

Detekcija loga u digitalnom marketingu koristeći modele dubokog učenja

Jelušić, Matea

Master's thesis / Diplomski rad

2024

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:122:296343>

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

Download date / Datum preuzimanja: **2024-11-15**

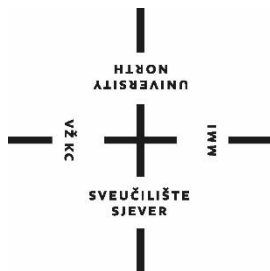


Repository / Repozitorij:

[University North Digital Repository](#)



SVEUČILIŠTE SJEVER
SVEUČILIŠNI CENTAR VARAŽDIN



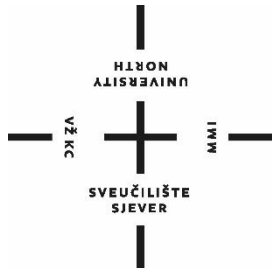
DIPLOMSKI RAD br. 136-MMD-2024

DETEKCIJA LOGA U DIGITALNOM
MARKETINGU KORISTEĆI MODELE DUBOKOG
UČENJA

Matea Jelušić

Varaždin, rujan 2024.

SVEUČILIŠTE SJEVER
SVEUČILIŠNI CENTAR VARAŽDIN
Studij Multimedija



DIPLOMSKI RAD br. 136-MMD-2024

DETEKCIJA LOGA U DIGITALNOM
MARKETINGU KORISTEĆI MODELE DUBOKOG
UČENJA

Student:

Matea Jelušić

Mentor:

izv. prof. dr. sc. Emil Dumić

Varaždin, rujan 2024.

Prijava diplomskog rada

Definiranje teme diplomskog rada i povjerenstva

ODJEL	Odjel za multimediju		
STUDIJ	diplomski sveučilišni studij Multimedija		
PRISTUPNIK	Jelušić Matea	JMBAG	0130332888
DATUM	26.08.2024.	KOLEGIJ	Računalni vid
NASLOV RADA	Detekcija loga u digitalnom marketingu koristeći modele dubokog učenja		
NASLOV RADA NA ENGL. JEZIKU	Logo detection in digital marketing 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. art. dr. sc. Robert Geček - član		
	3. izv. prof. dr. sc. Emil Dumić - mentor		
	4. doc. dr. sc. Andrija Bernik - zamjenski član		
	5.		

Zadatak diplomskog rada

BROJ	136-MMD-2024		
OPIS	<p>U ovom radu će biti opisani i ispitani različiti modeli dubokog učenja za detekciju loga od različitih brendova, kao jedna od mogućnosti korištenja umjetne inteligencije u marketingu.</p> <p>Glavni zadatak detekcije loga je odrediti lokaciju određenog loga na slikama (ili videozapisima) i identificirati ih. Iako se može smatrati posebnim tipom detekcije objekata, otkrivanje loga u slikama stvarnog svijeta može biti prilično izazovno jer brojne marke mogu imati vrlo različite kontekste, raznolike skale, promjene u osvjetljenju, veličini, razlučivosti i druge deformacije. U novije vrijeme, rješenja detekcije objekata temeljena na dubokom učenju s mogućnošću jasnijeg predstavljanja značajki nude bolju robusnost, točnost, brzinu i time privlače sve veću pozornost. U teoretskom dijelu diplomskog rada bit će opisane neke od metoda za detekciju objekata s naglaskom na modele dubokog učenja: konvolucijske neuronske mreže bazirane na pretraživanju regija (R-CNN, Fast R-CNN, Faster R-CNN) te različite verzije YOLO detektora modela (koji spadaju u SSD modele) s naglaskom na YOLOv7. Potom će se opisati različite metode detekcije loga bazirane na ranije opisanim detektorima objekata. Opisat će se mjere za usporedbu metoda: za klasifikaciju (preciznost, odziv, točnost, F1 mjera) i za detekciju (IoU, AP i mAP).</p> <p>U praktičnom dijelu rada će se koristiti YOLOv7 model za detekciju loga. Bit će korištena neka od postojećih baza loga za trening. Koristit će se YOLOv7 model prilagođen za detekciju loga i uspoređeni koristeći bazu za test te mjerama za evaluaciju detekcije objekata.</p>		
ZADATAK URUČEN	02.09.2024.	POTPIS MENTORA	Emil Dumić



IZJAVA O AUTORSTVU

Završni/diplomski/specijalistički 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, Matea Jelušić pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključiva autorica diplomskog rada pod naslovom Detekcija loga u digitalnom marketingu koristeći modele dubokog učenja te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:
Matea Jelušić



(vlastoručni potpis)

Sukladno članku 58., 59. i 61. Zakona o visokom obrazovanju i znanstvenoj djelatnosti završne/diplomske/specijalističke radove sveučilišta su dužna objaviti u roku od 30 dana od dana obrane na nacionalnom repozitoriju odnosno repozitoriju visokog učilišta.

Sukladno članku 111. Zakona o autorskom pravu i srodnim pravima student se ne može protiviti da se njegov završni rad stvoren na bilo kojem studiju na visokom učilištu učini dostupnim javnosti na odgovarajućoj javnoj mrežnoj bazi sveučilišne knjižnice, knjižnice sastavnice sveučilišta, knjižnice veleučilišta ili visoke škole i/ili na javnoj mrežnoj bazi završnih radova Nacionalne i sveučilišne knjižnice, sukladno zakonu kojim se uređuje umjetnička djelatnost i visoko obrazovanje.

Sažetak

U ovom radu, istražene su različite metode detekcije logotipa u digitalnim slikama, s posebnim naglaskom na modele dubokog učenja koje uključuju konvolucijske neuronske mreže bazirane na jednofaznim i dvofaznim modelima. Kroz praktičan zadatak, analizirani su različiti parametri koji utječu na konačne rezultate detekcije. Proveden je proces treniranja i testiranja YOLOv7 modela za detekciju logotipa. Evaluacija rezultata otkrila je određene prednosti i nedostatke koji se javljaju tijekom procesa detekcije. Cilj rada je pružiti bolje razumijevanje teme detekcije logotipa, njezinih izazova s kojima se suočava kao i istražiti mogućnosti primjene, osobito u kontekstu digitalnog marketinga, gdje ima potencijal postati ključni alat u optimizaciji marketinških strategija i analizi tržišta.

Ključne riječi: detekcija logotipa, digitalni marketing, duboko učenje, konvolucijske neuronske mreže, YOLO

Summary

This paper explores various methods for logo detection in digital images, with a particular focus on deep learning models, including convolutional neural networks based on single-stage and two-stage models. Through a practical task, different parameters that influence the final detection results were analyzed. The process of training and testing of the YOLOv7 model for logo detection was conducted. The evaluation of results revealed certain strengths and weaknesses that arise during the detection process. The aim of this paper is to provide a better understanding of logo detection, the challenges it faces, and to explore its potential applications, particularly in the context of digital marketing, where it has the potential to become a key tool in optimizing marketing strategies and market analysis.

Keywords: logo detection, digital marketing, deep learning, convolutional neural networks, YOLO

Sadržaj

1. Uvod.....	1
1.1. Logotip.....	1
1.2. Detekcija logotipa.....	2
1.3. Predmet i cilj rada.....	3
1.4. Metode istraživanja i struktura rada	3
2. Modeli dubokog učenja za detekciju logotipa.....	4
2.1. Konvolucijske neuronske mreže (CNN).....	4
2.2. Arhitektura konvolucijskih neuronskih mreža	5
2.3. Primjeri konvolucijskih neuronskih mreža.....	7
2.4. Dvofazni modeli	8
2.5. R-CNN.....	8
2.5.1. Brzi R-CNN	10
2.5.2. Brži R-CNN	11
2.6. Jednofazni modeli	12
2.7. SSD detektori objekata	12
2.8. YOLO	13
3. Metoda detekcije logotipa	22
3.1. Baze za učenje	24
3.2. Mjere za uspješnost detekcije	25
3.3. Mjere usporedbe kod detekcije.....	28
4. Praktični zadatak	31
4.1. Odabir arhitekture.....	31
4.2. Odabir baze za učenje i anotacija	31
4.3. Podjela baze u tri dijela (treniranje/validacija/testiranje).....	32
4.4. Treniranje modela.....	32

4.5. Evaluacija rezultata treniranja modela.....	34
4.6. Testiranje modela.....	37
4.7. Evaluacija rezultata testiranja modela	38
5. Diskusija.....	42
5.1. Primjena detekcije logotipa u digitalnom marketingu.....	42
5.2. Izazovi u detekciji logotipa.....	45
5.3. Budućnost detekcije logotipa.....	46
6. Zaključak.....	49

1. Uvod

Digitalni marketing pojavio se krajem prošlog desetljeća kao nastavak na direktni marketing te obuhvaća širok spektar marketinških aktivnosti koje se provode putem digitalnih kanala kako bi se promovirali proizvodi ili usluge [1]. Slike su u posljednjih nekoliko godina postale dominantna sila u digitalnom marketingu. Osim što se koriste kako bi privukle pažnju, potiču emocije te oblikuju prvi dojam potrošača o određenom proizvodu ili brendu. Stoga ne čudi da objave koje se svakodnevno dijele na popularnim društvenim mrežama fokusiranim prvenstveno na vizualni sadržaj (npr. Instagram, Facebook, TikTok), pružaju izvanredne mogućnosti za razvoj prepoznatljivosti brenda i olakšavaju oglašavanje putem interneta.

1.1. Logotip

Sušтина vizualnog sadržaja u digitalnom marketingu leži u njegovoj sposobnosti da uspješno komunicira s publikom koristeći se vizualnim elementima kako bi izrazio određene ideje ili poruke. Logotip, odnosno vizualni identitet nekog brenda ili proizvoda, možemo definirati kao grafički simbol koji se sastoji od stiliziranog teksta, slike ili kombinacije tih elemenata te predstavlja tvrtku, brend, proizvod, uslugu ili organizaciju [2]. Glavni mu je zadatak privući pažnju i ostaviti trajan utisak, a njegovo pojavljivanje u digitalnom prostoru od velike je važnosti za sveukupnu promidžbu brenda. Analizom konteksta u kojem se logotip pojavljuje, profitne i neprofitne organizacije mogu dobiti uvid u ponašanje i stavove korisnika te otkriti kako ih korisnici percipiraju i kako s njima interaktivno djeluju. Konkretno, brend sportske opreme može analizirati slike na društvenim mrežama poput Instagrama kako bi dublje razumio gdje se sve njegov logotip pojavljuje i kako korisnici percipiraju brend. Primjerice, ako primijete da se njihov logotip često pojavljuje u slikama vezanim uz aktivnosti na otvorenom poput planinarenja ili trčanja, mogu zaključiti da potrošači njihov brend povezuju s aktivnim načinom života i prirodom. S druge pak strane, ako primijete da se logotip često pojavljuje uz slike sportskih događaja ili natjecanja, mogu shvatiti da njihov brend ima jaku prisutnost u sportskim krugovima. Analizirajući takve kontekste, brendovi mogu prilagoditi svoje marketinške kampanje kako bi bolje ciljali određenu publiku, stvarajući sadržaj i poruke koje će se bolje povezivati s interesima i potrebama njihovih potrošača. Koristeći informacije dobivene iz analize slika na društvenim mrežama, mogu poboljšati učinkovitost svojih marketinških strategija i bolje odgovoriti na potrebe svoje ciljane publike.

1.2. Detekcija logotipa

Detekcija logotipa podrazumijeva lociranje i identifikaciju logotipa unutar slike, odnosno pretraživanje slike kako bi se pronašle određene regije koje sadrže logotipe te identificiranje, odnosno razlikovanje tih istih logotipa od ostalih objekta ili pozadine. Ovaj proces može biti izazovan jer se mnogi logotipovi često pojavljuju u različitim veličinama, razlučivosti, rotacijama, osvjetljenjima, pa čak i deformacijama, što znatno otežava njihovo ispravno prepoznavanje (Slika 1.1). Također, slike na društvenim mrežama često sadrže složene pozadine i vizualni šum što dodatno može ometati detekciju logotipa. Mnogi prijašnji radovi koji se bave detekcijom logotipa koristili su ručno odabrane značajke poput SIFT algoritma (engl. *Scale-Invariant Feature Transform*) kod kojeg se ključne točke u slici (npr. rubovi, uglovi ili promjene intenziteta boje) opisuju na način koji ostaje konzistentan čak i ako se slika mijenja u veličini, rotaciji ili osvjetljenju. Takav pristup nije pouzdan kada se suočava s velikim brojem logotipa. U posljednjih nekoliko godina, rješenja temeljena na dubokom učenju pokazuju bolju pouzdanost, preciznost i brzinu u detekciji logotipa zahvaljujući velikim bazama podataka, strategijama dubokog učenja, arhitekturi neuronskih mreža i funkcijama gubitka (funkcije koje se koriste za procjenu točnosti modela) [3].



Slika 1.1. Primjer detekcije logotipa Starbucks pod različitim kutovima promatranja (https://www.researchgate.net/figure/Logo-detection-through-our-Two-Layer-system_fig1_321325807, 2024.)

1.3. Predmet i cilj rada

Predmet ovog diplomskog rada je istraživanje i analiza različitih metoda detekcije logotipa u digitalnim slikama s naglaskom na modele dubokog učenja (konvolucijske neuronske mreže bazirane na jednofaznim i dvofaznim modelima). Cilj je pružiti dublje razumijevanje izazova i potencijalnih rješenja u području detekcije logotipa, posebice u primjeni detekcije logotipa u digitalnom marketingu te pružiti smjernice za daljnji razvoj i primjenu ove tehnologije u marketinškoj praksi.

