (engl. *False Positive*). *Specificity* je metrika za izračunavanje koji je dio negativne klase ispravno predviđen i klasificiran. Formula za specifičnost prikazana je u formuli 6.3.

$$\text{Specifičnost} = TN / (FP + TN) \quad (6.3.)$$

*Specificity* i *Sensitivity* najčešće su korištene mjere. Ali, prvo se mora razumjeti i FPR da bi se dobio ROC.

- **lažno pozitivna stopa (engl. *false positive rate – FPR*) -** Ovo izračunava koliko je negativnih uzoraka pogrešno klasificirano kao pozitivno. FPR formula prikazana je u 6.4., a preciznost u formuli 6.5.

$$FPR = \text{False Positives} / (\text{False Positives} + \text{True Negatives}) \quad (6.4.)$$

$$FPR = 1 - \text{Preciznost}$$

$$\text{Preciznost} = TP / (TP + FP) \quad (6.5.)$$

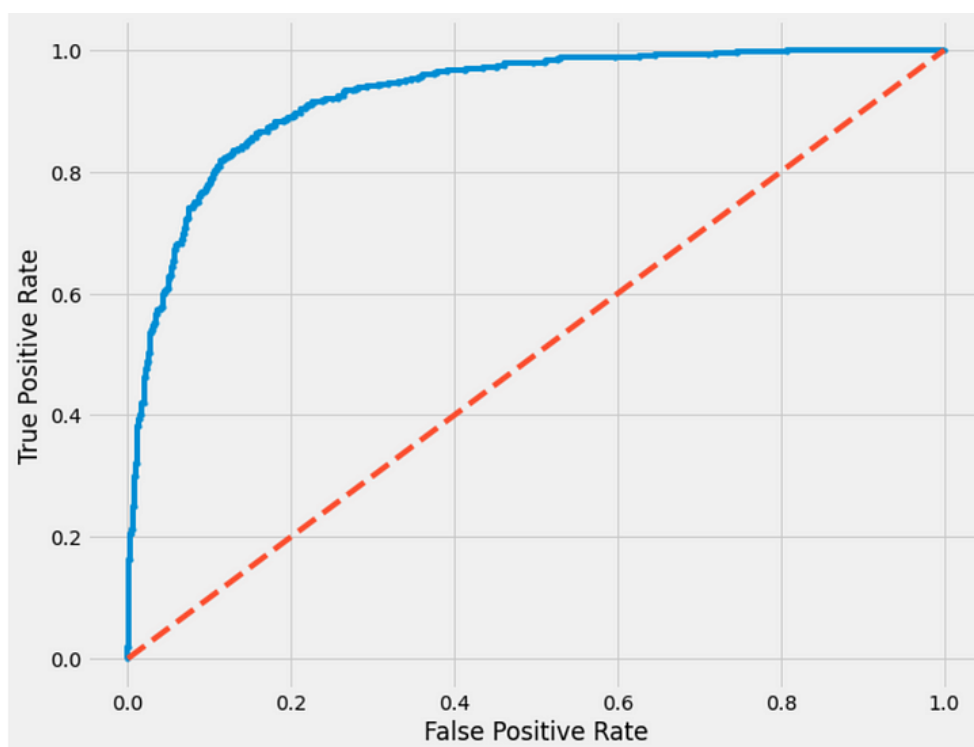
## MATRICA ZABUNE

		Stvarna slika	
		Pozitiv	Negativ
Predviđena slika	Pozitiv	Pravi pozitiv	Lažni pozitiv
	Negativ	Lažni negativ	Pravi negativ

Slika 6.1. Matrica zabune

Za dobar model klasifikacije, traži se viši TPR i niži FPR. Druga korisna metoda je dobivanje krivulje AUC ROC (engl. „*Area Under the Curve*” of the “*Receiver Operating Characteristic*”) za matricu zabune.

- **AUC ROC krivulja** – ROC je skraćeno od *Receiver Operator Characteristic*, a AUC od *Area under the curve*. Ovdje se prikazuje TPR u odnosu na FPR za različite pragove. Općenito, ako predviđanje ima vrijednost iznad 0,5, svrstava ga se u pozitivnu klasu, inače u negativnu klasu. Ovdje je ova odlučujuća granica 0,5 označena kao prag. Nije uvijek potrebno koristiti 0,5 kao prag, ponekad bi druge vrijednosti mogle dati najbolje rezultate. Da bi se to saznalo, crta se TPR u odnosu na FPR u odnosu na raspon vrijednosti praga. Pragovi obično variraju od 0,1, 0,2, 0,3, 0,4 i tako dalje do 1 [22]. Graf TPR i FPR u odnosu na raspon vrijednosti praga vidljiv je na Slici 6.2.



**Slika 6.2.** TPR i FPR u odnosu na raspon vrijednosti praga, izvor:

<https://ichi.pro/th/khumux-reim-tn-sahrab-roc-curve-laea-auc-70748239405639>

Za određeni prag, ako se želi izračunati ROC AUC rezultat, *sklearn* nudi funkciju.

Može se koristiti na sljedeći način prikazan u Programskom kodu 6.1.:

```
from sklearn.metrics import roc_auc_score

roc_auc = roc_auc_score(labels, predictions)
```

### Programski kod 6.1. Programski kod za ROC AUC

Gornji lijevi kut grafikona je mjesto gdje se traži optimalni prag.

Ako se želi iscrtati ROC AUC grafikon, koristi se sljedeća linija koda 6.2.:

```
fpr, tpr, thresholds = metrics.roc_curve(y,
                                         scores, pos_label=2)

import matplotlib.pyplot as plt

plt.plot(fpr, tpr)

plt.show()
```

### Programski kod 6.2. Programski kod za iscrtavanje ROC AUC grafikona

Ovdje će FPR i TPR koje daje funkcija biti popis/niz koji sadrži odgovarajuće vrijednosti za svaku vrijednost praga na popisu.

Također se mogu iscrtati *osjetljivost* i *specifičnost* u odnosu na pragove kako bi dobili više informacija.

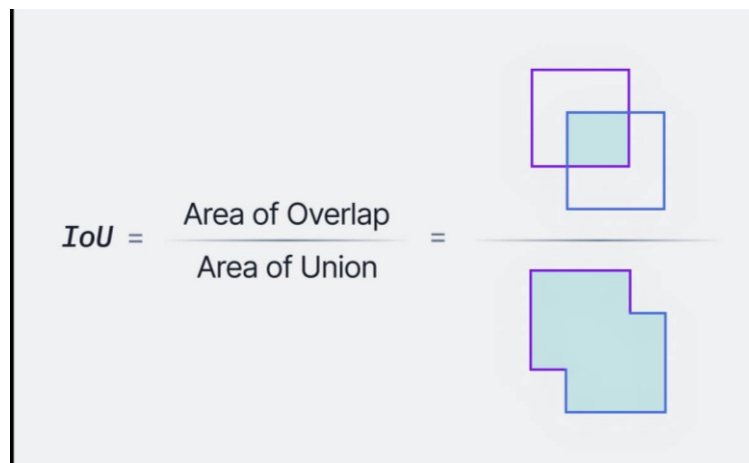
## 6.2. Mjere detekcije objekata

Detekcija objekata ima mnogo primjena uključujući detekciju lica, detekciju trake u sustavima s automatskim vozačem i tako dalje. Ovdje se koristi drugačiji skup mjera za procjenu. Najpopularniji je IoU (engl. *Intersection over Union*).

- **Presjek nad unijom (engl. *intersection over union*, IoU)**- Dakle, u problemima detekcije ili segmentacije objekta, osnovne oznake istine su maske dijela ili graničnog okvira gdje je objekt prisutan. IoU mjera utvrđuje razliku između



graničnog okvira predviđanja i graničnog okvira temeljne istine. IoU je prikazan na Slici 6.3.



**Slika 6.3.** Prikaz IoU, izvor: <https://www.v7labs.com/blog/yolo-object-detection>

IoU će biti vrijednost između 0-1. Za okvire koji se savršeno preklapaju, to će biti 1, a za predviđanje bez preklapanja je 0. Općenito, IoU bi trebao biti iznad 0,5 za pristojan model detekcije objekta [23, 24].

Recimo da se prag detekcije postavi na 0,5, u tom slučaju:

- ako je  $IoU \geq 0,5$ , klasificira se detekcija objekta kao *Istinito pozitivan*,
- ako je  $IoU < 0,5$ , tada je to pogrešno otkrivanje i klasificira se kao *Lažno pozitivan*,
- Kada je temeljna istina prisutna na slici i model nije uspio detektirati objekt, klasificira se kao *Lažno negativan*.
- *Istinito negativan* je svaki dio slike gdje se nije predvidio objekt. Ova mjera nije korisna za detekciju objekata, stoga se zanemaruje [25].

Kao mjera za procjenu izvedbe se koriste *Preciznost* i *Odziv*. Oni se izračunavaju korištenjem TP, FP i FN.

Također se mora uzeti u obzir ocjena pouzdanosti za svaki objekt koji je otkrio model na slici. Treba uzeti u obzir sve predviđene granične okvire s ocjenom pouzdanosti iznad određenog praga. Granični okviri iznad vrijednosti praga smatraju se pozitivnim okvirima, a svi predviđeni granični okviri ispod vrijednosti praga smatraju se negativnima.

- **Srednja prosječna preciznost (engl. *mean Average Precision*, mAP)**– AP se obračunava pojedinačno. To znači da postoji onoliko AP vrijednosti koliko je klasa. Ove AP vrijednosti su prosječne kako bi se dobila mAP. mAP je prosjek AP vrijednosti za sve klase. Formula za mAP prikazana je u donjoj formuli 6.6.

$$mAP@ = \frac{1}{n} \sum_{i=1}^n AP_i \text{ za } n \text{ klasa} \quad (6.6.)$$

Pomoću IOU mogu se izračunati *preciznost* i *odziv*. I to na sljedeći način: postavlja se vrijednost praga IoU. Na primjer, recimo da se zadrži prag IoU na 0,5. Onda za predviđanje IoU od 0,8, može se klasificirati kao *Istinito pozitivan*. Ako je 0,4 (manje od 0,5) onda je *Lažno pozitivan*. Također treba imati na umu da ako se promijeni prag na 0,4, tada će se ovo predviđanje klasificirati kao *True Positive*. Dakle, različiti pragovi mogu dati različite mjere.