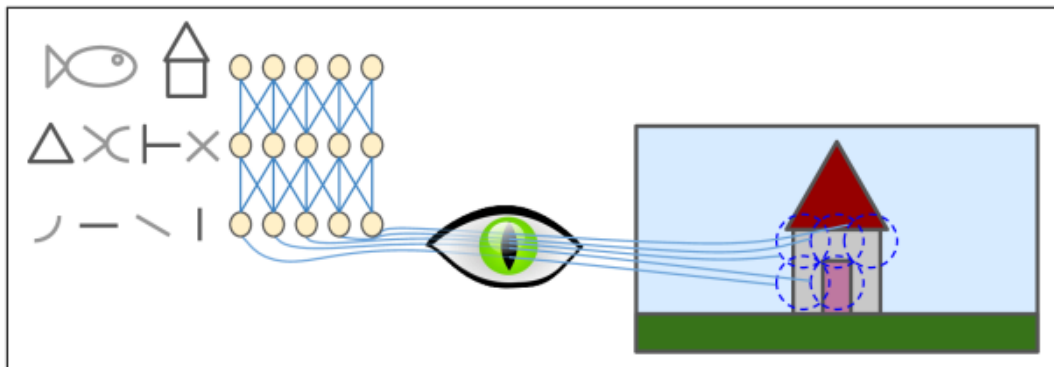
1.4. Metode istraživanja i struktura rada

Za potrebe izrade ovog rada korištena je relevantna znanstvena i stručna literatura iz domene računalnog vida. Rad je podijeljen na šest poglavlja. Nakon uvodnog predstavljanja teme, svrhe i cilja rada, slijedi drugo poglavlje koje obuhvaća pregled različitih modela dubokog učenja za detekciju logotipa. Treće poglavlje analizira metode korištene za evaluaciju i mjerenje uspješnosti detekcije logotipa. U četvrtom dijelu se detaljno opisuje implementacija jednog od modela, YOLOv7. Peto poglavlje naglašava primjenu detekcije u digitalnom marketingu te raspravlja o izazovima i mogućnostima za daljnje napredovanje u kontekstu postavljenih ciljeva. Končano, šesto poglavlje sažima istraživanje te donosi konačne zaključke.

2. Modeli dubokog učenja za detekciju logotipa

2.1. Konvolucijske neuronske mreže (CNN)

Konvolucijske neuronske mreže (engl. *convolutional neural networks, CNN*) nastale su osamdesetih godina prošlog stoljeća na temelju različitih istraživanja vizualnog korteksa mozga. Znanstvenici David H. Hubel i Torsten Wiesel krajem pedesetih godina uspjeli su razjasniti kako neuroni u vizualnom korteksu reagiraju na specifične vizualne podražaje. Naime, istraživanja su pokazala da mnogi neuroni u vizualnom korteksu sadrže manja lokalno receptivna polja, što znači da reagiraju samo na određene vizualne podražaje koji se nalaze u ograničenom prostoru tih vizualnih polja (Slika 2.1). Receptivna polja se međusobno preklapaju te tako pokrivaju cijelo vizualno polje. Također, primijetili su da određeni neuroni reagiraju samo na prizore linija horizontalnih orijentacija, dok drugi reagiraju na linije različitih orijentacijama (npr. vertikalne ili dijagonale), pri čemu dva neurona mogu imati isto receptivno polje, ali reagirati na sasvim različite orijentacije linija. Proučavanjem su saznali da neki neuroni sadrže veće receptivno polje i mogu reagirati na složenije uzorke koji se sastoje od više pojedinačnih i jednostavnijih uzoraka. Time se utvrdilo da se neuroni mogu podijeliti na neurone više i niže razine [4].



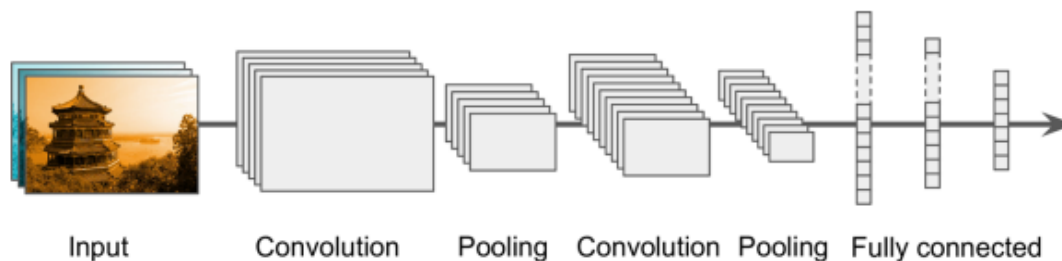
Slika 2.1. Primjer arhitekture neurona. Žuti krugovi predstavljaju pojedine neurone, dok isprekidani krugovi predstavljaju lokalna receptivna polja. Svaki neuron je povezan samo s nekoliko neurona iz prethodnog sloja, što određuje da li spadaju u neurone više ili niže razine. (Geron, 2019.)

Ovi rezultati su 1980. godine potaknuli razvoj neokognitrona (engl. *neocognitron*), preteče konvolucijske neuronske mreže kakve danas poznajemo. Upravo zato što su razvijene na temelju vizualnog korteksa, učinkovito prepoznaju i obrađuju vizualne podatke organizirane u obliku mreže ili rešetke. Jedna od najčešćih primjena konvolucijskih mreža je u detekciji i klasifikaciji objekata u slikama. S obzirom na to da uspješno mogu identificirati prisutnost i

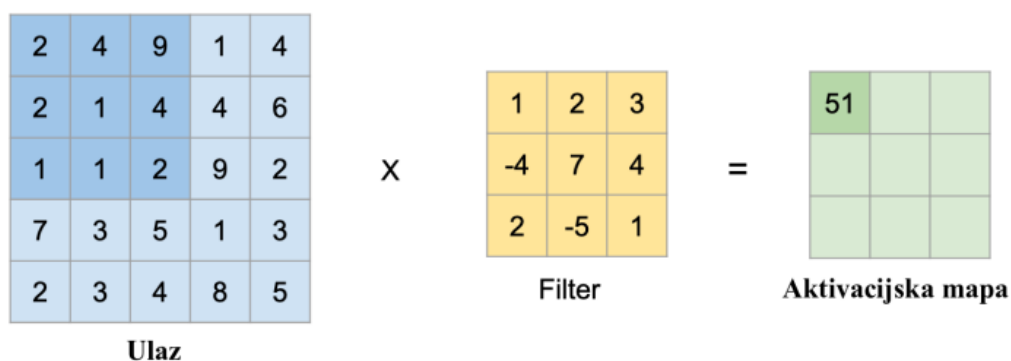
položaj objekta unutar neke slike, često se koriste u detekciji automobila, pješaka i prometnih znakova na cesti. Ova sposobnost je ključna za autonomna vozila, sustave nadzora sigurnosti i robotiku. Osim toga, mogu se koristiti za prepoznavanje lica, otisaka prstiju i drugih biometrijskih obrazaca što je od izrazite važnosti u područjima medicine i forenzike.

2.2. Arhitektura konvolucijskih neuronskih mreža

Osnovna arhitektura konvolucijskih neuronskih mreža temelji se na nizu slojeva koji obuhvaćaju konvolucijske slojeve i slojeve sažimanja (Slika 2.2). Na samom početku, ulazni sloj prima i zadržava informacije o pikselima ulazne slike, uključujući dimenzije slike poput širine, visine i dubinu slike. Dubina slike odnosi se na broj kanala slike, kao što su crvena, zelena i plava kod RGB slika, pri čemu svaki kanal nosi informacije o određenoj boji. Slika prolazi kroz konvolucijski sloj gdje se na njoj primjenjuje određen broj filtera s ciljem prepoznavanja obrasca (npr. rubova, oblika, teksture, objekta) te se stvara izlazna slika zvana mapa značajki (engl. *feature map*). Filter je matrica pravokutnih dimenzija (obično 3x3 ili 5x5) koja postepeno prolazi preko slike po dužini i širini (engl. *stride*) te izvlači obrasce. Svaki put kad se filter primijeni na određenu poziciju slike, izračunava se skalarni umnožak između podatka unutar filtera i ulaza na toj poziciji unutar slike (Slika 2.3) [5].

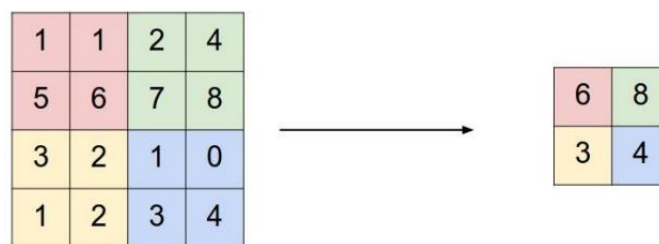


Slika 2.2. Arhitektura konvolucijske neuronske mreže (Geron, 2019.)



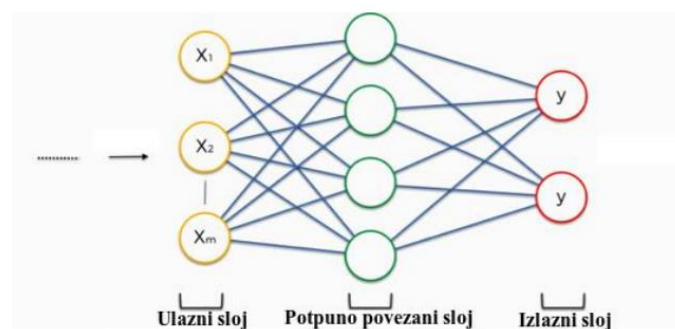
Slika 2.3. Grafički prikaz rada filtera. (Vlahović, 2020.)

Taj skalarni proizvod predstavlja težinu filtera koja se množi s pikselom te se u konačnici rezultati zbrajaju kako bi se dobila nova vrijednost za svaki piksel u izlaznoj slici. Na taj način stvara se dvodimenzionalna mapa značajki koja daje odgovore filtra za različite dijelove slike. U inicijalnom konvolucijskom sloju mreža uči filtre detektirati rubove i promjenu boje dok svaki slijedeći konvolucijski sloj spaja te spoznaje u složenije oblike poput krugova, kocku i sl. Što je više konvolucijskih slojeva, to se više složenijih obrasca mogu ispravno detektirati. Između više konvolucijskih slojeva pojavljuje se sloj sažimanja (engl. *pooling layer*). Glavni mu je zadatak smanjiti dimenzionalnost podataka mape značajki kako bi se smanjilo opterećenje računala prilikom računanja sljedećeg konvolucijskog sloja, smanjiti upotrebu memorije i broj parametara (time se također smanjuje rizik od *preinaučivosti* modela) [4]. To se postiže grupiranjem podataka iz mapa značajki prethodnog sloja i uzimanjem srednje vrijednosti (maksimalne ili minimalne) unutar svakog područja. Najčešća forma filtera u sloju sažimanja je 2x2, te se na taj način odbacuju 75% podataka (Slika 2.4).



Slika 2.4. Primjer postavljanja maksimalne vrijednosti (Vlahović, 2020.)

Naposlijetku, nakon što su podaci maksimalno smanjeni i sažeti jer su prošli kroz određene slojeve konvolucije i sažimanja, matrica se pretvori u *ID vektor* (Slika 2.5) te se potom koristi *potpuno povezani sloj* (engl. *fully connected layer*) u kojem su sažeti podaci povezani sa svim neuronima. To omogućuje mreži da uči složenije obrasce i donese konačne zaključke. Ovaj sloj je ključan jer omogućuje povezivanje informacija iz prethodnih slojeva potrebnih za rješavanje određenog zadatka, npr. klasifikaciju ulaznih slika (ili detekciju ako bi se klasificirala regija po regija slike).



Slika 2.5. Grafički prikaz potpuno povezanog sloja (Vlahović, 2020.)

2.3. Primjeri konvolucijskih neuronskih mreža

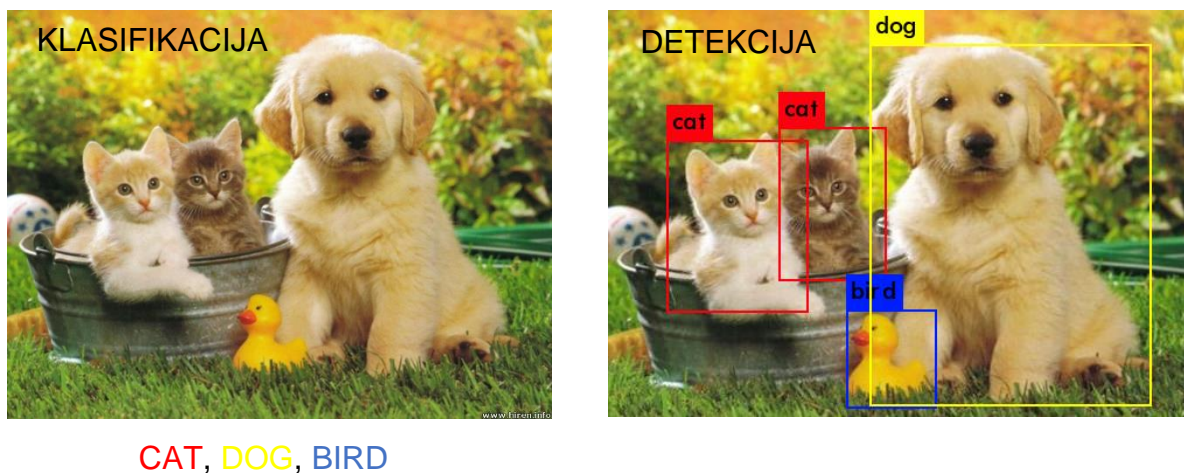
Ovisno o složenosti zadatka razvile su se različite arhitekture konvolucijske neuronske mreže, a među najpoznatijima su *LeNet*, *AlexNet*, *GoogLeNet* i *ResNet* (Tablica 2.1). Njihova specifičnost leži u njihovoj strukturi i načinu na koji iskorištavaju obrasce u slikama kako bi donijele kvalitetne rezultate. *LeNet* je jedna od prvih uspješnih arhitektura neuronskih mreža, razvijena 1998. za prepoznavanje rukom pisanih brojeva na čekovima, koristeći MNIST bazu (engl. *Modified National Institute of Standards and Technology database*). *AlexNet* arhitektura je znatno revolucionirala način obrade podataka s obzirom na to da je prva direktno postavljala konvolucijske slojeve jednog na drugoga, umjesto da između svakog dodaje sloj sažimanja kao što je to bio slučaj kod *LeNet*-a [4]. *AlexNet* osigurava da su vrijednosti podataka unutar svakog sloja približno jednake što pomaže u procesu učenja te koristi matematičku aktivacijsku funkciju ReLu (engl. *Rectified Linear Unit*) koja brže i stabilnije prenosi informacije kroz mrežu. Osim toga, *AlexNet* je uveo tehniku isključivanja (engl. *dropout*) koja sprječava neurone od prekomjernog učenja [6], odnosno ne dopušta da se previše specijaliziraju za određene obrasce što naposljetku pomaže u stvaranju robusnijeg modela koji bolje reagira na nove podatke. Par godina kasnije pojavio se *GoogLeNet*, s ciljem rješavanja problema dubine odnosno broja slojeva. Mreža se sastoji od dvadeset i dva sloja temeljenih na manjim konvolucijama pod imenom *Inception* [5] u kojima se paralelno koriste više filtera koji hvataju obrasce različitih veličina. *ResNet* mreža koristi takozvane rezidualne blokove. Osnovni princip rada blokova je uvođenje preskočnih veza (engl. *skip connections*) čija je uloga da u sljedeći blok prenose informacije preko slojeva bez gubitka ili distorzije signala, što znatno olakšava proces učenja modela [4].