Zatim se srednja preciznost (engl. *Average Precision*, AP) dobiva pronalaženjem područja ispod krivulje *preciznost-odziv*. Definiira se na sljedeći način 6.7.:

$$AP@a = \int_0^1 p(r)dr \quad (6.7.)$$

$AP@a$  ili  $AP\alpha$  znači da se preciznost AP-a procjenjuje na  $\alpha$  IoU pragu. Ako vidite mjerne podatke kao što su AP50 i A75, oni znače AP izračunat na IoU=0,5 odnosno IoU=0,75.

Visoko područje ispod PR krivulje znači visoki *odziv* i visoku *preciznost*. Naravno, PR krivulja je cik-cak crta. To znači da ne opada monotono. Ovo svojstvo može se ukloniti pomoću metoda interpolacije. U nastavku ćemo raspravljati o dvije od tih metoda interpolacije:

- Interpolacijska metoda u 11 točaka - Grafikon interpoliranih rezultata preciznosti za rezultate modela na 11 jednako razmaknutih vrijednosti odaziva, naime, 0,0, 0,1, 0,2, . . . 1.0. Definiira se kao formula 6.8.:

$$AP@a_{11} = \frac{1}{11} \sum_{r \in R} p_{interp}(r) \quad (6.8.)$$

gdje je,  $R = \{0.0, 0.1, 0.2, \dots 1.0\}$  i formula 6.9.:

$$p_{interp}(r) = \max_{r': r' \geq r} p(r') \quad (6.9.)$$

to jest, interpolirana preciznost pri povratnoj vrijednosti,  $r$  — To je najveća preciznost za bilo koju povratnu vrijednost  $r' \geq r$ .

- Interpolacijska metoda u svim točkama - U ovom slučaju radi se interpolacija za sve pozicije (vrijednosti odziva), tj. Prikazanu u formuli 6.10.:

$$AP@a = \sum_i (r_{i+1} - r_i) p_{interp}(r + 1),$$

$$gdje je p_{interp}(r_{i+1}) = \max_{r': r' \geq r_{i+1}} p(r') \quad (6.10.)$$

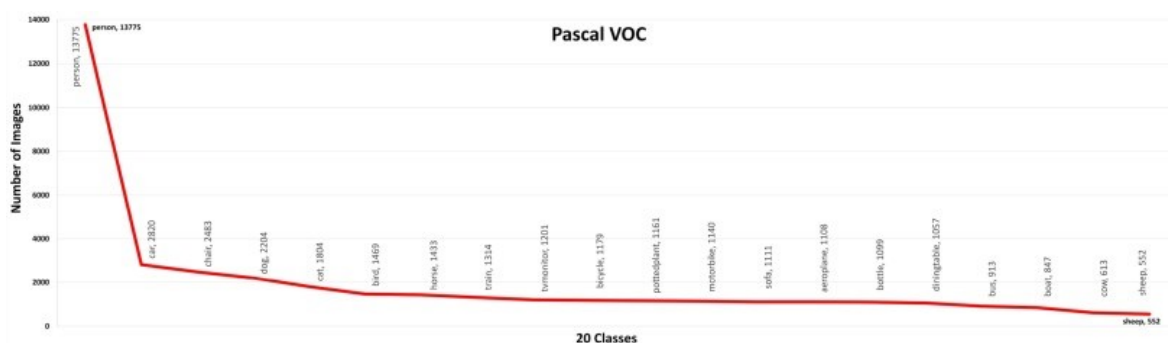
mAP za detekciju objekta prosjek je AP-a izračunatog za sve klase kako bi se odredila točnost skupa detekcija objekata iz modela u usporedbi s temeljnim položajima objekata (engl. *ground truth*) objekta skupa podataka. mAP se izračunava uzimanjem srednje vrijednosti AP-a za sve klase i/ili ukupne IoU pragove.

Mnogi algoritmi za detekciju objekata uključujući Faster R-CNN, MobileNet koriste ovu mjeru. Ova metrika daje numeričku vrijednost što olakšava usporedbu s drugim modelima [23, 25].

## 6.3. Baze za učenje

### 6.3.1. Pascal VOC 07/12

VOC (The Pascal Visual Object Classes) baza slika je višegodišnji napor da se ubrza razvoj u području vizualne percepcije. Započelo je 2005. sa zadacima klasifikacije i detekcije na četiri klase objekata, ali dvije verzije ovih izazova uglavnom se koriste kao standardna referentna vrijednost. Dok je izazov VOC07 imao 5000 slika za trening i više od 12000 označenih objekata, izazov VOC12 ih je povećao na 11000 slika za trening i više od 27000 označenih objekata. Klase objekata proširene su na 20 kategorija, a uključeni su i zadaci poput segmentacije i otkrivanja radnji. Pascal VOC uveo je mAP na 0,5 IoU kako bi procijenio izvedbu modela [24]. Slika 6.4. prikazuje distribuciju broja slika u skupu podataka Pascal VOC.

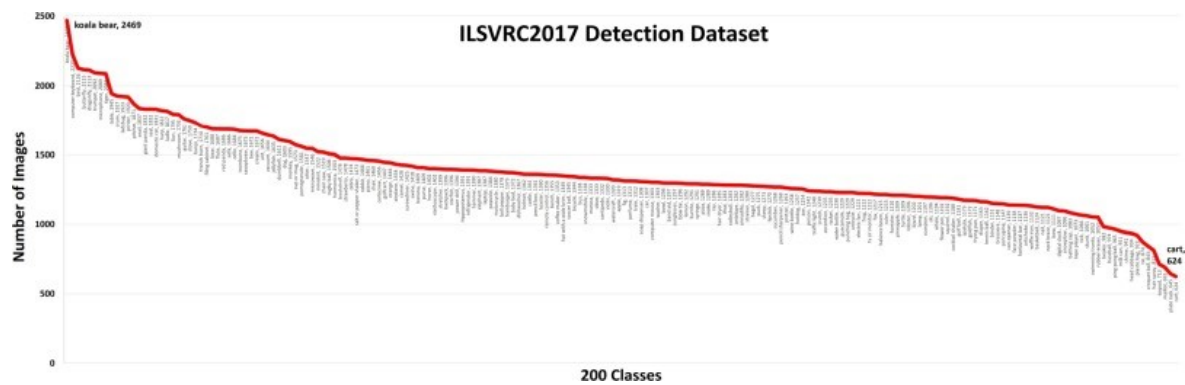


**Slika 6.4.** Distribucija broja slika u skupu podataka Pascal VOC, izvor: <https://www.sciencedirect.com/science/article/pii/S1051200422001312>

### 6.3.2. MS-COCO

*MS-COCO (Microsoft Common Objects in Context)* jedan je od najizazovnijih dostupnih skupova podataka. Sadrži 91 uobičajeni predmet pronađen u njihovom prirodnom kontekstu koje četverogodišnji čovjek može lako prepoznati. Objavljen je 2015. godine i od tada je njegova popularnost samo rasla. Ima više od dva milijuna primjeraka i prosječno 3,5 kategorije po slici. Nadalje, sadrži 7,7 instanci po slici, više od ostalih popularnih skupova podataka. MS COCO također se sastoji od slika s različitim gledišta. Također je uvela strožu metodu za mjerenje performansi detektora. Za razliku od Pascal VOC i ILSVCR, izračunava IoU od 0,5 do 0,95 u koracima od 0,5, zatim koristi





**Slika 6.6.** Distribucija broja slika u skupu podataka ImageNet, izvor:  
<https://www.sciencedirect.com/science/article/pii/S1051200422001312>

#### 6.3.4. Open Image

Google-ov „Open Images“ sastoji se od 9,2 milijuna slika, označenih oznakama na razini slike, graničnim okvirima objekata i segmentacijskim maskama, između ostalog. Pokrenut je 2017. godine i od tada je dobio šest ažuriranja. Ima 16 milijuna graničnih okvira za 600 kategorija na 1,9 milijuna slika za otkrivanje, što ga čini najvećim skupom podataka za lokalizaciju objekata. Njegovi su kreatori posebno pazili na odabir zanimljivih, složenih i raznolikih slika, imajući 8,3 kategorije objekata po slici. Učinjeno je nekoliko promjena u AP-u uvedenom u Pascal VOC kao što je ignoriranje neoznačene klase, zahtjev za detekciju za klasu i njenu podklasu itd. Slika 6.7. prikazuje distribuciju broja slika s obzirom na različite klase u skupu podataka Open Images [24].



**Slika 6.7.** Distribucija broja slika s obzirom na različite klase u skupu podataka

Open Images, izvor:

<https://www.sciencedirect.com/science/article/pii/S1051200422001312>

## 6.4. Primjer usporedbe modela

Skup podataka MS COCO koristi se za provođenje eksperimenata. Svi modeli obučavani su od nule. Dizajniran je osnovni model za edge GPU, normal GPU, i cloud GPU, a zovu se: YOLOv7-tiny, YOLOv7 i YOLOv7-W6. Također, koristi se osnovni model za skaliranje modela za različite zahtjeve usluge. Za YOLOv7 radi se „*scaling on neck*“ i koristi predložena složena metoda skaliranja za izvođenje skaliranja dubine i širine cijelog modela kako bi se dobio YOLOv7-X. Što se tiče YOLOv7-W6, koristi se novopredložena metoda skaliranja spojeva za dobivanje YOLOv7-E6 i bYOLOv7-D6. Osim toga, koristi se predloženi E-ELAN za YOLOv7-E6, i time se kompletira YOLOv7-E6E. S obzirom na to da je YOLOv7-tiny edge GPU-orijentirana arhitektura, koristit će propušteni ReLU kao funkciju aktivacije. Za druge modele koristit će se „SiLU“ kao aktivacijska funkcija [25].

Odabrala se prethodna verzija YOLO-a i najsuvremeniji detektor objekata YOLOR kao modeli za osnovnu usporedbu. Slika 23. prikazuje usporedbu predloženih YOLOv7

modela i onih osnovnih koji su trenirani s istim postavkama. Iz rezultata vidi se da modeli temeljeni na YOLOv7 nadmašuju osnovne modele u smislu broja parametara, količine izračuna i točnosti. Za modele dizajnirane za normalan GPU, YOLOv7 i YOLOv7x smanjuju oko 40% parametara i 20% izračunavanja, a još uvijek postižu veći AP. Za model dizajniran za edge GPU, YOLOv7-tiny smanjuje 39% parametara i 49% proračuna i postiže isti AP kao YOLOv4-tiny-3l.

PPYOLOE-S [83]	7.9M	17.4G	640	208	43.1% / 42.7%	60.5%	46.6%
YOLOv5-N (r6.1) [22]	1.9M	4.5G	640	159	- / 28.0%	-	-
YOLOv5-S (r6.1) [22]	7.2M	16.5G	640	156	- / 37.4%	-	-
PPYOLOE-M [83]	23.4M	49.9G	640	123	48.9% / 48.6%	66.5%	53.0%
YOLOv5-N6 (r6.1) [22]	3.2M	18.4G	1280	123	- / 36.0%	-	-
YOLOv5-S6 (r6.1) [22]	12.6M	67.2G	1280	122	- / 44.8%	-	-
YOLOv5-M (r6.1) [22]	21.2M	49.0G	640	122	- / 45.4%	-	-
YOLOv7	36.9M	104.7G	640	118	51.4% / 51.2%	69.7%	55.9%
YOLOR-CSP [79]	52.9M	120.4G	640	106	51.1% / 50.8%	69.6%	55.7%
YOLOX-S [20]	9.0M	26.8G	640	102	40.5% / 40.5%	-	-
YOLOv5-L (r6.1) [22]	46.5M	109.1G	640	99	- / 49.0%	-	-
YOLOv7-X	71.3M	189.9G	640	98	53.1% / 52.9%	71.2%	57.8%
YOLOv5-M6 (r6.1) [22]	35.7M	200.0G	1280	90	- / 51.3%	-	-
YOLOR-CSP-X [79]	96.9M	226.8G	640	87	53.0% / 52.7%	71.4%	57.9%
YOLOv5-X (r6.1) [22]	86.7M	205.7G	640	83	- / 50.7%	-	-
YOLOX-M [20]	25.3M	73.8G	640	81	47.2% / 46.9%	-	-
YOLOv7-W6	70.4M	360.0G	1280	80	54.9% / 54.6%	72.6%	60.1%
PPYOLOE-L [83]	52.2M	110.1G	640	78	51.4% / 50.9%	68.9%	55.6%
YOLOR-P6 [79]	37.2M	325.6G	1280	76	53.9% / 53.5%	71.4%	58.9%
YOLOX-L [20]	54.2M	155.6G	640	69	50.1% / 49.7%	-	-
YOLOR-W6 [79]	79.8G	453.2G	1280	66	55.2% / 54.8%	72.7%	60.5%
YOLOv5-L6 (r6.1) [22]	76.8M	445.6G	1280	63	- / 53.7%	-	-
YOLOX-X [20]	99.1M	281.9G	640	58	51.5% / 51.1%	-	-
YOLOv7-E6	97.2M	515.2G	1280	54	56.0% / 55.9%	73.5%	61.2%
YOLOR-E6 [79]	115.8M	683.2G	1280	45	55.8% / 55.7%	73.4%	61.1%
PPYOLOE-X [83]	98.4M	206.6G	640	45	52.2% / 51.9%	69.9%	56.5%
YOLOv7-D6	154.7M	806.8G	1280	43	56.6% / 56.3%	74.0%	61.8%
YOLOv5-X6 (r6.1) [22]	140.7M	839.2G	1280	38	- / 55.0%	-	-
YOLOv7-E6E	151.7M	843.2G	1280	35	56.8% / 56.8%	74.4%	62.1%
YOLOR-D6 [79]	151.7M	935.6G	1280	34	56.5% / 56.1%	74.1%	61.9%
Deformable DETR [97]	40.0M	173.0G	-	19	- / 46.2%	-	-

**Tablica 6.1.** Usporedba najsvremenijih detektora objekata, izvor:

[https://openaccess.thecvf.com/content/CVPR2023/papers/Wang\\_YOLOv7\\_Trainable\\_Bag-of-Freebies\\_Sets\\_New\\_State-of-the-Art\\_for\\_Real-Time\\_Object\\_Detectors\\_CVPR\\_2023\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2023/papers/Wang_YOLOv7_Trainable_Bag-of-Freebies_Sets_New_State-of-the-Art_for_Real-Time_Object_Detectors_CVPR_2023_paper.pdf)



Predložena metoda uspoređuje se s najsvremenijim detektorima objekata, a mogu se vidjeti u Tablici 6.1. Iz tablice znamo da predložena metoda ima sveobuhvatno najbolji kompromis između brzine i točnosti.

Tablica 6.2. prikazuje rezultate dobivene korištenjem različitih strategija skaliranja modela za povećanje. Među njima, predložena složena metoda skaliranja je povećanje dubine računskog bloka za 1,5 puta i širine prijelaznog bloka za 1,25 puta. Iz rezultata može se vidjeti da predložena složena strategija skaliranja može učinkovitije koristiti parametre i izračunavanje.

<b>Model</b>	<b>#Param.</b>	<b>FLOPs</b>	<b>Size</b>	<b>AP<sup>val</sup></b>	<b>AP<sub>50</sub><sup>val</sup></b>	<b>AP<sub>75</sub><sup>val</sup></b>
<b>base (v7-X light)</b>	47.0M	125.5G	640	51.7%	70.1%	56.0%
<b>width only (1.25 <i>w</i>)</b>	73.4M	195.5G	640	52.4%	70.9%	57.1%
<b>depth only (2.0 <i>d</i>)</b>	69.3M	187.6G	640	52.7%	70.8%	57.3%
<b>compound (v7-X)</b>	71.3M	189.9G	640	<b>52.9%</b>	<b>71.1%</b>	<b>57.5%</b>
<b>improvement</b>	-	-	-	<b>+1.2</b>	<b>+1.0</b>	<b>+1.5</b>

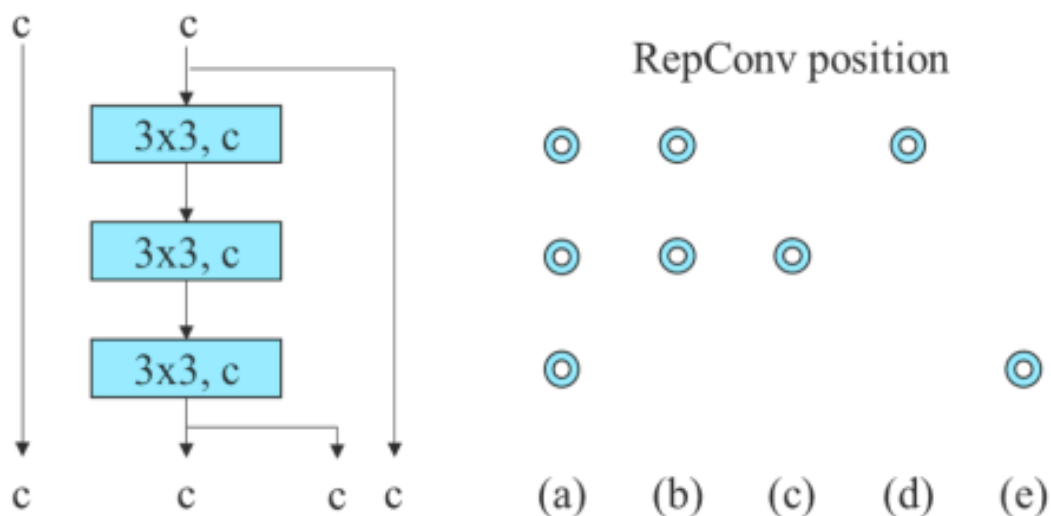
**Tablica 6.2.** Ispitivanje parametara predloženog skaliranja modela, izvor:

[https://openaccess.thecvf.com/content/CVPR2023/papers/Wang\\_YOLOv7\\_Trainable\\_Bag-of-Freebies\\_Sets\\_New\\_State-of-the-Art\\_for\\_Real-Time\\_Object\\_Detectors\\_CVPR\\_2023\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2023/papers/Wang_YOLOv7_Trainable_Bag-of-Freebies_Sets_New_State-of-the-Art_for_Real-Time_Object_Detectors_CVPR_2023_paper.pdf)

Da bi se provjerila općenitost predloženog planiranog modela reparametrizacije, koristi se model koji se temelji na ulančavanju i modelu koji se temelji na provjeri. Model temeljen na ulančavanju i model temeljen na ostatku koje smo odabrali za verifikaciju su 3-složni ELAN (engl. *Efficient Layer Aggregation Network*) i CSPDarknet (engl. *Cross Stage Partial Network*, koji se koristi u Darknet neuronskoj mreži).