Arhitektura	Godina razvoja	Broj slojeva	Broj parametara	Značajke i primjena
LeNet	1998.	7	~60,000	Konvolucijski slojevi, pooling slojevi, potpuno povezani slojevi, primjena: prepoznavanje rukom pisanih brojeva (MNIST)
AlexNet	2012.	8	~60 milijuna	Konvolucijski slojevi bez pooling sloja između, ReLu aktivacija, dropout slojevi, primjena: klasifikacija slika
GoogLeNet	2014.	22	~6.8 milijuna	Inception moduli, kombinacija različitih filtera, globalni average pooling, primjena: klasifikacija i detekcija objekta
ResNet	2015.	50+	~25 milijuna	Rezidualni blokovi, skip connections za sprječavanje distorzije signala, primjena: različita primjena u računalnom vidu

Tablica 2.1. Usporedba konvolucijskih neuronskih mreža

2.4. Dvofazni modeli

Iako su klasične CNN mreže učinkovite u klasifikaciji slika, nisu precizne u lokaliziranju objekta, pa samim time neprikladne za zadatke detekcije objekta. Glavni razlog tomu leži u fiksnoj arhitekturi klasične konvolucijske neuronske mreže. Naime, duljina izlaznog sloja je fiksna i određena brojem elemenata koje model treba prepoznati. To znači da će CNN imati isti broj izlaznih neurona bez obzira na to koliko objekata postoji na slici. Primjerice, ako se na slici nalaze tri životinje (npr. pas, mačka, ptica), CNN model će imati fiksni broj izlaznih neurona koji odgovara svakoj od tih životinja. No, ako se na novoj slici pojavljuju samo dvije životinje (npr. pas i mačka), CNN će i dalje imati tri izlaza neurona, što može otežati precizno prepoznavanje i lociranje objekta (Slika 2.6). Drugim riječima, broj informacija u izlaznom vektoru ne ovisi o broju objekta koji se pojavljuje na slici, već je unaprijed predodređen [7]. Upravo zbog toga se koriste modeli bazirani na pretraživanju regija poput modela R-CNN (engl. *Region-based Convolutional Neural Network*), Fast-CNN i Faster-CNN-a. Ovi modeli omogućuju detekciju različitih objekata na slici i precizno određivanje njihovih lokacija kroz dvije faze (problem klasifikacije i lokalizacije) čak i kada je broj objekata promjenjiv.

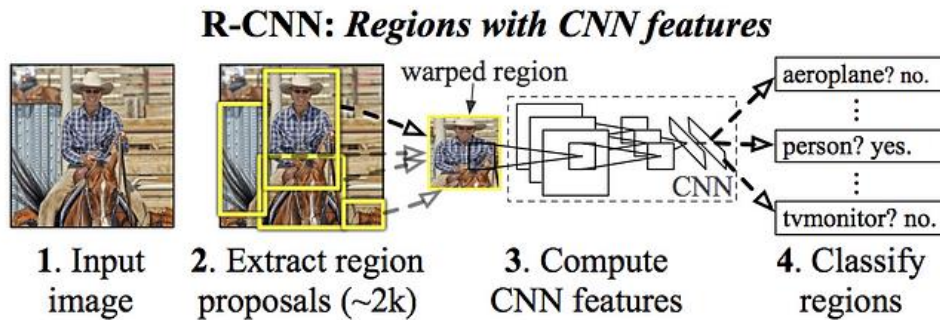


Slika 2.6. Razlika između klasifikacije i detekcije objekata pomoću modela baziranih na pretraživanju regija (<https://iq.opengenus.org/evaluation-metrics-for-object-detection-and-segmentation/>, 2024.)

2.5. R-CNN

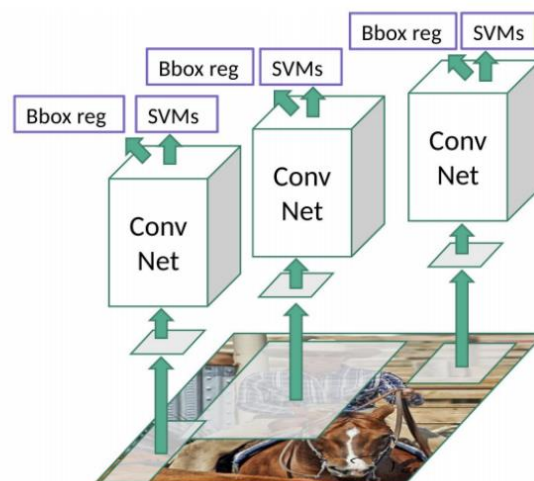
Glavni zadatak konvolucijske neuronske mreže bazirane na pretraživanju regija je točno identificirati i lokalizirati glavne objekte na slici putem okvira koji ih obuhvaćaju (engl. *bounding boxes*). Da bi se odredila lokacija granica tih okvira, slika mora proći kroz algoritam selektivnog pretraživanja. Radi se o procesu kojeg su predložili Ross Girshick i suradnici 2013.

godine pri čemu se analizira slika kroz 2000 regija različitih veličina, grupirajući piksele u okvire na temelju teksture, boje ili intenziteta kako bi se identificirali potencijalni objekti (Slika 2.7) [8].



Slika 2.7. Proces identifikacije i lokalizacije objekta pomoću R-CNN neuronske mreže
(Gandhi, 2018.)

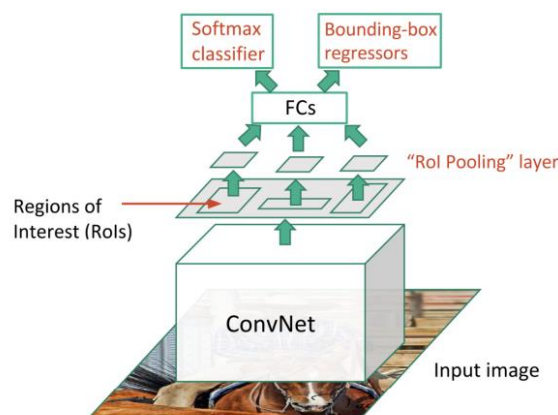
Selektivno pretraživanje odvija se u tri koraka: prvo se slika podijeli na manje regije odvojene okvirima, zatim se kombiniraju slične regije u veće okvire korištenjem selektivnog algoritma i naposljetku se koriste te nove regije za konačno definiranje prijedloga regija objekta (Slika 2.8) **Pogreška! Izvor reference nije pronađen..** Nakon što se stvore prijedlozi, R-CNN s tandardizira regije na kvadratnu veličinu i provodi ih kroz modificiranu verziju *AlexNet* kako bi se točno klasificirali objekti unutar regija.



Slika 2.8. Arhitektura R-CNN neuronske mreže (Gandhi, 2018.)

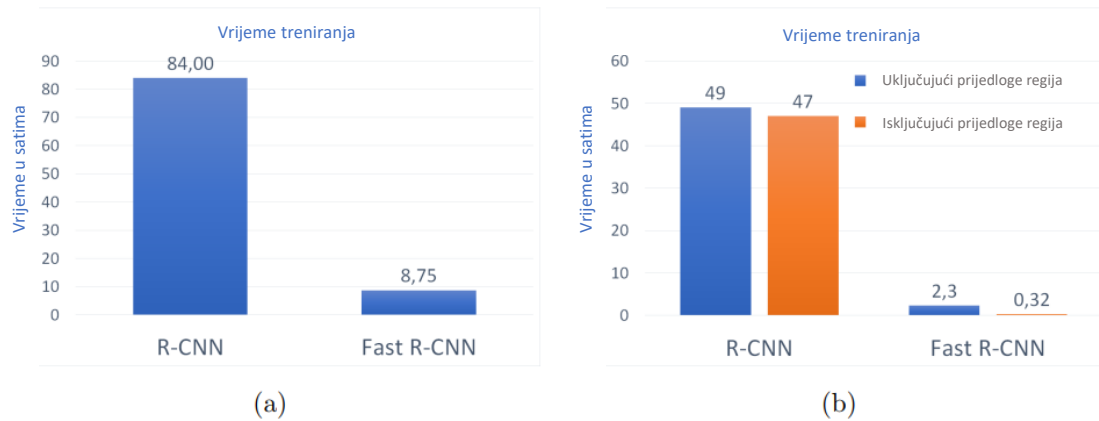
2.5.1. Brzi R-CNN

Kako bi ispravio nedostatke R-CNN mreže, Girshick je 2015. godine predložio model brze mreže R-CNN (engl. *Fast Region-based Convolutional Neural Network*) koji osigurava točnost i brzinu detekcije **Pogreška! Izvor reference nije pronađen..** Na početku se pomoću algoritma selektivnog pretraživanja analizira slika i generira prijedlog regija koje bi mogle sadržavati objekte od interesa (engl. *ROI – Region of Interest*). Zatim se svaki prijedlog regije kombinira s početnom slikom te se prolazi samo jednom kroz konvolucijsku mrežu gdje se kroz nekoliko konvolucijskih slojeva i slojeva sažimanja stvara mapa značajki koja sadrži informacije o različitim dijelovima slike. Nakon toga slijedi sloj poznat kao RoIPooling (engl. *Region of Interest Pooling*) pri čemu se izdvajaju vektori značajki fiksne duljine za svaki prijedlog regije (Slika 2.9). Ovaj korak omogućuje da svaki prijedlog regije bude reprezentiran vektorom točno definirane duljine, što olakšava daljnju obradu podataka [7]. Potom se vektori prosljeđuju kroz niz potpuno povezanih slojeva koji su odgovorni za klasifikaciju objekta te za regresiju okvira kako bi se poboljšala pozicija okvira. Konačno, brzi R-CNN generira dva izlazna sloja od kojih je jedan zadužen za procjenu vjerojatnosti klase objekta (pri čemu svaka klasa ima svoju vjerojatnost koja ukazuje na prisutnost određenog objekta na slici), dok drugi izlazni sloj daje četiri stvarne vrijednosti za svaku od klasa objekta. Te četiri vrijednosti predstavljaju preciznije pozicije granica okvira u kojima se nalazi objekt. Iako brzi R-CNN obrađuje sliku devet puta brže od prethodnog modela (Slika 2.10), on i dalje koristi algoritam selektivnog pretraživanja. Budući da je takav algoritam vrlo zahtjevan, poprilično je nepraktičan i neprecizan jer iziskuje previše vremena te se ne može prilagoditi svim zadacima detekcije objekta [10].



Slika 2.9. Arhitektura brzog R-CNN-a

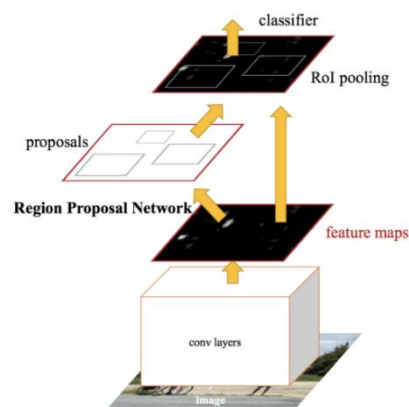
(<https://www.geeksforgeeks.org/fast-r-cnn-ml/>, 2024.)



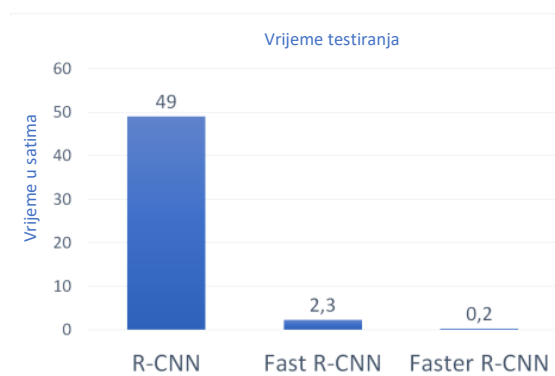
Slika 2.10. Usporedba vremena između R-CNN-a i Fast R-CNN-a. Vrijeme treniranja za R-CNN puno je duže nego za brzi R-CNN (a). Brzi R-CNN također je znatno brži u analizi slika (b). (Voithofer, 2019.)

2.5.2. Brži R-CNN

Takozvani brži R-CNN predstavlja daljnji napredak algoritma jer eliminira potrebu za selektivnim pretraživanjem regija. Zamjenjuje ga posebna konvolucijska mreža koja sama uči i generira prijedloge regija (engl. *Region Proposal Network, RPN*) [10]. Njoj se prosljeđuju prethodno generirane mape značajki cijele slike dobivenih iz konvolucijskog sloja. RPN mreža koristi posebne referentne okvire (engl. *anchor boxes*) kako bi predložila regije na slici. Okviri se sastoje od različitih veličina i omjera te služe kao referenca za potencijalne objekte. RPN koristi devet različitih referentnih okvira koji se postavljaju na svaku lokaciju na mapi značajki. Nakon toga se, pomoću već spomenutog RoIPooling sloja, klasificiraju odvojene regije u jedan od dva razreda: „objekt“ i „ne-objekt“. Zatim se koristi regresija pomaka kako bi se okviri bolje prilagodili na objekte. Radi se o tehnici koja se bazira na matematičkim operacijama te analizira razlike između početnog okvira i stvarne pozicije ili veličine objekta (Slika 2.11). Ovakav pristup znatno smanjuje vrijeme detekcije u stvarnom vremenu (Slika 2.12) [11].



Slika 2.11. Arhitektura bržeg R-CNN-a (Gandhi, 2018.)



Slika 2.12. Usporedba vremena testiranja između R-CNN-a, brzog R-CNN-a i bržeg R-CNN-a.

Brži R-CNN uspio je drastično smanjiti vrijeme u usporedbi s brzim R-CNN-om.

(Voithofer, 2019.)

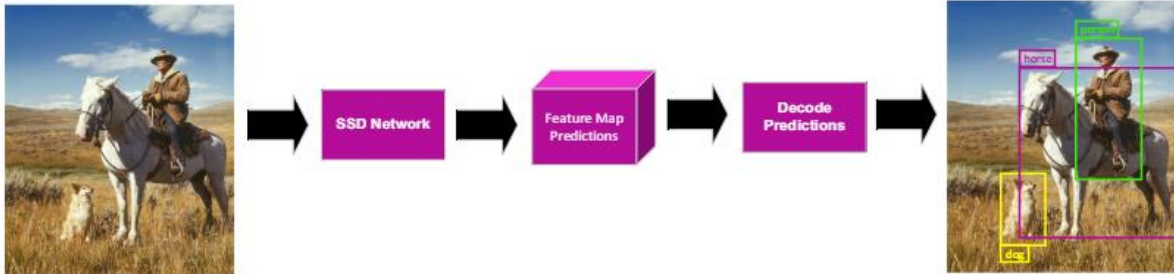
2.6. Jednofazni modeli

Modeli u dvije faze poput R-CNN i njegovih inačica su izrazito precizni, budući da prvo izvršavaju zadatak prepoznavanja lokacije objekta, a tek onda prelaze na klasifikaciju objekta. No, takva točnost dolazi uz relativno dugo vrijeme izvršavanja i velik broj parametara. Nasuprot njima, jednofazne modele karakterizira istovremeno izvršavanje zadataka lokalizacije i klasifikacije [12]. Zbog toga se upravo jednofazni modeli najčešće koriste za potrebe detekcije u stvarnom vremenu (npr. kod video nadzora ili kod autonomne vožnje). U nastavku su predstavljeni jednofazni SSD (engl. *Single Shot MultiBox Detector*) i YOLO (engl. *You Only Look Once*) modeli, s posebnim naglaskom na YOLO arhitekturu.

2.7. SSD detektori objekata

Prije razvoja jednofaznog modela pokušavalo se ubrzati detektiranje objekta mijenjanjem različitih parametara u pojedinim fazama klasifikacije i lokalizacije. Međutim, takve promjene obično su rezultirale smanjenjem točnosti detekcije. To je dovelo do stvaranja SSD modela [13] koji spaja dvije faze u jednu, bez gubitka preciznosti u detekciji. Osnovni princip rada SSD modela temelji se na takozvanom pristupu unaprijed (engl. *feed-forward*), što znači da podaci prolaze kroz konvolucijsku mrežu u jednom smjeru, od ulaza do izlaza, bez povratnih informacija. Zadatak modela je pronaći granične okvire zadanih dimenzija oko objekata na slici te odrediti kojoj klasi ti objekti pripadaju. Arhitektura SSD modela se sastoji od tri koraka (Slika 2.13). Na početku se pregledava slika kako bi se iz nje izvukle glavne značajke. Za to se koristi osnovna konvolucijska mreža (npr. *ResNet*) koja je prethodno trenirana na velikom skupu podataka poput *ImageNet* skupa. Zatim se dodaju dodatni slojevi koji pretražuju sliku i pronalaze objekte različitih veličina te generiraju granične okvire. Za svaki okvir, SSD provodi

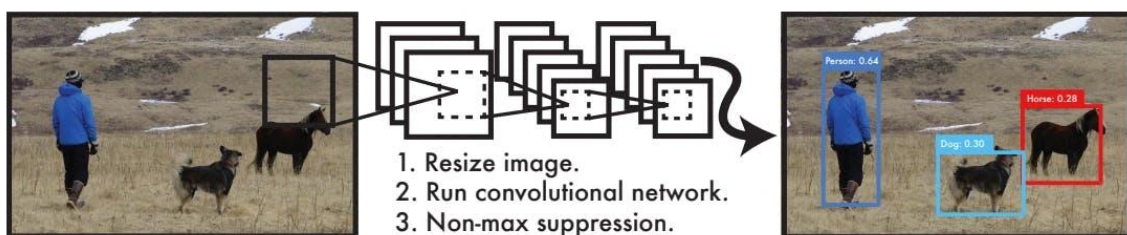
klasifikaciju i lokalizaciju objekta te se zajedno eliminiraju duplikati okvira. Upravo zato što koristi niz konvolucijskih slojeva za ispravno identificiranje objekta različitih veličina, SSD je izrazito točan u detekciji manjih objekata [14].



Slika 2.13. Arhitektura SSD modela (Coiras, 2020.)

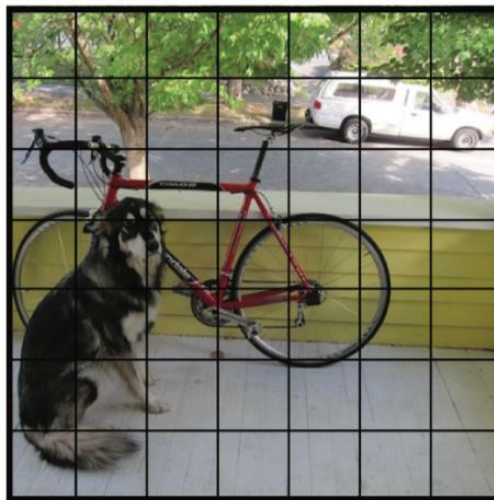
2.8.YOLO

YOLO model predstavljen je 2015. godine kao revolucionarni pristup dubokog učenja za detekciju objekata u slikama [15]. Prilikom testiranja ispostavilo se da može procesirati 45 slika u sekundi (koristeći *Nvidia Titan X GPU*) uz paralelnu, preciznu detekciju objekta. Sam naziv dolazi od engleske skraćenice za „*You Only Look Once*“ što odlično sumira brzinu i metodu koju ovaj model koristi. YOLO je značajan po tome što kompletno odbacuje prethodne principe detekcije u dvije faze te objedinjuje sve to u jednu, prolaznu konvolucijsku mrežu. Ona predviđa okvire objekata i njihovu lokaciju na temelju značajki iz slike što znači da se cijela slika analizira samo jednom, eliminirajući tako potrebu za složenim postupcima poput generiranja prijedloga regija ili višestrukog prolaza kroz konvolucijsku mrežu (Slika 2.14). Ključna ideja iza YOLO modela je transformirati problem detekcije objekta koji uključuje klasifikaciju i lokalizaciju u problem regresije (što znači da model izravno, na temelju podataka, predviđa kontinuirane vrijednosti kao što su koordinate okvira i vjerojatnosti klasa unutar slike) [16]. U nastavku slijedi detaljan opis arhitekture YOLO modela i matematičkih izraza koji se koriste kako bi se dobio bolji uvid u ovu vrstu tehnologije.



Slika 2.14. Osnovni princip rada YOLO modela (Elsheblly, 2020.)

Slika se na početku dijeli na mrežu s poljima veličina $S \times S$ (Slika 2.15). Ako se unutar određenog polja nađe centar nekog objekta, to polje je odgovorno za detekciju cijelog objekta (Slika 2.16). Za svako polje predviđa se broj (B) graničnih okvira kao i ocjene vjerojatnosti koje im pripadaju. Te ocjene vjerojatnosti (engl. *confidence scores*) su ključne jer odražavaju koliko je model siguran da se unutar okvira nalazi objekt [15]. Svaki od tih okvira može pridodati određenu vjerojatnost na temelju pet vrijednosti: visina (h) i širina (w) okvira u odnosu na cijelu sliku (vrijednosti između 0 i 1) (Slika 2.17), X-pozicija (x) i Y-pozicija (y) u odnosu na granice polja (vrijednosti između 0 i 1) (Slika 2.18).



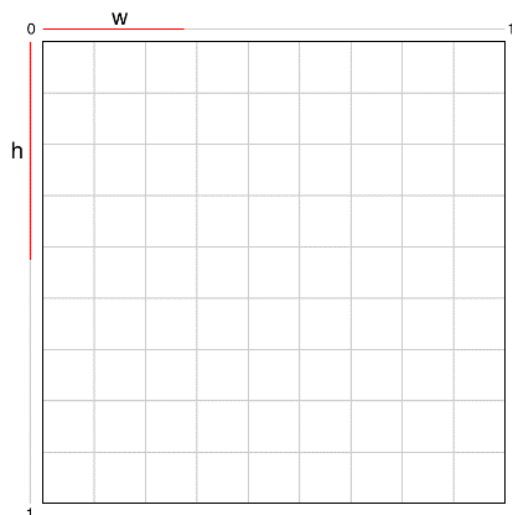
Slika 2.15. Primjer postavljanja mreže

(<https://kikaben.com/yolo-version-1/#yolo-architecture>, 2024.)

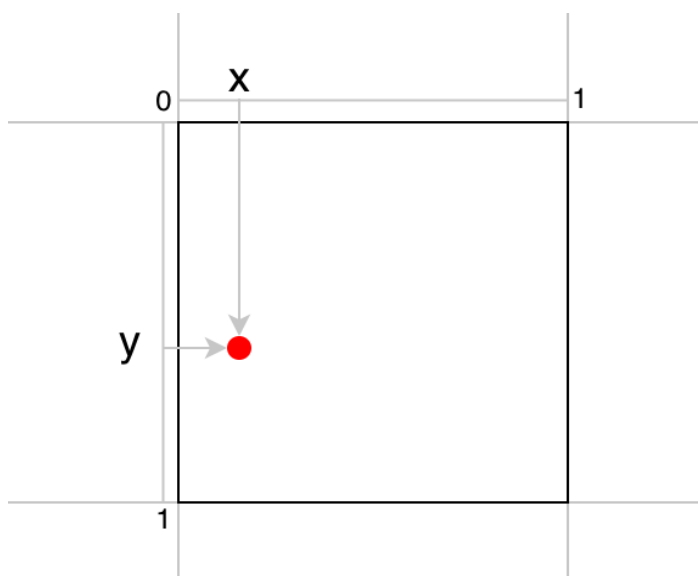


Slika 2.16. Polje koje sadrži centar objekta

(<https://kikaben.com/yolo-version-1/#yolo-architecture>, 2024.)

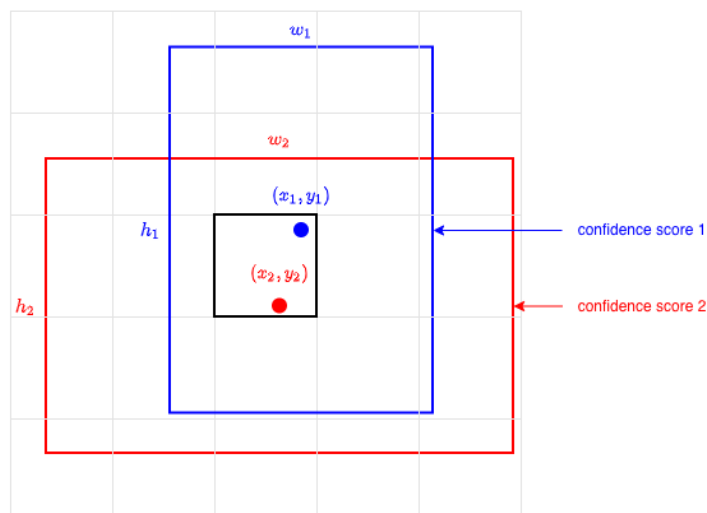


Slika 2.17. Visina i širina u odnosu na cijelu sliku
 (<https://kikaben.com/yolo-version-1/#yolo-architecture>, 2024.)

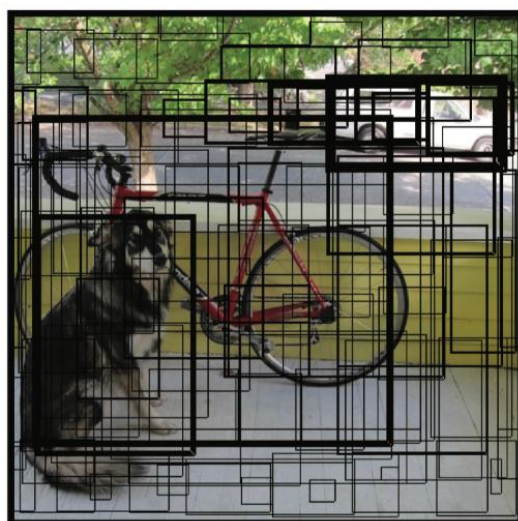


Slika 2.18. X i Y koordinate u odnosu na pojedinačno polje
 (<https://kikaben.com/yolo-version-1/#yolo-architecture>, 2024.)

Konfiguracijom se može odrediti koliko će se graničnih okvira generirati po polju jedne mreže. Primjera radi koriste se dva okvira po polju od kojih svaki daje svoju ocjenu vjerojatnosti (Slika 2.19). Kada se tako proanalizira cijela mreža koja sadrži npr. 49 polja (7x7), dobivaju se 98 graničnih okvira što je znatno manje od 2000 prijedloga regija koje su se prethodno koristile u brzom R-CNN modelu (Slika 2.20).

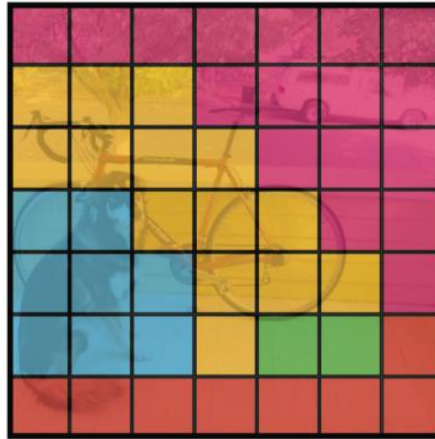


Slika 2.19. Primjer dva okvira po jednom polju
 (<https://kikaben.com/yolo-version-1/#yolo-architecture>, 2024.)