U eksperimentu modela koji se temelji na ulančavanju, zamjenjuju se  $3 \times 3$  konvolucijska sloja na različitim pozicijama u 3-složenom ELAN-u s RepConv, a detaljna

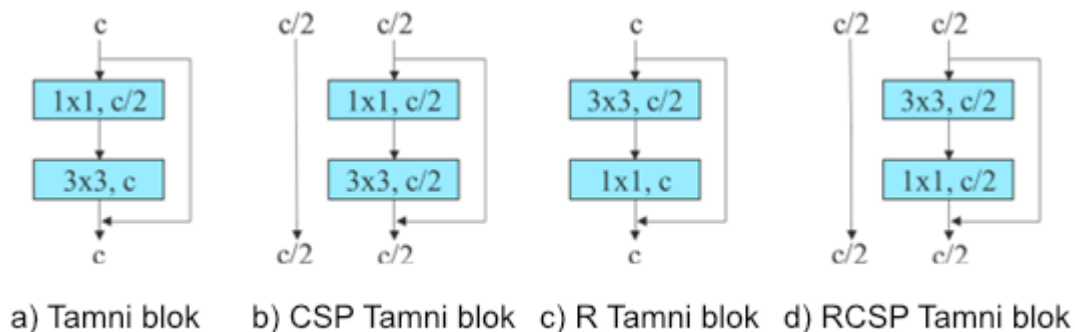
konfiguracija prikazana je na Slici 6.8. Plavi kružići na slici su pozicija koja zamjenjuje Conv s RepConv.



**Slika 6.8.** Planirani RepConv 3-složni ELAN

Iz rezultata prikazanih na Slici 6.8. možemo vidjeti da su sve AP vrijednosti prisutne u predloženom planiranom modelu reparametrizacije.

U eksperimentu modela temeljenog na ostatku, dizajnirao se obrnuti tamni blok kako bi odgovarao strategiji dizajna za eksperiment, čija je arhitektura prikazana na Slici 6.9. Mijenja se položaj konvolucijskog sloja  $1 \times 1$  i  $3 \times 3$  u tamnom bloku kako bi odgovarao planiranoj strategiji dizajna modela reparametrizacije.



**Slika 6.9.** Obrnuti CSPDarknet

Rezultati eksperimenta ilustrirani u Tablici 6.3 u potpunosti potvrđuju da je predloženi planirani model ponovne parametrizacije jednako učinkovit na rezidualnom modelu. Otkrilo se da dizajn RepCSPResNet također odgovara predloženom uzorku dizajna.

Model	$AP^{val}$	$AP_{50}^{val}$	$AP_{75}^{val}$	$AP_S^{val}$	$AP_M^{val}$	$AP_L^{val}$
<b>base (YOLOR-W6)</b>	54.82%	72.39%	59.95%	39.68%	59.38%	<b>68.30%</b>
<b>RepCSP</b>	54.67%	72.50%	59.58%	40.22%	<b>59.61%</b>	67.87%
<b>RCSP</b>	54.36%	71.95%	59.54%	40.15%	59.02%	67.44%
<b>RepRCSP</b>	<b>54.85%</b>	<b>72.51%</b>	<b>60.08%</b>	<b>40.53%</b>	59.52%	68.06%
<b>base (YOLOR-CSP)</b>	50.81%	69.47%	55.28%	33.74%	<b>56.01%</b>	65.38%
<b>RepRCSP</b>	<b>50.91%</b>	<b>69.54%</b>	<b>55.55%</b>	<b>34.44%</b>	55.74%	<b>65.46%</b>

**Tablica 6.3.** Studija ablacije na planiranom modelu RepResidual

U pomoćnom gubitku za *pokuse pomoćne glave* (engl. *auxiliary head*), uspoređuje se opće neovisna dodjela oznaka za *metode glavne glave* (engl. *lead head*) i *pomoćne glave*, a također se uspoređuju dvije predložene metode dodjele oznaka. Rezultati se nalaze na Tablici 6.4. Iz rezultata na slici, jasno je da bilo koji model koji povećava pomoćne gubitke može značajno poboljšati ukupnu izvedbu.

Model	Size	$AP^{val}$	$AP_{50}^{val}$	$AP_{75}^{val}$
<b>base (v7-E6)</b>	1280	55.6%	73.2%	60.7%
<b>independent</b>	1280	55.8%	73.4%	60.9%
<b>lead guided</b>	1280	55.9%	73.5%	61.0%
<b>coarse-to-fine lead guided</b>	1280	<b>55.9%</b>	<b>73.5%</b>	<b>61.1%</b>

**Tablica 6.4.** Istraživanje parametara za predloženu pomoćnu glavu

Na Tablici 6.5. dodatno su se usporedili rezultati sa/bez uvođenja ograničenja gornje granice. Sudeći po brojevima u tablici, metodom ograničenja gornje granice objektnosti udaljenošću od središta objekta može se postići bolja izvedba.

<b>Model</b>	<b>Size</b>	<b>AP<sup>val</sup></b>	<b>AP<sub>50</sub><sup>val</sup></b>	<b>AP<sub>75</sub><sup>val</sup></b>
<b>base (v7-E6)</b>	1280	55.6%	73.2%	60.7%
<b>aux without constraint</b>	1280	55.9%	73.5%	61.0%
<b>aux with constraint</b>	1280	<b>55.9%</b>	<b>73.5%</b>	<b>61.1%</b>

**Tablica 6.5.** Istraživanje parametara za predloženu ograničenu pomoćnu glavu

Izravno spojite *pomoćnu glavu* na piramidu u srednji sloj za trening jer ono može nadoknaditi informacije koje se mogu izgubiti u predviđanju piramide sljedeće razine. Kako bi se riješio ovaj problem, treba spojiti *pomoćnu glavu* nakon jednog od skupova značajki prije spajanja kardinalnosti E-ELAN-a. Tablica 6.6. prikazuje rezultate dobivene korištenjem metode vođene glavnom glavom iz grubog u fino (engl. *coarse-to-fine lead guided*) i metode djelomično vođene glavnom glavom iz grubog u fino (engl. *partial coarse-to-fine lead guided*) metode. Očito je da *partial coarse-to-fine lead guided* metoda ima bolji učinak.

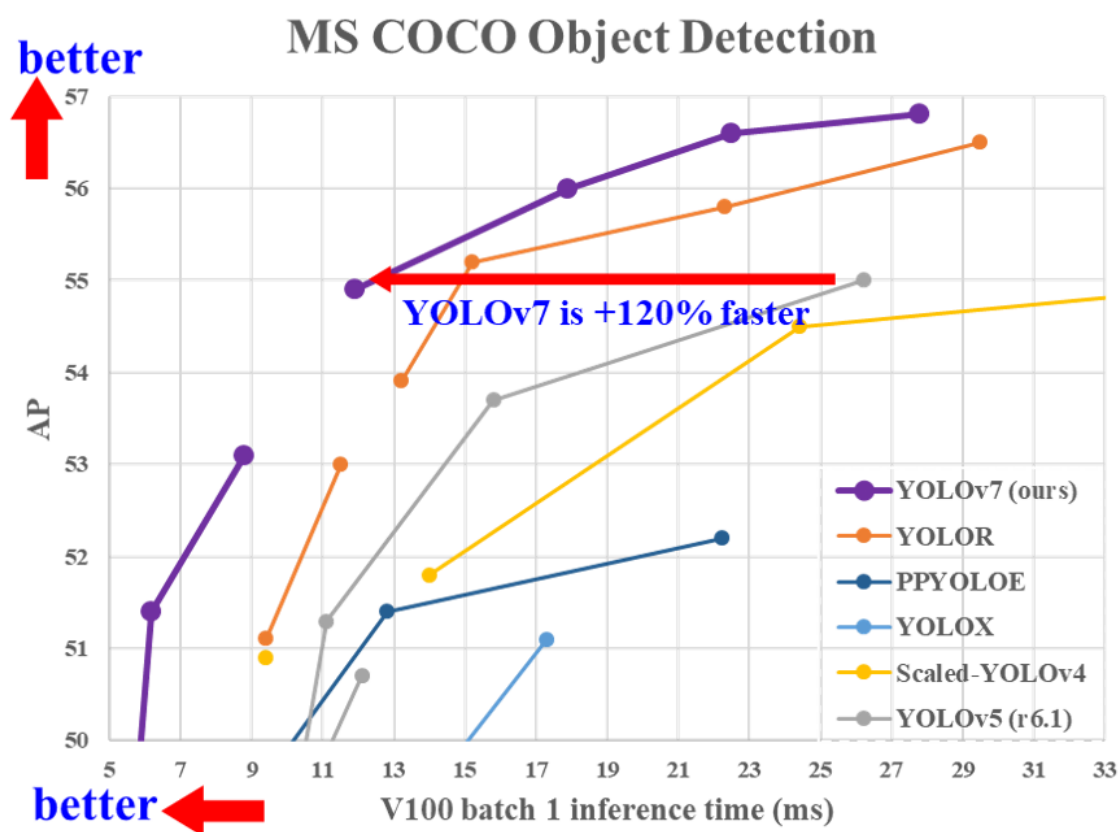
<b>Model</b>	<b>Size</b>	<b>AP<sup>val</sup></b>	<b>AP<sub>50</sub><sup>val</sup></b>	<b>AP<sub>75</sub><sup>val</sup></b>
<b>base (v7-E6E)</b>	1280	56.3%	74.0%	61.5%
<b>aux</b>	1280	56.5%	74.0%	61.6%
<b>partial aux</b>	1280	<b>56.8%</b>	<b>74.4%</b>	<b>62.1%</b>

**Tablica 6.6.** Istraživanje parametara za predloženu pomoćnu glavu

## 7. YOLOv7

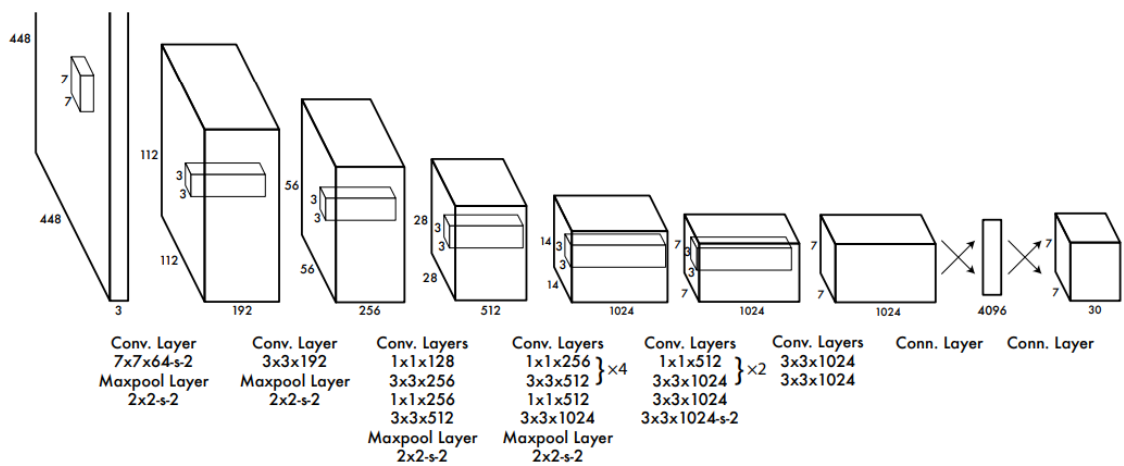
YOLO algoritam prošao je kroz nekoliko iteracija, pri čemu je svaka verzija donijela poboljšanja i inovacije u detekciji objekata. YOLOv7 donio je znatna poboljšanja u brzini i preciznosti prepoznavanja objekata, te je superiorniji od mnogih postojećih modela, prema provedenim testiranjima [25].