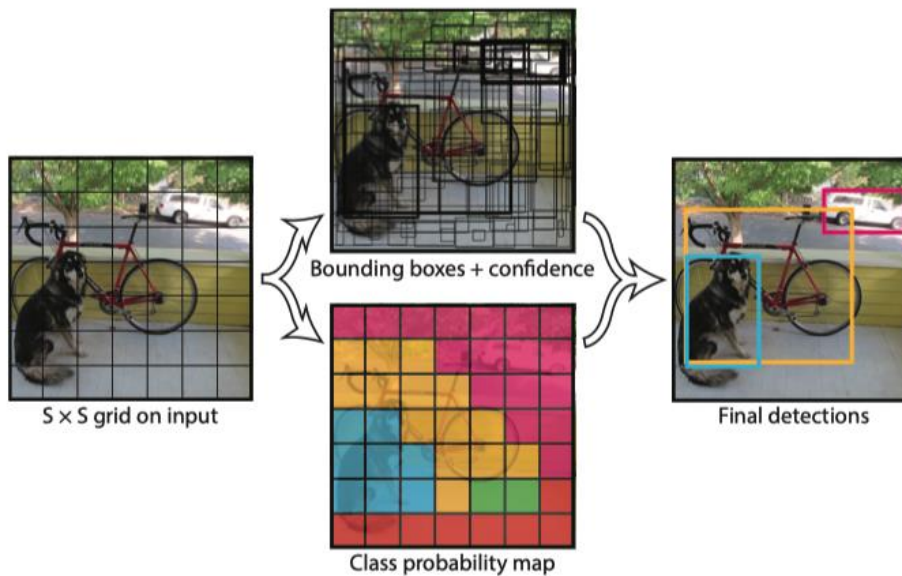


Slika 2.20. Primjer slike koja sadrži sve okvire i ocjene vjerojatnosti.
 Deblji okviri označavaju veću vjerojatnost da se unutar tih okvira nalazi objekt.
 (<https://kikaben.com/yolo-version-1/#yolo-architecture>, 2024.)

Svako polje također predviđa skup vjerojatnosti uvjetnih klasa (engl. *conditional class probabilities*) za vrstu objekta koji bi se mogao nalaziti unutar njega $Pr(Class_i|Object)$. Uvjet je da se u polju nalazi centar objekta, odnosno da je ocjena rezultat pouzdanosti bliži 1. Neovisno o broju graničnih okvira, po polju se generira samo jedan skup vjerojatnosti uvjetnih klasa. U suštini, vjerojatnosti uvjetnih klasa u YOLO sustavu odražavaju predviđanje koje klase objekata najvjerojatnije postoje unutar određenog polja mreže (Slika 2.21), uzimajući u obzir i prisutnost objekta u tom polju. Te se vjerojatnosti tijekom testiranja množe s već prethodno spomenutim ocjenama vjerojatnosti kako bi se dobila pouzdanost po klasama, formula 2.1 [17].



Slika 2.21. Primjer predikcije klasa. Svaka boja polja označava najvjerojatniju klasu za to polje. (<https://kikaben.com/yolo-version-1/#yolo-architecture>, 2024.)



Slika 2.22. Grafički prikaz YOLO modela (<https://kikaben.com/yolo-version-1/#yolo-architecture>, 2024.)

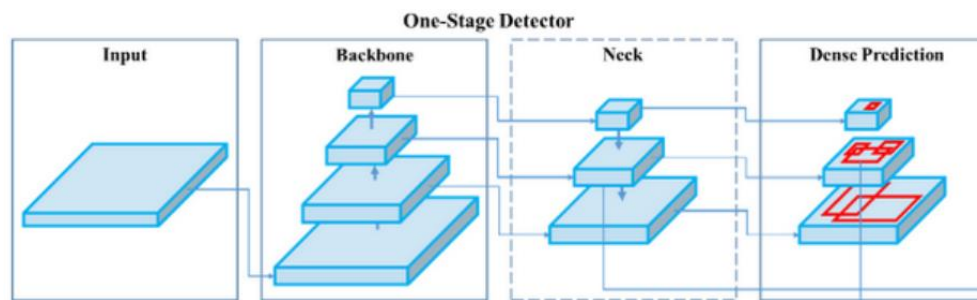
$$\text{Pr(Object)} * \text{IOU} * \text{Pr(Class}_i | \text{Object)} = \text{Pr(Class}_i) * \text{IOU} \quad (2.1)$$

Nakon obrade ulazne slike dobiva se izlazni tenzor koji se može zapisati u obliku formule 2.2:

$$S \times S \times (B \times 5 + C) \quad (2.2)$$

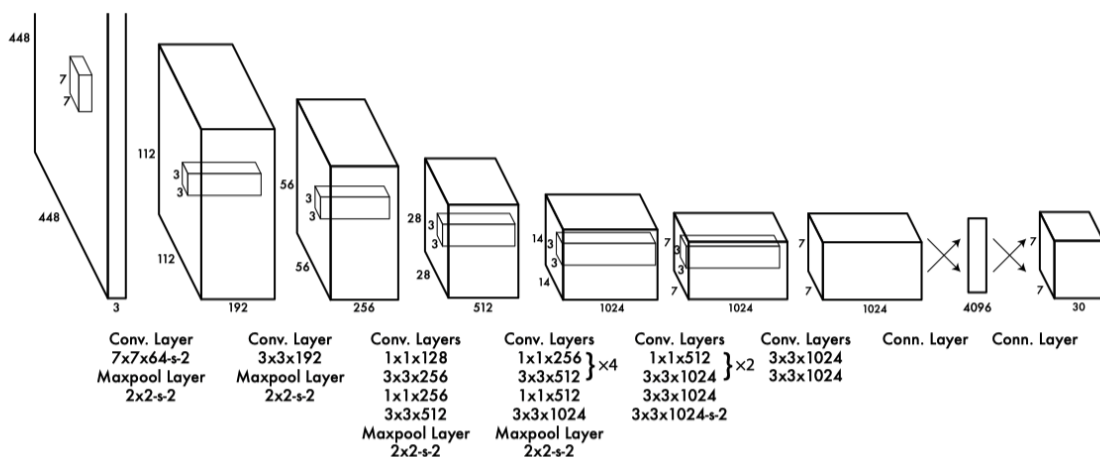
pri čemu je $S \times S$ broj polja, B broj graničnih okvira kojeg predviđa svako polje, broj 5 označava sve parametre koji se promatraju (x,y koordinate, visina, širina i parametar vjerojatnosti) te C koji predstavlja broj klasa po polju. [19] Osnovna arhitektura YOLO modela sastoji se od tri dijela kroz koja moraju proći svi ulazni podaci. To su redom: leđa (engl. *backbone*), vrat (engl.

neck) i glava (engl. *head*) (Slika 2.23). Leđa se sastoje od duboke konvolucijske neuronske mreže koja računa značajke poput rubova, oblika i tekstura iz ulaznih podataka i na temelju toga stvara mapu značajki s najbitnijim informacijama. Obično se radi o mreži prethodno treniranoj na skupu slika za klasifikaciju (npr. *ImageNet* ili *DarkNet*). Nadalje, vrat sa svojim konvolucijskim slojevima prikuplja te značajke te pokušava poboljšati semantičko značenje, odnosno razumijevanje konteksta slike i objekta u njoj što će kasnije znatno pomoći u kvaliteti i brzini detekcije. U završnom dijelu glave računa se funkcija gubitka (engl. *loss function*) temeljena na razlici između predviđenih i stvarnih vrijednosti tijekom treninga mreže. Također, ovaj dio uključuje i postupak potiskivanja preklapajućih graničnih okvira (engl. *Non Maximum Suppression, NMS*) pomoću kojeg se prepoznaju i uklanjaju svi suvišni granični okviri ostavljajući samo jedan glavni granični okvir po objektu [18].



Slika 2.23. Osnovna arhitektura YOLO modela (https://www.researchgate.net/figure/Schematic-of-a-typical-one-stage-detector-9_fig1_362203223, 2024.)

Prva verzija YOLO modela inspirirana je *GoogLeNet* neuronskom mrežom te se njegova struktura sastoji od 24 konvolucijskih sloja koji izvlače značajke iz slika te dva potpuno povezana sloja koja ih obrađuju. Unutar tih 24 sloja nalaze se i četiri sloja sažimanja maksimalnih vrijednosti koji reduciraju dimenzije značajki te se fokusiraju na najvažnije informacije (Slika 2.24). Za razliku od *GoogLeNet*-a koji za to koristi složene *Inceptions* slojeve, YOLO koristi nešto jednostavniju strukturu redukcijских slojeva pomoću filtera dimenzija 1x1 čiji je glavni zadatak smanjiti prostor značajki dobiven iz prethodnih slojeva. Konvolucijski slojevi prethodno uče razne uzorke i značajke slika prema *ImageNet* skupu podataka s 1000 klasa za prepoznavanje prepolovljenih rezolucija (224x224) te se kasnije, u fazi same detekcije objekta, rezolucija slike udvostručuje (448x448) kako bi preciznost detekcije bila bolja. Zadnji sloj predviđa koordinate okvira i vjerojatnosti klasa [17].



Slika 2.24. Grafički prikaz strukture arhitekture YOLO modela (Redmon,2016.)

Važno je razumjeti funkciju gubitka koja se koristi u procesu treniranja YOLO modela kako bi se pratila uspješnost predikcije mreže. Funkcija gubitka može se podijeliti na gubitak lokalizacije, gubitak pouzdanosti i gubitak klasifikacije [18].

Gubitak lokalizacije pomaže mjeriti koliko su točni modeli u određivanju pozicije objekata na slici. On mjeri odstupanja nastala između koordinata (x i y) i veličina (visina h i širina w) predviđenih graničnih okvira i stvarnih okvira objekata. Ovisno o tome koliko se predikcije razlikuju od stvarnih podataka, pripisuju im se pogreške te se penaliziraju samo oni okviri koji su odgovorni za objekt (odnosno samo okviri s najvećim IoU čija polja prekrivaju i stvarne okvire objekta). Gubitak lokalizacije računa se prema formuli (2.3):

$$\begin{aligned}
 \lambda_{coord} \sum_{i=1}^{S^2} \sum_{j=1}^B \mathbb{I}_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
 + \lambda_{coord} \sum_{i=1}^{S^2} \sum_{j=1}^B \mathbb{I}_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]
 \end{aligned} \tag{2.3}$$

Pri čemu je $\mathbb{I}_{ij}^{obj} = 1$, ako je j -ti granični okvir u polju i odgovoran za detekciju objekta, u protivnom vrijedi 0. Prema tome pogreška lokalizacije se daje samo onim okvirima koji su odgovorni za objekt. Simbol $\hat{}$ iznad varijable označava da je ta varijabla samo predikcija (npr. \hat{x}_i), dok ona bez simbola označava da se radi o stvarnoj varijabli (npr. x_i). λ_{coord} postavljena je na 5 i povećava utjecaj gubitka u koordinatama graničnih okvira. Suma kvadratnih pogrešaka jednako vrednuje pogreške i u velikim i u malim okvirima. No, jasno je da su pogreške u manjim

okvirima znatno bitnije za ispravnu detekciju. Kako bi se taj problem riješio, umjesto procjene stvarnih vrijednosti širine i visine, predviđa se kvadratni korijen širine i visine graničnih okvira.

Gubitak pouzdanosti računa kolika je greška u procjeni postojanja objekta unutar graničnog okvira i koliko je pogrešna pouzdanost u njegovu odgovornost za taj objekt. Međutim u ovom gubitku se ne računaju sve sume kvadratnih pogrešaka, već se gubitak dijeli u dva slučaja – u kojem je prisutan objekt i u kojem nije. Budući da većina okvira ne sadrži objekte, model se trenira da češće detektira pozadinu nego objekte. Stoga se, kao i u prethodnom gubitku, koristi λ_{noobj} faktor koji ograničava gubitke iz detekcije pozadine. Gubitak pouzdanosti definiran je prema formuli (2.4):

$$\sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{noobj} (C_i - \hat{C}_i)^2 \quad (2.4)$$

Pri čemu je

$\mathbb{I}_{ij}^{obj} = \begin{cases} 1, & \text{ako je granični okvir } j \text{ u polju } i \text{ odgovoran za prepoznavanje objekta} \\ 0, & \text{u protivnom vrijedi} \end{cases}$

$\mathbb{I}_{ij}^{noobj} = \begin{cases} 1, & \text{ako granični okvir } j \text{ u polju } i \text{ nije odgovoran za prepoznavanje objekta} \\ 0, & \text{u protivnom vrijedi} \end{cases}$

$\lambda_{noobj} = \text{postavljen na } 0,5$

$\hat{C}_i = \text{ocjena pouzdanosti graničnog okvira } j \text{ u polju } i$

Za svako polje i koje sadrži objekt, mora se izračunati i gubitak klasifikacije. On uzima u obzir koliko dobro model klasificira pronađeni objekt, prema tome mjeri koliko su predviđene vjerojatnosti uvjetnih klasa u tom polju pogrešne (2.5) [18].

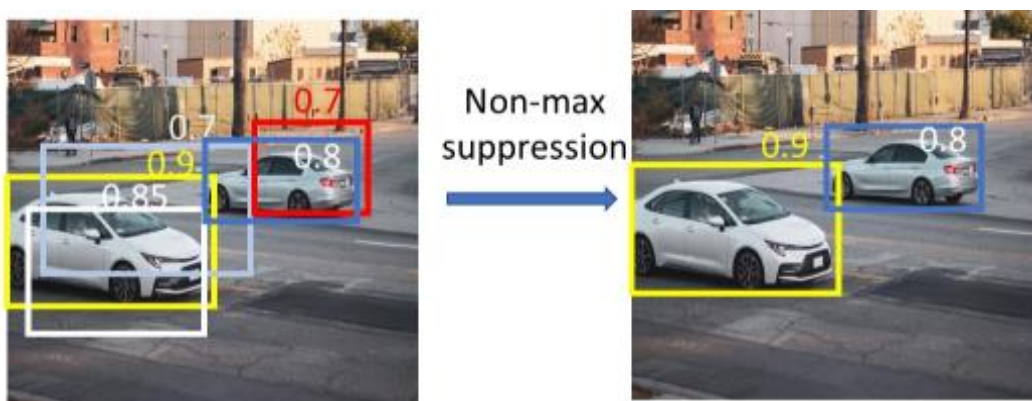
$$\sum_{i=1}^{S^2} \mathbb{I}_i^{obj} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2 \quad (2.5)$$

Pri čemu je $\mathbb{I}_{ij}^{obj} = 1$ ako se objekt pojavi u polju i . $\hat{p}_i(c)$ predstavlja predviđenu vjerojatnost da se u polju i nalazi uvjetna klasa c . Ako objekt u polju i uistinu pripada klasi c , tada je vjerojatnost $p_i(c)$ za tu klasu 1,0, u a za sve ostale klase 0,0.

Ukupan gubitak dobiva se zbrajanjem tih triju gubitaka (2.6) [17].

$$\begin{aligned}
 Loss = & \lambda_{coord} \sum_{i=1}^{S^2} \sum_{j=1}^B \mathbb{I}_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
 & + \lambda_{coord} \sum_{i=1}^{S^2} \sum_{j=1}^B \mathbb{I}_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
 & + \sum_{i=1}^{S^2} \sum_{j=1}^B \mathbb{I}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=1}^{S^2} \sum_{j=1}^B \mathbb{I}_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\
 & + \sum_{i=1}^{S^2} \mathbb{I}_i^{obj} \sum_{c \in classes} \left(p_i(c) - \hat{p}_i(c) \right)^2
 \end{aligned} \tag{2.6}$$

Nakon što model detektira objekte na slici, koristi se postupak potiskivanja preklapajućih graničnih okvira kako bi se uklonili suvišni okviri koji predviđaju isti objekt. Zadržavaju se granični okviri s najvećim indeksom preklapanja sa stvarnim okvirom (IoU). [19] U navedenom primjeru (Slika 2.25) pet graničnih okvira ima ocjene pouzdanosti iznad definiranog praga (npr. 0,6). Tri okvira pripadaju jednom, druga dva drugom automobilu. Nakon primjene NMS algoritma, ostaju samo dva najbolja granična okvira, jedan za svaki automobil [18].



Slika 2.25. Primjer provođenja potiskivanja preklapajućih graničnih okvira (Kuang, 2024.)

3. Metoda detekcije logotipa

U prethodnom poglavlju predstavljeni su različiti modeli za detekciju objekata. Iako su svi spomenuti modeli primjenjivi za prepoznavanje logotipa na stvarnim fotografijama, važno je istaknuti da je detekcija logotipa nešto kompleksniji proces, s obzirom na to da se logotipi često koriste u više različitih varijacija. Na primjer, jedan se brend može predstavljati publici kao samostalni grafički simbol, kao samostalni tekst, ili kao kombinacija tih elemenata (Slika 3.1). Također, u modernije vrijeme sve su češći apstraktni, minimalistički dizajni logotipa.

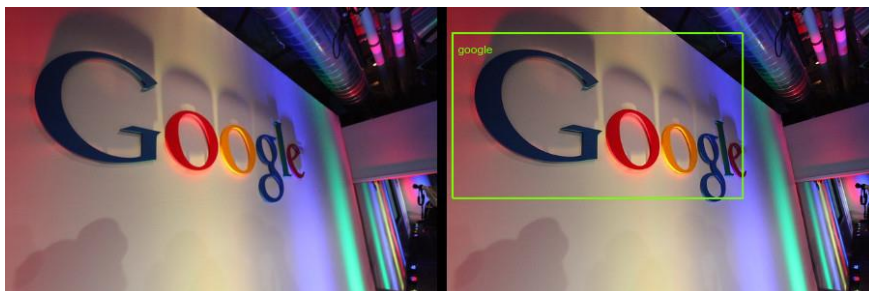


*Slika 3.1. Primjer varijacija logotipa brenda Kodak
(<https://www.enganagraphics.com.au/blog/logo-design-styles>, 2024.)*

Logotipi se na stvarnim fotografijama pojavljuju u raznovrsnim kontekstima i okolnostima. Oni mogu biti različitih veličina, rotacija, deformacija, s promjenama boje ili osvjetljenja, sa šumom i bez šuma, kao i pod utjecajem mnogih drugih faktora. Primjerice, prikaz logotipa brenda na majici koju netko nosi može biti u određenoj mjeri deformiran (Slika 3.2) [19], što uvelike otežava modelu ispravnu detekciju logotipa.



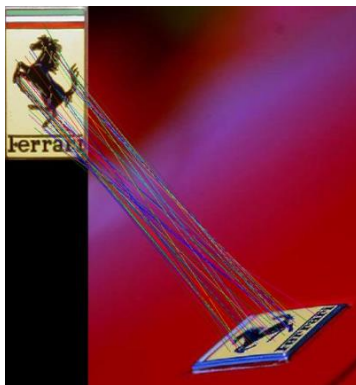
*Slika 3.2. Deformacija logotipa brenda Adidas na stvarnoj fotografiji
(<https://www.pinterest.com.au/pin/211458144981723323/>, 2024.)*



Slika 3.3. Deformacija Google logotipa

(<https://cloud.google.com/vision/docs/detecting-logos?hl=de>, 2024.)

Iz ovoga se može zaključiti da je detekcija logotipa izrazito izazovan zadatak. Slično kao i kod detekcije objekata, potrebno je logotip točno klasificirati i lokalizirati. Prvi istraživački radovi koji su se bavili detekcijom logotipa datiraju iz 1993. godine te su bazirani na ručno definiranim značajkama (vizualne značajke logotipa poput oblika, boje, teksture, simetrije i sl.) i mjerama koje su se koristile za uspoređivanje tih značajki (mjere poput korelacije, kosinusne sličnosti, euklidske udaljenosti i dr.) Kasnijih godina razvio se SIFT algoritam (engl. *Scale Invariant Feature Transform*) koji se temelji na homografiji, metodi koja identificira i transformira ključne točke logotipa tako da se one naknadno mogu usporediti sa značajkama logotipa spremljenog u bazi podataka (Slika 3.4). Iako precizna, glavna mana ove metode je u njezinom sporom procesiranju slike jer je potrebno izračunati nekoliko operacija prije prelaska na uspoređivanje logotipa [14].



Slika 3.4. Ilustracija homografije (Coiras, 2020.)

Zbog svoje visoke točnosti i brzine rada u detekciji objekata, mnogi su se istraživački radovi u posljednjem desetljeću usmjerili prema korištenju konvolucijskih neuronskih mreža poput brzog R-CNN-a i YOLO modela. Najveći izazov s kojim se suočavaju ovi modeli jest nedostatak velike baze podataka, odnosno nedovoljan izvor uzoraka logotipa za stvaranje dobro istreniranog i testiranog modela kojeg bi se smatralo pouzdanim. U nastavku su predstavljene najkorištenije baze za učenje za detekciju logotipa kao i mjere uspješnosti koje se koriste za evaluaciju modela.

3.1. Baze za učenje

Baze za učenje (engl. *datasets*) bitan su čimbenik za uspješan razvoj modela za detekciju logotipa. Radi se o skupu podataka koji pruža modelu pristup stvarnim slikama iz kojih on uči prepoznavati i razlikovati logotipe u različitim okolnostima. Također, dobre baze omogućuju kvalitativne i kvantitativne usporedbe različitih algoritama koji se koriste u detekciji [20]. U nastavku su predstavljene najpoznatije baze stvorene specifično za logo detekciju te njihove osnovne karakteristike. Iako se mogućnosti detekcije logotipa proučavaju već puna dva desetljeća, većina istraživanja ima ograničen pristup bazama, odnosno koriste baze manjih reprezentativnih uzoraka. Razlog tome je učestalo privatno vlasništvo nad većim bazama. Prema tome, baze za učenje mogu se podijeliti na javne i privatne, nadalje količinski na manje i veće baze, ovisno o broju slika i podataka koje sadrže (Tablica 3.1) [19].

Baza za učenje	Broj slika	Broj klasificiranih logotipa	Okviri	Javno dostupno
FlickrLogos-27	810	27	3 711	Da
BelgaLogos	10 000	37	2 695	Da
QMUL-OpenLogo	27 083	309	51 207	Da
LOGO-Net	73 414	160	130 608	Ne
WebLogo-2M	2 190 757	194	0	Da * pod uvjetom

Tablica 3.1. Tablica baza za učenje logotipa

FlickrLogos-27 je jedna od najpopularnijih javnih baza za detekciju logotipa. Sadrži 27 različitih klasa s 30 slika u svakoj klasi što ju čini manjom bazom sa ukupno 810 slika. Radi se o slikama iz stvarnog svijeta, od kojih mnoge sadrže logotipe promijenjene u izgledu i osvijetljenu. BelgaLogos baza sastoji se od ukupno 10 000 stvarnih slika na kojima se pojavljuje 37 različito klasificiranih logotipa. Graničnim okvirima je označeno 2 695 logotipa, pri čemu logotipi imaju maksimalnu visinu i širinu od 800 piksela [19]. QMUL-OpenLogo je stvoren na temelju podataka sedam postojećih baza te koristi otvoreni protokol za dosljedno testiranje i uspoređivanje učinkovitosti algoritama za detekciju logotipa. Iako se radi o manjoj bazi koja

sadrži 27,083 slika unutar kojih je klasificirano 309 logotipa [14], njegova kvaliteta je znatno bolja u usporedbi s prijašnjim bazama. Naime, klasificirani logotipi se pojavljuju u različitim veličinama i u znatno više primjeraka. To omogućuje bolje testiranje algoritma u uvjetima koji su slični stvarnim situacijama gdje su logotipi često zastupljeni u različitim deformacijama i veličinama [20]. Logo-Net je trenutno jedna od najopsežnijih baza s ukupno 73 414 slika na kojima je graničnim okvirom označeno preko 130 608 logotipa u 160 različitim klasa. S obzirom na to da je svaka slika ručno anotirana, konstruiranje ovako velike baze je vremenski vrlo zahtjevno i skupo te se stoga radi o privatnoj bazi. Zbog svoje veličine i izrazito visoke kvalitete anotacije, Logo-Net je idealan za istraživanje i razvoj naprednih tehnika strojnog učenja [19]. WebLogo-2M baza je jedna od najvećih baza podataka za detekciju logotipa, s preko 2 milijuna slika koje uključuju više od 194 različitih brendova. Dobivena automatskim prikupljanjem i obradom podataka s Interneta, isključuje slike male širine i/ili visine te duplicirane slike. No, upravo zbog opsežnosti i automatskog procesa prikupljanja podataka, baza sadrži mnoge slike koje ne prikazuju ispravne logotipe ili imaju pogrešne oznake. Također, često postoji neujednačen broj slika za različite logotipe, što znači da su neki logotipi mnogo zastupljeniji od drugih. WebLogo-2M baza je javno dostupna, no često uz ograničenja koja su definirana od strane autora [20].

3.2. Mjere za uspješnost detekcije

Nakon treniranja modela pomoću baza, ključno je utvrditi njegovu učinkovitost u rješavanju problema detekcije. U prethodnim poglavljima je spomenuto da postoje dva osnovna parametra koja se moraju zadovoljiti: (1) klasifikacija, odnosno ispravno prepoznavanje logotipa, i (2) lokalizacija logotipa na slici. Precizna evaluacija performansi modela omogućuje nam utvrditi je li model dovoljno dobar za praktičnu primjenu. Osim toga, evaluacija pomaže u prepoznavanju slabosti modela u određenim uvjetima te pruža potrebne informacije za usporedbu različitih algoritama i pristupa. Kako bi se postigli empirijski točni rezultati, s vremenom su se uspostavile različite mjere koje objektivno mjere uspješnost modela. Na samom početku valja spomenuti pojam binarne klasifikacije, sustava koji rastavlja problem detekcije na samo dvije mogućnosti: (1) logotip je prisutan ili (2) logotip ne postoji na slici. Ovo pojednostavljeno koristi se za izgradnju matrice zabune (engl. *confusion matrix*) (Tablica 3.2), koja je temelj za izračunavanje ključnih evaluacijskih mjera.

		Stvarnost	
		U stvarnosti pozitivno	U stvarnosti negativno
Pretpostavka	Testirano pozitivno	Istinski pozitivno (TP, <i>true positive</i>) Stvarnost: logotip je prisutan Pretpostavka: logotip je prisutan	Lažno pozitivno (FP, <i>false positive</i>) Stvarnost: logotip ne postoji Pretpostavka: logotip je prisutan
	Testirano negativno	Lažno negativno (FN, <i>false negative</i>) Stvarnost: logotip je prisutan Pretpostavka: logotip ne postoji	Istinski negativno (TN, <i>true negative</i>) Stvarnost: logotip ne postoji Pretpostavka: logotip ne postoji

Tablica 3.2. Prikaz matrice zabune

Matrica zabune sastoji se od četiri osnovne vrijednosti: istinski pozitivna (TP), istinski negativna (TN), lažno negativna (FN) i lažno pozitivna (FP). U kontekstu detekcije logotipa, istinski pozitivna (TP) vrijednost označava da je na slici stvarno prisutan logotip i da je uspješno detektiran, dok istinski negativna (TN) vrijednost pokazuje koliko puta je model ispravno prepoznao da logotipa nema na slici. Nadalje, lažno negativna (FN) označava propust modela u detekciji logotipa koji je uistinu prisutan na slici, dok lažno pozitivna (FP) vrijednost pokazuje koliko je puta model pogrešno označio prazno područje slike kao logotip [21]. S ovim podacima se zatim mogu izračunati mjere za evaluaciju poput osjetljivosti i preciznosti.