Za usporedbu (Slika 7.1.), YOLO je u testiranjima bio brži od: SWINL Cascade-Mask R-CNN za 509% u brzini i 2% u preciznosti, kao i ConvNeXt-XL Cascade-Mask R-CNN za 551% u brzini i 0.7 AP u preciznosti. Ova usporedba je rađena na isključivo MS-COCO skupu podataka bez predtreniranih težina.



Kao što samo ime modela otkriva, YOLO (engl. *You Only Look Once*) sliku sustav pretražuje samo jednom kako bi predvidio gdje se nalaze tražene klase na njoj. Sustav se sastoji od jedne konvolucijske mreže koja istovremeno može predvidjeti više okvira za više klasa, sve u jednoj iteraciji. Ta mogućnost čini YOLO izuzetno korisnim za prepoznavanje u realnom vremenu, odnosno s kašnjenjem koje se može smatrati dovoljno malim kako bi se nazvalo realnim vremenom (kašnjenje do 25 milisekundi). Sam sustav kroz trening optimizira svoje performanse detekcije. Osnovni model YOLO može raditi s preko 45 sličica u sekundi.

Implementacija YOLO modela je zasnovana na konvolucijskoj neuronskoj mreži. Početni konvolucijski sloj ekstrahira karakteristike iz slike, dok potpuno povezani slojevi predviđaju izlazne vrijednosti i koordinate objekata. U YOLO sustavu, nakon mreže od 24 konvolucijska sloja slijede 2 potpuno povezana sloja. Koriste se redukcijski slojevi  $1 \times 1$  koji se nadovezuju na  $3 \times 3$  konvolucijske slojeve, prikazano na Slici 7.2.



**Slika 7.2.** Arhitektura YOLO sustava, izvor:

[https://openaccess.thecvf.com/content/CVPR2023/papers/Wang\\_YOLOv7\\_Trainable\\_Bag-of-Freebies\\_Sets\\_New\\_State-of-the-Art\\_for\\_Real-Time\\_Object\\_Detectors\\_CVPR\\_2023\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2023/papers/Wang_YOLOv7_Trainable_Bag-of-Freebies_Sets_New_State-of-the-Art_for_Real-Time_Object_Detectors_CVPR_2023_paper.pdf)

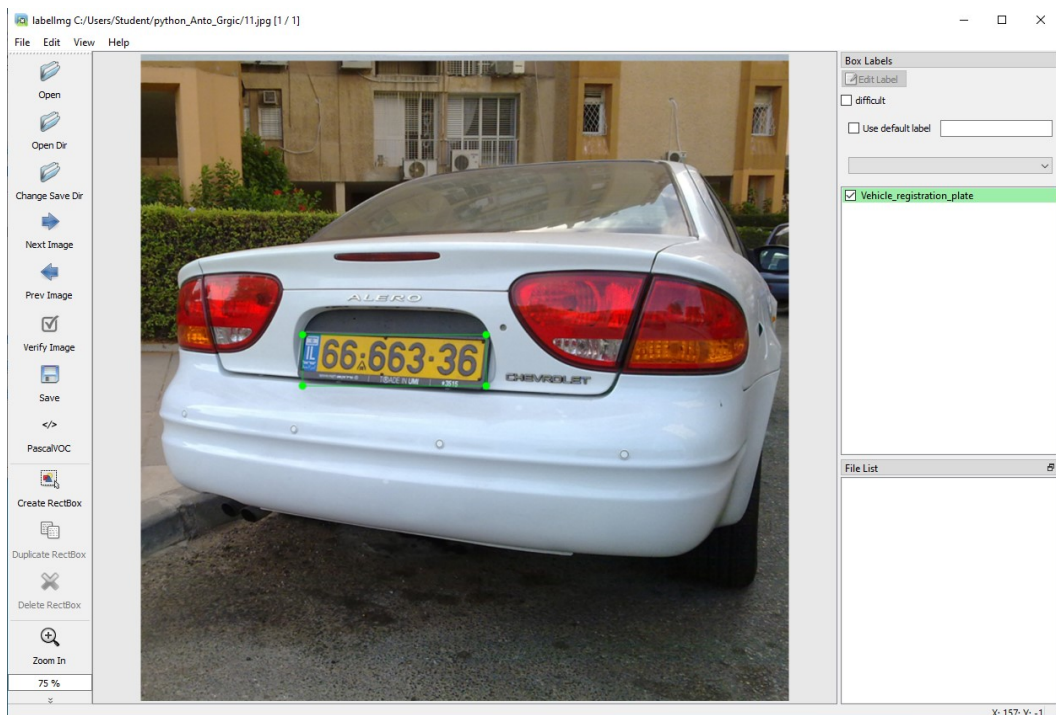
## 8. Praktični zadatak

### 8.1. Prikupljanje skupa podataka i anotacija

Za izradu sustava detekcije registarskih pločica i prepoznavanja znakova temeljenog na modelima dubokog učenja, kritičan korak je odabir i prikupljanje odgovarajućeg skupa podataka. U tu svrhu, koristio sam javno dostupan skup podataka, već ranije spomenuti *Open Images Dataset v7* preko kojeg sam prikupio 400 predanotiranih slika za detekciju, predodređene klase "registracijska pločica vozila" (engl. "Vehicle Registration Plate"). Uz taj skup koristio sam i svoj osobni skup slika, njih oko 150 kako bih dobio što robusniji sustav u realnim uvjetima. Osobni skup slika sam ručno anotirao putem "LabelImg" Python grafičkog programa za anotaciju slika. Program se jednostavno koristi, a potrebno je samo uvesti biblioteku LabelImg preko naredbe:

- `pip install labelImg`

Dokumentacija je dostupna ovdje: <https://github.com/heartexlabs/labelImg>



Slika 8.1. LabelImg

## 8.2. Arhitektura modela i treniranje

Za detekciju registarskih pločica i prepoznavanje znakova, koristio sam adaptiranu verziju popularne arhitekture dubokog učenja, YOLOv7, koja se temelji na mreži konvolucijskih neuronskih mreža (CNN). YOLOv7 model kao konvolucijsku neuronsku mrežu koristi E-ELAN arhitekturu (engl. *Extended Efficient layer aggregation network*), baziranu na prijašnjoj verziji ELAN arhitekture.

Programski jezik koji sam koristio za implementaciju YOLO je Python, uz pomoć Pytorcha. Pytorch je okvir za duboko učenje koji se koristio kao podložena biblioteka za implementaciju i trening YOLO.

Nakon eksperimenata s hiperparametrima, postigao sam zadovoljavajuću ravnotežu između brzine i preciznosti svog modela.

Komandna linija kojom sam krenuo s treniranjem modela je sljedeća:

```
python train.py --workers 1 --device 0 --batch-size 8 --epochs 100 --img 640  
640 --data data/custom_data.yaml --hyp data/hyp.scratch.custom.yaml --cfg  
cfg/training/yolov7-custom.yaml --name yolov7-custom
```

**Programski kod 8.1.** Kod komandne linije za pokretanje treniranja modela

Opis pojedinih parametara komandne linije:

python train.py: Ovo je osnovna naredba za pokretanje Python skripte nazvane "train.py" pomoću Python interpretatora. Skripta "train.py" je službena skripta otvorenog koda YOLO i sadrži sav kod potreban za treniranje YOLOv7 modela.

--workers 1: Ovaj parametar određuje broj procesorskih jezgri (engl. *Central Processing Unit*, CPU) za učitavanje podataka tijekom treninga. U ovom slučaju postavljen je na 1, što znači da će jedna jezgra biti odgovorna za učitavanje i obradu podataka za trening. Ovaj broj se može povećati kako bi se ubrzalo učitavanje podataka ako imate više jezgri CPU.

--device 0: Ovaj parametar postavlja uređaj (obično GPU) na kojem će se izvoditi trening. U ovom slučaju, postavljen je na 0, što znači da će se prva GPU (grafička kartica)



u sustavu koristiti za trening. Ukoliko računalo na kojem se izvodi treniranje ima više od jedne grafičke kartice, `--device` se može postaviti na 1 ili više.

`--batch-size 8`: Ovaj parametar određuje veličinu serije (engl. *batch size*) za trening. Označava koliko slika će se obraditi u svakom prolazu unaprijed i unatrag tijekom treninga. U ovom slučaju, postavljen je na 8, što znači da će model obraditi 8 slika u svakoj iteraciji treninga. Za jača računala, ovaj parametar se može postaviti i na 16 ili 32.