Osjetljivost (engl. *Recall*) ili odziv mjeri sposobnost modela da ispravno prepozna istinski pozitivnu vrijednost (TP), u ovom kontekstu da detektira sve slike na kojima je stvarno prisutan logotip. Računa se kao omjer broja istinski pozitivne vrijednosti (TP) i ukupnog zbroja svih pozitivnih vrijednosti, odnosno zbroj svih slika koje stvarno sadrže logotipove (TP + FN). Ukoliko je postotak visoki, znači da model uspješno odrađuje detekciju. Varijabla (t) označava definirani prag odluke (engl. *decision threshold*) uz pomoć kojeg model odlučuje da li na određenoj slici prepoznaje logotip ili ne. Jednadžba za osjetljivost 3.1. glasi:

$$Recall(t) = \frac{TP(t)}{TP(t) + FN(t)} \quad (3.1)$$

Mjera preciznosti (engl. *Precision*) označava udio ispravno detektiranih logotipa u odnosu na ukupan broj svih detekcija koje su provedene. Dakle, ova mjera naglašava koliko je model sposoban razlikovati istinski pozitivne od istinski negativnih vrijednosti, odnosno koliko su točne detekcije kada model tvrdi da slika sadrži logotip [22]. Ako je postotak preciznosti visok,

to znači da većina slika koje je model svrstao u slike s logotipom stvarno i sadrže taj logotip. Jednadžba 3.2. stoga glasi:

$$Precision(t) = \frac{TP(t)}{TP(t) + FP(t)} \quad (3.2)$$

U idealnim uvjetima, obje vrijednosti su visoke, što bi značilo da je točno detektiran objekt jedne klase. Ovisno o specifičnoj primjeni detekcije logotipa, potrebno je uspostaviti odgovarajući prag odluke i ravnotežu između preciznosti i osjetljivosti kako bi se postigli željeni, točni rezultati. Kada model ima postavljen viši prag odluke, on je sigurniji u svoje detekcije, što povećava njegovu preciznost jer će manje griješiti u prepoznavanju objekta (u ovom slučaju logotipa). Primjerice, ako se postavi prag odluke na 0,7, model će oprezno označavati samo one oblike za koje je gotovo siguran da su logotipi. Međutim, zbog ovisnosti preciznosti i osjetljivosti, to također znači da model može propustiti prepoznati i neke stvarne logotipe, što mu smanjuje osjetljivost. S druge strane, niži prag odluke omogućuje modelu prepoznavanje više objekata zbog visoke osjetljivosti. Na primjer, postavljanjem praga na 0,3, model će gotovo svaki sumnjivi oblik označiti kao logotip. Dakle, model će na ovaj način otkriti većinu stvarnih logotipa, ali će također imati i mnogo pogrešnih detekcija, poput označavanja slučajnih objekata ili sjena kao logotip, čime se smanjuje njegova preciznost [23].

Kao primjer, modelu postavimo zadatak prepoznavanja Apple logotipa na uzorku od 100 slika. Od tih 100 slika, u stvarnosti samo 20 sadrži Appleov logotip. Ukoliko model radi s postavljenim visokim pragom odluke priznat će da je na slici logotip samo kada je potpuno siguran u to. Za primjer uzmimo da će prepoznati 15 slika, a pogrešno označiti 2, što znači da mu je ostalo 5 slika na kojima uopće nije detektirao logotip.

$t=0,7$

$$Precision(t) = \frac{TP(t)}{TP(t) + FP(t)} = \frac{15}{15+2} = \frac{15}{17} \approx 0,88 \text{ (88\%)},$$

$$Recall(t) = \frac{TP(t)}{TP(t) + FN(t)} = \frac{15}{15 + 5} = \frac{15}{20} = 0,75 \text{ (75\%)}$$

Izračunom se vidi da je preciznost visoka (88%), ali zbog niže osjetljivosti (75%) propustit će prepoznati logotipe na nekim slikama. Ukoliko je prag odluke niži, model će bilo kakav objekt u slici koji bi mogao biti Appleov logotip automatski svrstati pod tu klasifikaciju, čak i uz veći

rizik pogreške. Uzmimo da je prepoznato 18 slika na kojima je stvarno prisutan logotip, no zbog veće osjetljivosti pogrešno je svrstano 10 slika te nisu klasificirane 2 slike.

$t=0,3$

$$Precision(t) = \frac{TP(t)}{TP(t)+FP(t)} = \frac{18}{18+10} = \frac{18}{28} \approx 0,64 \text{ (64\%)},$$

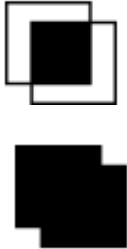
$$Recall(t) = \frac{TP(t)}{TP(t) + FN(t)} = \frac{18}{18 + 2} = \frac{18}{20} = 0,90 \text{ (90\%)}$$

Dakle, s visokom osjetljivošću (90%) model može prepoznati većinu logotipa, ali mu je zato preciznost niža jer je čak 36% prepoznatih logotipa u ovom primjeru potpuno krivo klasificirano. Visoki prag povjerenja je koristan kada je važno da detekcija bude vrlo točna (npr. u detekciji logotipa za pravne ili u marketinške svrhe). S druge pak strane, niski prag povjerenja je učinkovit ako je važno prepoznati što više logotipa, pa čak i uz rizik pogreške (npr. u analizi prisutnosti brendova). Prema tome, važno je odrediti ravnotežu između preciznosti i osjetljivosti. U tu svrhu, stvorena je statistička vrijednost, takozvana F1 mjera (Formula 3.3) koja objedinjuje ove dvije mjere. U izračunu F1 mjere koristi se harmonijska sredina koja, za razliku od aritmetičke sredine, značajnije naglašava ekstremne vrijednosti. Ukoliko je jedna od mjera vrlo niska, F1 mjera će također biti niska. To znači da pragovi preciznosti i osjetljivosti moraju biti zadovoljeni kako bi i vrijednost F1 bila visoka [21].

$$F1 = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (3.3)$$

3.3. Mjere usporedbe kod detekcije

Presjek nad unijom IoU (engl. *Intersection over Union*), poznatiji i kao *Jaccardov indeks*, jest vrijednost koja se koristi za procjenu točnosti lokalizacije objekta na slici. Dakle, mjeri se točnost poklapanja graničnog okvira kojeg je model postavio (engl. *bounding box*) s temeljnim položajima objekata (engl. *ground truth*). Kao područje preklapanja (engl. *Area of Overlap*) definira se preklapajuća površina između graničnog okvira i temeljnog položaja. Područje unije (engl. *Area of Union*) predstavlja ukupnu površinu koju zauzimaju oba okvira zajedno. Područje preklapanja i područje unije se stavlja u omjer.

$$IoU = \frac{\text{područje preklapanja } (A \cap B)}{\text{područje unije } (A \cup B)} \quad (3.4)$$




Slika 3.4. IoU, područje preklapanja, područje unije na primjeru
(<https://showcase.itcarlow.ie/C00266169/documents/Final%20Report.pdf>, 2024.)

IoU se računa za svaku klasu posebno te rezultira vrijednošću između 0 i 1. Savršeno preklapanje ocjenjuje se vrijednošću od 1, a predviđanje bez preklapanja s 0. Općenito, prag vrijednosti postavlja se na 0,5. Ako je IoU vrijednost manja od 0,5, detekcija se smatra lažno pozitivnom, što znači da model nije dovoljno precizan. Ako je IoU vrijednost veća ili jednaka

0,5, detekcija se smatra istinito pozitivnom, što ukazuje na uspješnu lokalizaciju objekta [24]. Prethodno objašnjen odnos preciznosti i osjetljivosti može se grafički prikazati pomoću preciznost-odaziv (engl. *precision-recall*) krivulje. Ova krivulja nam pomaže u računanju srednje preciznosti (engl. *average percision, AP*) (Slika 3.5) koja mjeri učinkovitost modela u prepoznavanju logotipa za određenu klasu. Uzimajući IoU kao definirani prag, detekcija se smatra točnom ako je on veći ili jednak 0,5. AP vrijednosti s ovim pragom označavaju se sa AP50%. Dakle, detekcija je uspješna ako predviđeni granični okvir logotipa prekriva temeljni položaj logotipa barem u 50% površine. S ovim pragom se zatim generira preciznost-odaziv krivulja $p(r)$. AP tada odgovara površini ispod te krivulje [21]. Nadalje, srednji prosjek AP vrijednosti za sve klase (engl. *Mean Averege Precision, mAP*) objekata omogućuje cjelovitu procjenu modela na cijelom skupu podataka, odnosno pokazuje koliko dobro model funkcionira među različitim klasama (Formula 3.5). Varijabla K označava sve klase koje model može prepoznati [22]. Ukoliko je visoka mAP vrijednost to znači da je model dobar u prepoznavanju i lokalizaciji logotipa.



Slika 3.5. Primjer preciznost-odaziv krivulje

(<https://www.neuralception.com/objectdetection-evaluationmetrics/>, 2024.)

$$mAP = \frac{1}{K} \sum_{i=1}^K \int_0^1 p_i(r_i) dr \quad (3.5)$$

4. Praktični zadatak

4.1. Odabir arhitekture

Za detekciju logotipa u praktičnom zadatku koristila se adaptirana arhitektura YOLOv7 modela. Ova verzija donosi nekoliko promjena u arhitekturi koje nadmašuju prethodnike u brzini i točnosti detekcije. Arhitektura YOLOv7 temelji se na proširenoj učinkovitoj mreži za agregaciju slojeva (engl. *Extended Efficient Layer Aggregation Network, E-ELAN*) koja unapređuje učenje modela i smanjuje potrebu za velikom memorijom. Također, novom verzijom modela uvodi se koncept „*bag-of-freebies*“ koji uključuje razne optimizacije i metode koje pospješuju performanse bez povećanja vremena potrebnog za donošenje predikcije **Pogreška! Izvor reference nije pronađen.** Korišten je predefiniрани model (engl. *base model*) p od nazivom *yolov7_training.pt*. Radi se o osnovnoj verziji modela treniranog na velikom skupu podataka (primjerice COCO dataset), a ovdje se koristi kao polazna točka za treniranje baze s logotipovima. Učitava se pomoću programskog jezika Python koristeći *PyTorch*. Pytorch je okvir za duboko učenje koji olakšava i ubrzava razvoj i testiranje modela za detekciju logotipa [25].

4.2. Odabir baze za učenje i anotacija

Za potrebe ovog praktičnog zadatka odabrana je manja Flickr27 baza za učenje jer svojom veličinom omogućuje brzo treniranje modela i lakšu procjenu učinkovitosti modela. U dvadeset i sedam klasifikacija logotipa spadaju poznati brendovi automobila (*BMW, Citroen, Ferrari, Ford, Porsche, Mini*), popularni brendovi sportske odjeće i obuće (*Adidas, Puma, Nike*), brendovi razne elektronike i tehnologije (*Apple, Google, HP, Intel, Vodafone, Yahoo*), proizvodi hrane i pića (*Coca Cola, Pepsi, Heineken, McDonalds, Red Bull, Sprite, Starbucks*), brendovi povezani s transportom (*DHL, Fedex, Texaco*) te logotipi zastupljeni u medijima (*NBC, UNICEF*). Za postavljanje baze Flickr27 u terminalu koristi se *getFlickr.sh* skripta napisana u Shell jeziku koja automatizirano preuzima i postavlja bazu u predodređen direktorij (Programski kod 4.1).

```
sh getFlickr.sh
```

Programski kod 4.1. Postavljanje baze Flicker27

Iako baza Flickr27 dolazi s gotovim anotacijama koje identificiraju logotipe u slikama, one moraju biti u formatu koji je kompatibilan s modelom koji se koristi. YOLO modeli koriste

anotacije u posebnom tekstualnom formatu gdje svaka linija predstavlja jedan objekt s informacijama o klasi (npr. „Nike“) i koordinatama graničnih okvira (x i y koordinate gornjeg lijevog kuta, širina i visina okvira). Stoga je potrebno provesti konverziju anotacija skupa

```
python src/convert_annotations.py --dataset flickr27
```

podataka u format koji koristi YOLOv7 (Programski kod 4.2).

Programski kod 4.2. Konverzija anotacija baze Flickr27

4.3. Podjela baze u tri dijela (treniranje/validacija/testiranje)

Nakon postavljanja baze potrebno je podijeliti podatke na tri dijela (dio za trening, validaciju i testiranje). Ovo je ključan korak u razvoju i evaluaciji modela jer se time model ne prilagođava specifičnom skupu podataka, već uči prepoznati generalne obrasce na širokom rasponu situacija (Programski kod 4.3).

```
python src/prepare_data.py --dataset flickr27
```

Programski kod 4.3. Podjela baze na tri djela (trening, validacija, testiranje)

Dio podataka za trening koristi se za učenje parametara kako bi model mogao prepoznati određene obrasce u podacima. Tijekom treninga, model pokušava smanjiti pogreške kroz iterativni proces optimizacije. S obzirom na to da model treba što više podataka za učenje pravilnosti, uzima se najveći dio ukupnog skupa podataka (80%) [27]. Kako bi se osigurala optimalna izvedba, važno je validirati model na određenom skupu podataka (10%) pri čemu se provjerava koliko je točan u prepoznavanju podataka koji nisu bili uključeni u trening. Kroz validaciju se testiraju i postavljaju hiperparametri (npr. brzina učenja i broj slojeva) i odabiru one postavke koje daju najbolje rezultate. Posljednji skup podataka (10%) koristi se za konačnu evaluaciju modela. Kroz testiranje daje se neovisna procjena kako će model raditi na stvarnim, njemu nepoznatim podacima.

4.4. Treniranje modela

U postupku treniranja detekcije logotipa, model uči prepoznati različite karakteristike logotipa analizirajući mnoge slike sa sličnim logotipima. Time postaje sposoban točno identificirati i klasificirati brendove na slikama koje nije vidio tijekom treniranja (Programski kod 4.4).

```
python src/yolov7/train.py --img-size 640 --cfg
src/cfg/training/yolov7.yaml --hyp data/hyp.scratch.custom.yaml --
batch 2 --epoch 300 --data data/logo_data_flickr.yaml --weights
src/yolov7_training.pt --workers 2 --name yolo_logo_det_custom --
device 0
```

Programski kod 4.4. Kod komandne linije za pokretanje treniranja modela

U nastavku su navedeni parametri naredbe za pokretanje treniranja modela:

- `python src/yolov7/train.py`: Osnovna naredba za pokretanje Python skripte nazvane „train.py“ koja sadrži sav kod potreban za treniranje YOLOv7 modela na danom skupu podataka.
- `--img-size 640`: Parametar za postavljanje veličine ulaznih slika za model. Veličina slike je iznimno važna jer utječe na količinu informacija koje model prima, kao i vrijeme potrebno za treniranje. Postavljanje veličine na 640 piksela pruža dobar kompromis između detaljnosti slike i performansi modela.
- `--cfg src/cfg/training/yolov7.yaml`: Radi se o specifičnoj putanji do YAML konfiguracijske datoteke. Ona sadrži informacije o arhitekturi mreže, slojevima, skupu podataka, nazivu klasa i drugim postavkama.
- `--hyp data/hyp.scratch.custom.yaml`: Parametar koji određuje put do YAML datoteke sa hiperparametrima za proces treniranja. Sadrži postavke kao što su brzina učenja, regularizacija težina i sl.
- `--batch 2`: Veličina serije (engl. batch size) označava broj slika koje se istovremeno obrađuju u svakom prolazu unaprijed i unatrag tijekom treninga. U ovom primjeru model će obraditi dvije slike u svakoj iteraciji treninga. Ovakva manja veličina serije pomaže kod ograničenog memorijskog kapaciteta grafičke kartice računala.
- `--epoch 300`: Postavljanje broja epoha za trening. Jedna epoha označava jedan prolaz kroz cijeli skup podataka za trening. To znači da će model 300 puta pregledati skup podataka za trening.
- `--data data/logo_data_flickr.yaml`: Put do YAML datoteke koja sadrži informacije o putanjama do slika i anotacija te podjelu skupa podataka na dijelove za trening, validaciju i testiranje.
- `--weights src/yolov7_training.pt`: Put do definiranog modela (`yolov7_training.pt`) koji sadrži unaprijed trenirane parametre modela, koji se koriste kao početna točka za daljnje treniranje.
- `--workers 2`: Ovaj parametar specificira broj procesorskih jezgri (engl. Central Processing Unit, CPU) koje paralelno učitavaju i obrađuju podatke. Povećavanjem ovog parametra može se ubrzati učitavanje podataka.