`--epochs 100`: Ovaj parametar postavlja broj epoha treninga. Epoha je jedan potpuni prolazak kroz cijeli skup podataka za trening. U ovom slučaju, trening se izveo na 100 epoha, što znači da će model vidjeti cijeli skup podataka 100 puta tijekom treninga.

`--img 640 640`: Ovaj parametar postavlja veličinu ulazne slike za model. YOLOv7 obično mijenja veličinu ulaznih slika na fiksnu veličinu prije obrade. U ovom slučaju, ulazne slike će biti promijenjene na 640 piksela u širinu i 640 piksela u visinu.

`--data data/custom_data.yaml`: Ovaj parametar određuje putanju do YAML konfiguracijske datoteke koja sadrži informacije o prilagođenom skupu podataka, uključujući putanje podataka, naziv klasa i druge postavke vezane uz skup podataka.

`--hyp data/hyp.scratch.custom.yaml`: Ovaj parametar određuje putanju do YAML datoteke koja sadrži hiperparametre za proces treninga. Hiperparametri su postavke koje kontroliraju proces treninga, kao što su stopa učenja i sl.

`--cfg cfg/training/yolov7-custom.yaml`: Ovaj parametar određuje putanju do konfiguracijske datoteke modela. Definira arhitekturu YOLOv7 modela, uključujući broj slojeva, filtre i sl.

`--name yolov7-custom`: Ovaj parametar postavlja naziv sesije treninga. Korišten je za nazivanje izlazne datoteke.

### 8.3. Testiranje rezultata treninga

Kao rezultat treniranja dobije se nekoliko izlaznih datoteka, od kojih je najznačajnija izlazna *weights* ili “težinska” datoteka/datoteka s težinama. Radi se o binarnoj datoteci koja sadrži naučene parametre (težine i pomake) neuronske mreže. Ovi parametri su rezultat treniranja modela na određenom skupu podataka za određeni zadatak. Datoteka je

u mom slučaju bila veličine oko 70 MB, i to je zapravo istrenirani model. Uz pomoć ove datoteke, moguće je pozvati sljedećom komandnom linijom samo prepoznavanje registarskih oznaka, što je i cilj nakon treniranja modela (8.2.):

```
python detect.py --weights yolov7_custom.pt --conf 0.5 --img-size 640 --source  
slika1.jpg --view-img --no-trace
```

**Programski kod 8.2.** Kod komandne linije za pokretanje Python skripte za detekciju objekata

`python detect.py`: Ovo je osnovna naredba za pokretanje Python skripte nazvane "detect.py" pomoću Python interpretatora. Skripta "detect.py" sadrži kod za detekciju objekata koristeći YOLOv7 model.

`--weights yolov7_custom.pt`: Ovaj parametar specificira putanju do datoteke sa spremljenim težinama prethodno opisanog vlastito istreniranog modela.

`--conf 0.8`: Ovaj parametar postavlja prag sigurnosti (engl. *confidence threshold*) za detekciju objekata na 0.8. To znači da će se prihvaćati samo detekcije s vjerojatnošću (sigurnošću) većom od 0.8, tj. 80 postotnom sigurnošću.

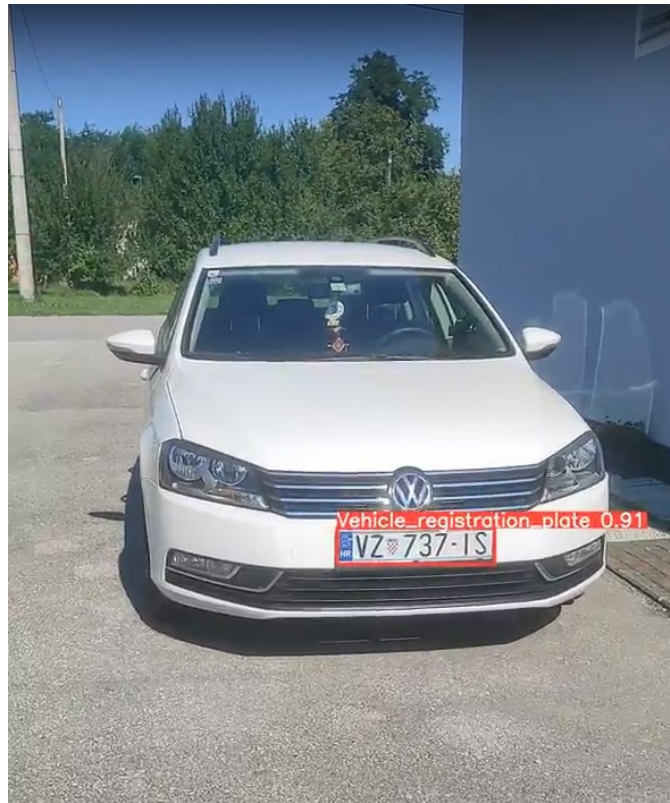
`--img-size 640`: Ovaj parametar postavlja veličinu ulaznih slika za model na 640 piksela u širini i visini. To će promijeniti veličinu ulaznih slika prije nego što se prosljede modelu za detekciju.

`--source slika1.jpg`: Ovaj parametar specificira ulazni video (ili sliku) kao izvor podataka za detekciju. U ovom slučaju, izvor je video datoteka "2.mp4". Model će primijeniti detekciju na svaki okvir tog videa.

`--view-img`: Ovaj parametar omogućuje prikazivanje slika sa detektiranim objektima tijekom izvođenja. To je korisno za praćenje rezultata detekcije u stvarnom vremenu.

`--no-trace`: Ovaj parametar onemogućuje dodavanje tragova objekata na slikama. Tragovi se često koriste za praćenje objekata kroz video, ali ovim se parametrom isključuje ta značajka.

Dobiveni rezultat je izlazna Slika 8.2, s korektno označenom registarskom oznakom - dakle postavljeni zadatak ispravno je riješen u ovom primjeru. Na slici vidljivo je kako je model sa sigurnošću od 91% (što je iznad postavljenog minimalnog praga od 80%) pronašao registracijsku oznaku na slici i označio je crvenim okvirom.

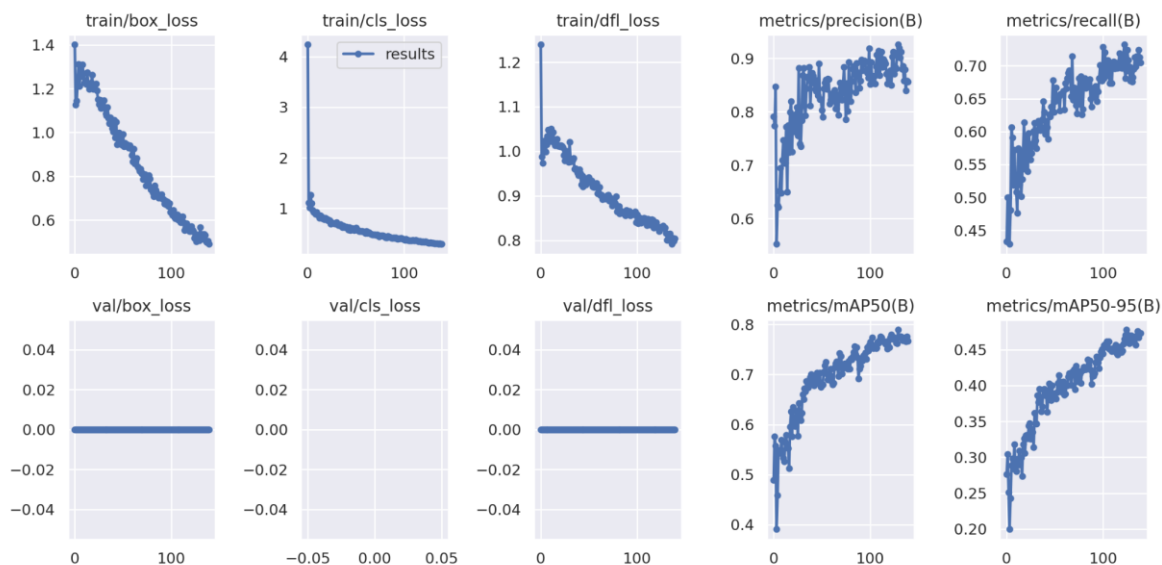


**Slika 8.2.** Rezultat treninga YOLO V7 modela

#### **8.4. Evaluacija performansi treniranog modela**

Za kvantifikaciju performansi koristio sam više mjera, uključujući preciznost detekcije (engl.*precision*), srednju točnost detekcije (mAP), i odziv (engl. *recall*). Trenirani sustav je postigao relativno dobre rezultate na svim mjerama, s preciznošću detekcije od preko 90% i mAP vrijednosti blizu 80%. To ukazuje na visoku pouzdanost i preciznost metode u identifikaciji i prepoznavanju registarskih pločica u različitim scenarijima.

Specifični rezultati mjera koje su prikazane kao izlaz prilikom treniranja modela YOLOv7 mogu se evaluirati za daljnje istraživanje i dobar su pokazatelj točnosti modela. Na Slici 8.3. prikazane su glavne mjere koje se evaluiraju u ovom procesu



**Slika 8.3.** Mjere evaluacije treniranja YOLO V7 modelom

Gubitak (engl. *Loss*): Gubitak je temeljna mjera tijekom treniranja i raščlanjuje se na nekoliko komponentata. Niža vrijednost gubitka je povoljnija. Gubitak je mjera koliko dobro predviđanja modela odgovaraju stvarnim oznakama u skupu za treniranje. Niže vrijednosti gubitka ukazuju da model radi preciznija predviđanja i da se uspješno uči iz skupa za treniranje. Različite mjere gubitka, vidljive na slici su sljedeće:

Gubitak za Regresiju Okvira (engl. *Bounding Box Regression Loss*): Mjeri koliko dobro model predviđa koordinate (x, y, širina, visina) okvira oko objekata.

Gubitak za Klasifikaciju (engl. *Classification Loss*): Mjeri točnost predviđenih klasnih oznaka za objekte.

Gubitak za Sigurnost Detekcije (engl. *Objectness Loss*): Fokusira se na ocjenu sigurnosti detekcije objekata u svakom okviru.

Ukupni Gubitak (engl. *Total Loss*): Zbroj svih pojedinačnih gubitaka, često se koristi kao glavna metrika za optimizaciju.

Vidljivo je kako model poboljšava svoje performanse tijekom treniranja, što je logično. Što više iteracija kroz čitav skup prolazi kroz treniranje, gubitak je sve manji, odnosno model je točniji.

Prosječna Srednja Preciznost (mAP): Izračunava prosječnu preciznost (AP) za svaku klasu i zatim prosječuje te vrijednosti. AP mjeri odnos preciznosti i odziva u detekciji objekata.

Preciznost i Odziv (engl. *Precision i Recall*): Preciznost mjeri koliko je predviđenih pozitivnih detekcija bilo točno, dok odziv mjeri koliko stvarnih pozitivnih instanci je ispravno detektirano.

F1-Ocjena (engl. *F1-Score*): F1-Ocjena je harmonijska sredina između preciznosti i odziva. Pruža ravnotežu između preciznosti i odziva i korisna je za nebalansirane skupove podataka.

IoU (Intersection over Union): IoU mjeri preklapanje između predviđenih okvira i stvarnih okvira objekata. Često se koristi za utvrđivanje je li detekcija ispravna ili ne.

## **8.5. Prepoznavanje znakova na registarskoj oznaci**

Nakon što je istreniran model za prepoznavanje registarskih oznaka, sljedeći korak zadatka bio je aplicirati model koji će prepoznati znakove na registarskoj oznaci. Postoje mnogi modeli za to, od plaćenih do besplatnih (otvorenog koda). Za svoj zadatak iskoristio sam model otvorenog koda EasyOCR. Dokumentacija je dostupna [26, 27].

Ovaj sustav koristi duboke neuronske mreže za obradu slika i prepoznavanje teksta. EasyOCR je sustav koji već dolazi s prethodno treniranim modelom za prepoznavanje znakova. To znači da nije potrebno dodatno trenirati model kako bi se koristio za osnovne funkcionalnosti prepoznavanja teksta na slikama. U većini slučajeva, kao i u ovom zadatku, korištenje prethodno obučenog EasyOCR modela će biti dovoljno za učinkovito prepoznavanje znakova na registarskim pločicama – Slika 8.4, 8.5, 8.6.



**Slika 8.4.** Prvi rezultat prepoznavanja znakova na registarskoj oznaci



**Slika 8.5.** Drugi rezultat prepoznavanja znakova na registarskoj oznaci



**Slika 8.6.** Treći rezultat prepoznavanja znakova na registarskoj oznaci

## 9. Zaključak

Promet na cestama je iz dana u dan sve veći, te je neophodno poboljšati sustav upravljanja prometom osiguravanjem sigurnosti prometa, prometne učinkovitosti i mobilnosti na pouzdan način. ANPR sustav je potpuno automatizirani, brzi sustav temeljen na kameri koja prati, bilježi i prijavljuje registarske pločice vozila. Napredak u ANPR tehnologiji i široka primjena mreža dubokog učenja mogu poboljšati postojeće sustave i također maksimizirati operativnu učinkovitost inteligentnih transportnih sustava.

ANPR igra glavnu ulogu u automatskom praćenju prometnih pravila i održavanju provedbe zakona na javnim cestama. ANPR se koristi za identifikaciju vozila po njihovim registarskim tablicama. Budući da svako vozilo ima jedinstvenu registarsku pločicu, nikakve vanjske kartice, oznake ili odašiljači ne moraju biti prepoznatljivi, već samo registarska pločica. Sve vidljive brojeve registarskih pločica, kao i vrijeme, datum i mjesto automatski bilježi ANPR. Informacije, koje povremeno uključuju slike vozača i putnika, kao i automobila, naknadno se prenose na glavni poslužitelj. Područje ANPR-a i njegova primjena privukli su mnoge istraživače da traže i razvijaju sustave koji mogu obraditi slike i iz njih dobiti korisne informacije.

Prednosti ANPR-a su brojne i uključuju analizu vremena putovanja, evidenciju naplatnih kućica, sigurnost maloprodajnih parkova i upravljanje prometom. Kao rezultat toga, razumljivo je zašto je ANPR u posljednje vrijeme stekao toliku popularnost. Tržište za automatsko prepoznavanje registarskih pločica brzo se širi i mogu se vidjeti inicijative u sektorima kao što su javni sektor, građevina, pametni gradovi, logistika, transport i sigurnost.

Upotreba automatskog prepoznavanja registarskih pločica izvrstan je način za zamjenu ručnog rada sofisticiranim sustavom koji daje rezultate brzo i učinkovito. Ova tehnologija ima širok raspon primjena, što podržava sve veću potražnju za njom.



## 10. Literatura

1. B. Rashed, H. Abd El Kader, H. Rafaat, M. Sharaf i M. Shehata: Automatic Number Plate Recognition, *Int J Sci Res Publ*, Vol. 3, no. 12, 2018.
2. S.-L. Chang, L.-S. Chen, Y.-C. Chung i S.-W. Chen: Automatic license plate recognition, *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 1, 2004., str. 42-53.
3. C. Patel, D. Shah i A. Patel: Automatic Number Plate Recognition System (ANPR): A Survey, *International Journal of Computer Applications*, Vol. 69, no. 9, 2013.,str. 21-33.
4. S. Sahil, A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar i Brian Lee: A survey of modern deep learning based object detection models, *Digital Signal Processing*, Vol. 126, 103514, 2022, str. 1-19.
5. Pragati Baheti: A Comprehensive Guide to Convolutional Neural Networks, <https://www.v7labs.com/blog/convolutional-neural-networks-guide>, dostupno 20.4.2023.
6. W. Zhiqiang i L. Jun: A review of object detection based on convolutional neural network, 2017 36th Chinese Control Conference (CCC), Dalian, China, 2017., str. 11104-11109
7. M.Z. Alom, T.M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M.S. Nasrin, M. Hasan, B.C. Van Essen, A.A.S. Awwal, V.K. Asari: A State-of-the-Art Survey on Deep Learning Theory and Architectures, *Electronics*, 2019, Vol. 8, no. 3:292., <https://www.mdpi.com/2079-9292/8/3/292>, dostupno: 20.4.2023.
8. Y. Luo, J. Zhu, M. Li, Y. Ren and B. Zhang: Smooth Neighbors on Teacher Graphs for Semi-Supervised Learning, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018., str. 8896-8905.
9. S. Russell i P. Norvig: *Artificial Intelligence: A Modern Approach (Fourth Edi)*, Pearson Education, 2021.

10. Koustubh Prem, Develop a Number Plate Scanner using MATLAB <https://www.skyfilabs.com/project-ideas/number-plate-scanner-using-matlab>, dostupno 21.4.2023.
11. S. Tripathi, S. Jain, S. Shetty i V. Sharma: Automatic Number Plate Recognition System : The Implementation, International journal of Innovative technology and exploring engineering, Vol. 10, no. 8, 2021.
12. Police.UK: Automatic Number Plate Recognition (ANPR), <https://www.police.uk/advice/advice-and-information/rs/road-safety/automatic-number-plate-recognition-anpr/>, dostupno: 21.04.2023.
13. Staqu: Top Use Cases of Automatic Number Plate Recognition(ANPR) Technology of AI-Powered Video Analytics, <https://www.staqu.com/top-use-cases-of-automatic-number-plate-recognitionanpr-technology-of-ai-powered-video-analytics/>, dostupno: 21.04.2023.
14. Viso: Automatic Number Plate Recognition (ANPR), <https://viso.ai/computer-vision/automatic-number-plate-recognition-anpr/>, dostupno: 23.04.2023.
15. EFF: Automated License Plate Readers (ANPRs), <https://www.eff.org/pages/automated-license-plate-readers-ANPR>, dostupno: 23.04.2023.
16. Alert Systems: How ANPR Works, <https://www.alertsystems.co.uk/security-solutions/anpr-automatic-number-plate-recognition/how-anpr-works/>, dostupno 24.04.2023.
17. Optophone, Wikipedia.com, <https://en.wikipedia.org/wiki/Optophone> , dostupno 25.4.2023.
18. PaddleOCR Quick Start: [https://github.com/PaddlePaddle/PaddleOCR/blob/release/2.5/doc/doc\\_en/quickstart\\_en.md](https://github.com/PaddlePaddle/PaddleOCR/blob/release/2.5/doc/doc_en/quickstart_en.md), dostupno 26.4.2023.
19. OpenCV: About, <https://opencv.org/about/>, dostupno 27.4.2023.
20. Viso: What is OpenCV? The Complete Guide (2023), <https://viso.ai/computer-vision/opencv/>, dostupno 27.4.2023.
21. S. Varsheni: Evaluate Your Model – Metrics for Image Classification and Detection, <https://www.analyticsvidhya.com/blog/2021/06/evaluate-your-model-metrics-for-image-classification-and-detection/>, dostupno: 28.4.2023.

22. Z. -Q. Zhao, P. Zheng, S. -T. Xu and X. Wu: Object Detection With Deep Learning: A Review, IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, 2019. str. 3212-3232.
23. R. Khandelwal: Evaluating performance of an object detection model, Izvor: <https://towardsdatascience.com/evaluating-performance-of-an-object-detection-model-137a349c517b>, dostupno 29.4.2023.
24. K. E. Koech: Object Detection Metrics With Worked Example, Izvor: <https://towardsdatascience.com/on-object-detection-metrics-with-worked-example-216f173ed31e>, dostupno 30.4.2023.
25. C. -Y. Wang, A. Bochkovskiy i H. -Y. M. Liao: YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors, 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Kanada, 2023., str. 7464-7475
26. Easy OCR, <https://github.com/JaidedAI/EasyOCR>, dostupno: 25.5.2023.
27. Easy OCR. <https://www.jaided.ai/easyocr/documentation/>, dostupno 25.5.2023.

## 11. Popis slika

**Slika 2.1.** Shematski prikaz arhitekture jednostavne konvolucijske neuronske mreže, Izvor: [https://www.researchgate.net/publication/336805909\\_A\\_High-Accuracy\\_Model\\_Average\\_Ensemble\\_of\\_Convolutional\\_Neural\\_Networks\\_for\\_Classification\\_of\\_Cloud\\_Image\\_Patches\\_on\\_Small\\_Datasets](https://www.researchgate.net/publication/336805909_A_High-Accuracy_Model_Average_Ensemble_of_Convolutional_Neural_Networks_for_Classification_of_Cloud_Image_Patches_on_Small_Datasets)

**Slika 4.1.** Prepoznavanje registarskih pločica automobila, Izvor: <https://www.skyfilabs.com/project-ideas/number-plate-scanner-using-matlab>

**Slika 4.2.** Način rada ANPR-a na parkiralištu, Izvor: <https://www.nortechcontrol.com/solutions/vehicle/vehicle-access-with-counting-using-anpr>

**Slika 4.3.** Inteligentno upravljanje prometom, Izvor: <https://viso.ai/computer-vision/automatic-number-plate-recognition-anpr/>

**Slika 4.4.** Stacionarne ANPR kamere, Izvor: <https://padailypost.com/2022/10/24/council-oks-cameras-that-read-license-plates-to-stop-crime/>

**Slika 4.5.** Mobilne ANPR kamere, Izvor: <https://www.quora.com/Do-police-ever-scan-number-plates-of-cars-going-into-McDonalds-drive-throughs-to-locate-criminals>

**Slika 4.6.** Detekcija objekata za vozila temeljena na dubokom učenju, Izvor: <https://viso.ai/edge-ai/google-coral/>

**Slika 4.7.** Primjer OCR-a, Izvor: [https://github.com/PaddlePaddle/PaddleOCR/blob/release/2.5/doc/doc\\_en/quickstart\\_en.md](https://github.com/PaddlePaddle/PaddleOCR/blob/release/2.5/doc/doc_en/quickstart_en.md)

**Slika 5.1.** OpenCV s demo kodom detektora objekata C++ za prepoznavanje lica, Izvor: <https://viso.ai/deep-learning/face-detection-overview/>

**Slika 6.1.** Matrica zabune

**Slika 6.2.** TPR i FPR u odnosu na raspon vrijednosti praga, Izvor:

<https://ichi.pro/th/khumux-reim-tn-sahrab-roc-curve-laea-auc-70748239405639>

**Slika 6.3.** Prikaz IoU, Izvor: <https://www.v7labs.com/blog/yolo-object-detection>

**Slika 6.4.** Distribucija broja slika u skupu podataka Pascal VOC, izvor:

<https://www.sciencedirect.com/science/article/pii/S1051200422001312>

**Slika 6.5.** Distribucija broja slika u skupu podataka MS -COCO, izvor:

<https://www.sciencedirect.com/science/article/pii/S1051200422001312>

**Slika 6.6.** Distribucija broja slika u skupu podataka ImageNet, izvor:

<https://www.sciencedirect.com/science/article/pii/S1051200422001312>

**Slika 6.7.** Distribucija broja slika s obzirom na različite klase u skupu podataka Open

Images, izvor: <https://www.sciencedirect.com/science/article/pii/S1051200422001312>

**Slika 6.8.** Planirani RepConv 3-složni ELAN, Izvor:

[https://openaccess.thecvf.com/content/CVPR2023/papers/Wang\\_YOLOv7\\_Trainable\\_Bag-of-Freebies\\_Sets\\_New\\_State-of-the-Art\\_for\\_Real-Time\\_Object\\_Detectors\\_CVPR\\_2023\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2023/papers/Wang_YOLOv7_Trainable_Bag-of-Freebies_Sets_New_State-of-the-Art_for_Real-Time_Object_Detectors_CVPR_2023_paper.pdf)

**Slika 6.9.** Obrnuti CSPDarknet, Izvor:

[https://openaccess.thecvf.com/content/CVPR2023/papers/Wang\\_YOLOv7\\_Trainable\\_Bag-of-Freebies\\_Sets\\_New\\_State-of-the-Art\\_for\\_Real-Time\\_Object\\_Detectors\\_CVPR\\_2023\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2023/papers/Wang_YOLOv7_Trainable_Bag-of-Freebies_Sets_New_State-of-the-Art_for_Real-Time_Object_Detectors_CVPR_2023_paper.pdf)

**Slika 7.1.** Usporedba YOLO s drugim detektorima objekata u stvarnom vremenu, Izvor:

[https://openaccess.thecvf.com/content/CVPR2023/papers/Wang\\_YOLOv7\\_Trainable\\_Bag-of-Freebies\\_Sets\\_New\\_State-of-the-Art\\_for\\_Real-Time\\_Object\\_Detectors\\_CVPR\\_2023\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2023/papers/Wang_YOLOv7_Trainable_Bag-of-Freebies_Sets_New_State-of-the-Art_for_Real-Time_Object_Detectors_CVPR_2023_paper.pdf)

**Slika 7.2.** Arhitektura YOLO sustava, Izvor:

[https://openaccess.thecvf.com/content/CVPR2023/papers/Wang\\_YOLOv7\\_Trainable\\_Bag-of-Freebies\\_Sets\\_New\\_State-of-the-Art\\_for\\_Real-Time\\_Object\\_Detectors\\_CVPR\\_2023\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2023/papers/Wang_YOLOv7_Trainable_Bag-of-Freebies_Sets_New_State-of-the-Art_for_Real-Time_Object_Detectors_CVPR_2023_paper.pdf)

**Slika 8.1.** LabelImg

**Slika 8.2.** Rezultat treninga YOLO V7 modela

**Slika 8.3.** Mjere evaluacije treninga YOLO V7 modelom

**Slika 8.4.** Prvi rezultat prepoznavanja znakova na registarskoj oznaci

**Slika 8.5.** Drugi rezultat prepoznavanja znakova na registarskoj oznaci

**Slika 8.6.** Treći rezultat prepoznavanja znakova na registarskoj oznaci

## 12. Popis tablica

**Tablica 6.1.** Usporedba najsvremenijih detektora objekata, izvor:

[https://openaccess.thecvf.com/content/CVPR2023/papers/Wang\\_YOLOv7\\_Trainable\\_Bag-of-Freebies\\_Sets\\_New\\_State-of-the-Art\\_for\\_Real-Time\\_Object\\_Detectors\\_CVPR\\_2023\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2023/papers/Wang_YOLOv7_Trainable_Bag-of-Freebies_Sets_New_State-of-the-Art_for_Real-Time_Object_Detectors_CVPR_2023_paper.pdf)

**Tablica 6.2.** Ispitivanje parametara predloženog skaliranja modela, izvor:

[https://openaccess.thecvf.com/content/CVPR2023/papers/Wang\\_YOLOv7\\_Trainable\\_Bag-of-Freebies\\_Sets\\_New\\_State-of-the-Art\\_for\\_Real-Time\\_Object\\_Detectors\\_CVPR\\_2023\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2023/papers/Wang_YOLOv7_Trainable_Bag-of-Freebies_Sets_New_State-of-the-Art_for_Real-Time_Object_Detectors_CVPR_2023_paper.pdf)

**Tablica 6.3.** Studija ablacije na planiranom modelu RepResidual, izvor:

[https://openaccess.thecvf.com/content/CVPR2023/papers/Wang\\_YOLOv7\\_Trainable\\_Bag-of-Freebies\\_Sets\\_New\\_State-of-the-Art\\_for\\_Real-Time\\_Object\\_Detectors\\_CVPR\\_2023\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2023/papers/Wang_YOLOv7_Trainable_Bag-of-Freebies_Sets_New_State-of-the-Art_for_Real-Time_Object_Detectors_CVPR_2023_paper.pdf)

**Tablica 6.4.** Istraživanje parametara za predloženu pomoćnu glavu, izvor:

[https://openaccess.thecvf.com/content/CVPR2023/papers/Wang\\_YOLOv7\\_Trainable\\_Bag-of-Freebies\\_Sets\\_New\\_State-of-the-Art\\_for\\_Real-Time\\_Object\\_Detectors\\_CVPR\\_2023\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2023/papers/Wang_YOLOv7_Trainable_Bag-of-Freebies_Sets_New_State-of-the-Art_for_Real-Time_Object_Detectors_CVPR_2023_paper.pdf)

**Tablica 6.5.** Istraživanje parametara za predloženu ograničenu pomoćnu glavu, izvor:

[https://openaccess.thecvf.com/content/CVPR2023/papers/Wang\\_YOLOv7\\_Trainable\\_Bag-of-Freebies\\_Sets\\_New\\_State-of-the-Art\\_for\\_Real-Time\\_Object\\_Detectors\\_CVPR\\_2023\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2023/papers/Wang_YOLOv7_Trainable_Bag-of-Freebies_Sets_New_State-of-the-Art_for_Real-Time_Object_Detectors_CVPR_2023_paper.pdf)

**Tablica 6.6.** Istraživanje parametara za predloženu pomoćnu glavu, izvor:

[https://openaccess.thecvf.com/content/CVPR2023/papers/Wang\\_YOLOv7\\_Trainable\\_Bag-of-Freebies\\_Sets\\_New\\_State-of-the-Art\\_for\\_Real-Time\\_Object\\_Detectors\\_CVPR\\_2023\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2023/papers/Wang_YOLOv7_Trainable_Bag-of-Freebies_Sets_New_State-of-the-Art_for_Real-Time_Object_Detectors_CVPR_2023_paper.pdf)