- --name yolo_logo_det_custom: Određuje naziv sesije treninga kako bi se lakše organizirali rezultati treniranja.
- --device 0: Parametar koji postavlja uređaj na kojem se izvodi trening (npr. grafička kartica). Ako je postavljen na `0`, označava da će se koristiti prva dostupna grafička kartica.

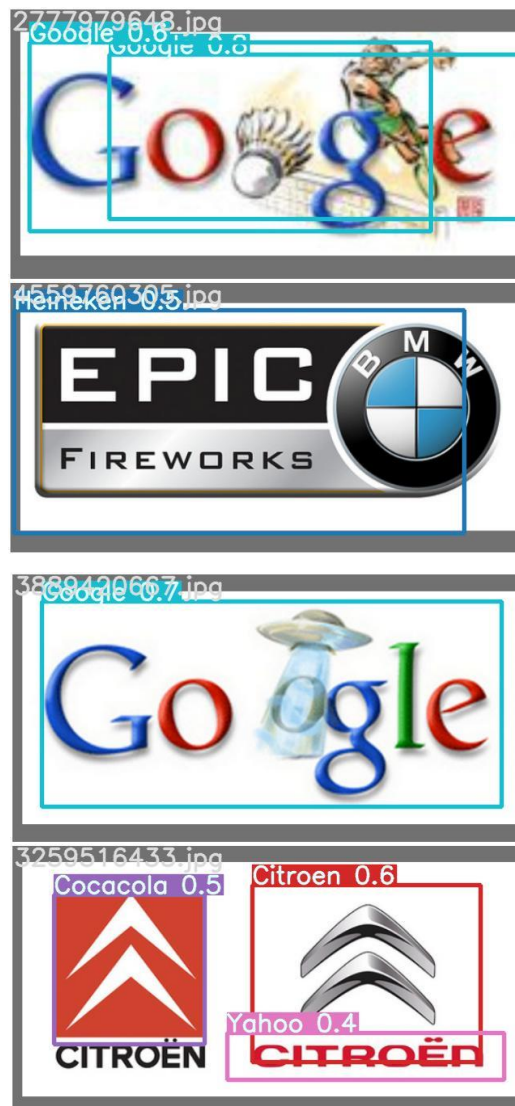
4.5. Evaluacija rezultata treniranja modela

Logotip	Broj slika	Broj klasa	P (preciznost)	R (osjetljivost)	mAP@0.5	mAP@0.5:0.95
Ukupno	108	109	0,612	0,49	0,529	0,32
Adidas	108	5	0,875	0,8	0,795	0,52
Apple	108	5	0,396	0,2	0,386	0,232
BMW	108	2	0	0	0,0738	0,0313
Citroen	108	9	0,833	0,555	0,718	0,394
Coca Cola	108	10	0	0	0,0598	0,0167
DHL	108	6	0,97	0,667	0,675	0,18
Fedex	108	5	0,588	0,305	0,492	0,273
Ferrari	108	4	0,374	0,5	0,407	0,262
Ford	108	2	0,379	0,5	0,512	0,401
Google	108	5	0,213	0,6	0,348	0,185
Heineken	108	1	1	0	0,142	0,0427
HP	108	3	1	0,623	0,995	0,3
Intel	108	1	0,316	1	0,995	0,896
McDonalds	108	2	1	0	0,249	0,0851
Mini	108	3	0,908	0,667	0,736	0,423
NBC	108	3	0,928	1	0,995	0,786
Nike	108	4	0,869	0,75	0,745	0,231
Porsche	108	2	0,871	1	0,995	0,796
Puma	108	5	0,292	0,2	0,229	0,159
RedBull	108	3	0,998	0,667	0,665	0,549
Sprite	108	1	0	0	0	0
Starbucks	108	7	0,869	0,571	0,594	0,306
Texaco	108	4	0,961	1	0,995	0,677
UNICEF	108	6	0,393	0,333	0,164	0,112
Vodafone	108	8	0,415	0,125	0,222	0,0745
Yahoo	108	3	0,476	0,667	0,571	0,386

Tablica 4.1. Rezultati treniranja modela

Rezultati treniranja (Tablica 4.1) provedeni su na skupu podataka koji se sastoji od 108 slika, pri čemu je model odradio 109 klasifikacija. Rezultati pokazuju uspješnost u prepoznavanju različitih logotipa. Ukupni mAP@0.5 (srednja prosječna preciznost pri 0.5 IoU) iznosi 52.9%, dok je stroži kriterij, mAP@0.5:0.95, postigao 32.0%. Rezultati ukazuju na to da model nije jednako precizan pri prepoznavanju svih logotipa. Primjerice, jednostavniji i jasniji logotipi poput BMWa (99,8%), Pepsija (81,6%), Adidasa (79.5%) i DHLa (67.5%), prepoznati su s visokim stupnjem preciznosti. Model se suočava s poteškoćama u prepoznavanju složenijih

logotipa poput Spritea(0%), RedBulla (0%), Heinekena(27,3%) i Coca Cole (5.98%). Nadalje, slikovni rezultati su pokazali da model ponekad u složenijim slikama prepoznaje različite dijelove logotipa kao zasebne instance (npr. Google) te da ima poteškoće u klasifikaciji logotipa koji su slični po oblicima ili boji (npr. pogrešno klasificiranje logotipa Citroena u „Cocacola“ ili „Yahoo“) (Slika 4.1). Ovakvi lažno pozitivni rezultati zahtijevaju dodatno usavršavanje modela uz pomoć većeg skupa podataka kako bi se poboljšala njegova preciznost detekcije.

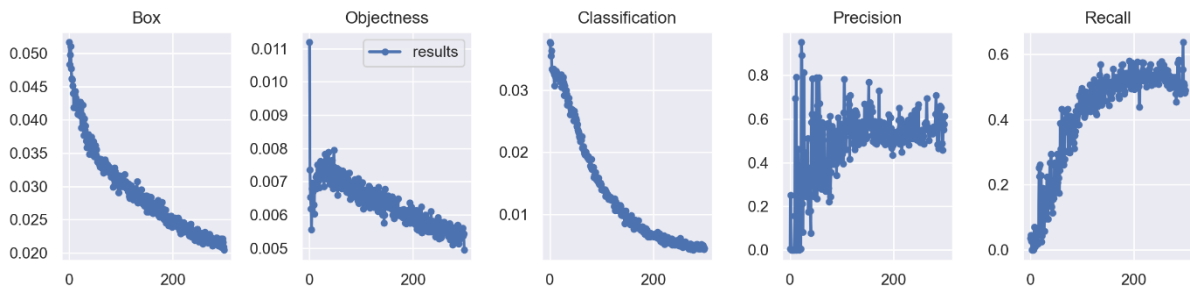


Slika 4.1. Primjer rezultata treniranja modela (Google, BMW, Citroen)

Rezultat treniranja uključuje i stvaranje nekoliko izlaznih datoteka, od kojih je najvažnija datoteka s težinama (engl. weights). U njoj su sadržani naučeni parametri, uključujući težine i pomake (engl. offsets), koje je model naučio tijekom treniranja. Navedena datoteka s težinama binarnog je formata te se pohranjuje na disku (u ovom slučaju radi se o datoteci veličine 75

MB). Ona je potrebna za daljnji razvoj modela jer se koristi u predikciji na novim podacima u procesu testiranja.

U izlazu rezultata su sadržani i specifični rezultati mjera prikazani u obliku grafova koji se mogu proučavati za daljnje istraživanje (Slika 4.2).



Slika 4.2. Grafovi evaluacije treninga YOLOv7 modelom

Graf gubitka okvira (engl. *box loss*) pokazuje koliko predviđeni granični okviri odstupaju od temeljnih položaja logotipa. Može se primijetiti da gubitak okvira pada kako se broj epoha povećava, što ukazuje na to da model postaje precizniji u određivanju točne lokacije logotipa.

Graf gubitka za sigurnost detekcije (engl. *objectness loss*) mjeri koliko je model siguran da je u predviđenom graničnom okviru prisutan logotip. Što je gubitak manji, to model postaje sigurniji i bolji u razlikovanju objekta i pozadine. Graf pokazuje pad gubitka sigurnosti.

Graf gubitka za klasifikaciju (engl. *classification loss*) prikazuje koliko su predviđanja klase logotipa točna. Graf prikazuje pad vrijednosti, što znači da model s vremenom postaje sve točniji u klasifikaciji logotipa.

Graf preciznosti pokazuje omjer točno predviđenih pozitivnih rezultata naprema svim pozitivnim predviđanjima. U ovom primjeru se vide oscilacije s tendencijom rasta, što ukazuje na to da model prolaskom kroz epohe postaje precizniji te je da sve manje lažno pozitivnih rezultata. S druge strane, graf osjetljivosti prikazuje omjer ispravno identificiranih logotipa naprema ukupnom broju stvarnih pozitivnih uzoraka. Rast sugerira da model postaje bolji u prepoznavanju svih stvarnih logotipa u slici, smanjujući broj promašenih logotipa.

4.6. Testiranje modela

Proces testiranja je ključan za procjenu stvarne sposobnosti modela u prepoznavanju logotipa izvan skupa podataka korištenog za treniranje. Ovime se osigurava njegova pouzdanost u korištenju u stvarnim okolnostima (Programski kod 4.5).

```
python src/yolov7/test.py --data data/logo_data_flickr.yaml
--img 640 --batch 32 --conf 0.001 --iou 0.65 --device 0 --
weights runs/train/yolo_logo_det_custom/weights/best.pt --
task test --name volo_logo_det_custom
```

Programski kod 4.5. Kod komandne linije za pokretanje testiranja modela

U nastavku su navedeni parametri naredbe za pokretanje testiranja modela:

- `python src/yolov7/test.py`: Osnovna naredba koja pokreće Python skriptu za testiranje modela, smještenu u direktoriju `src/yolov7`.
- `--data data/logo_data_flickr.yaml`: Definira putanju do YAML datoteke koja sadrži informacije o skupu podataka, kao što su putanje do slika i klase.
- `--img 640`: Skalira ulazne slike na 640 piksela prije nego što budu proslijeđene modelu.
- `--batch 32`: Definira veličinu serije. Dok se u prethodnom treniranju koristila manja veličina, u testiranju je cilj procijeniti kako model generalizira na novim podacima. Zbog toga se može koristiti veća serija kako bi se optimiziralo vrijeme testiranja.
- `--conf 0.001`: Ovim parametrom se postavlja prag odluke na 0.001. To znači da će detekcije s pragom manjim od ove vrijednosti biti ignorirane.
- `--iou 0.65`: Postavlja prag presjeka nad unijom (IoU) na 0,65. Ova vrijednost određuje koliko se predviđena granica logotipa mora preklapati s stvarnom granicom logotipa da bi se smatrala ispravnom detekcijom.
- `--device 0`: Parametar koji postavlja uređaj na kojem se izvodi trening (npr. grafička kartica). Ako je postavljen na `'0'`, označava da će se koristiti prva dostupna grafička kartica.
- `--weights runs/train/yolo_logo_det_custom/weights/best.pt`: Definira putanju do datoteke s težinama modela. U ovoj datoteci sadržan je model koji će biti testiran.
- `--task test`: Parametar koji definira zadatak kao testiranje. To znači da će se model testirati na novom testnom skupu podataka, a ne trenirati ili validirati.
- `--name yolo_logo_det_custom`: Određuje naziv sesije testiranja.

4.7. Evaluacija rezultata testiranja modela

Logotip	Broj slika	Broj klasa	P (preciznost)	R (osjetljivost)	mAP@.5	mAP@.5:.95
Sveukupno	108	150	0,557	0,638	0,612	0,368
Adidas	108	2	0,731	1	0,995	0,821
Apple	108	4	0,613	0,25	0,397	0,281
BMW	108	6	0,998	1	0,995	0,424
Citroen	108	8	0,723	0,334	0,425	0,241
Coca Cola	108	42	0,495	0,0937	0,254	0,119
DHL	108	3	0,58	0,667	0,665	0,301
Fedex	108	1	0,253	1	0,995	0,697
Ferrari	108	2	0,703	1	0,995	0,547
Ford	108	2	0,471	0,5	0,515	0,4
Google	108	2	0,356	0,5	0,495	0,396
Heineken	108	5	0,273	0,6	0,579	0,367
HP	108	1	0,322	1	0,995	0,398
Intel	108	2	0,489	0,966	0,498	0,398
McDonalds	108	3	0,457	0,333	0,188	0,075
Mini	108	2	0,427	1	0,745	0,534
NBC	108	4	0,37	0,737	0,442	0,119
Nike	108	3	1	0	0,121	0,0129
Porsche	108	28	0,816	0,857	0,889	0,57
Puma	108	3	0,643	1	0,913	0,681
RedBull	108	2	0	0	0,076	0,0307
Sprite	108	3	0,62	0,333	0,47	0,175
Starbucks	108	3	0,55	1	0,995	0,929
Texaco	108	5	1	0,798	0,795	0,416
UNICEF	108	7	0,346	0,571	0,378	0,228
Vodafone	108	4	0,638	0,457	0,456	0,255
Yahoo	108	3	0,638	1	0,641	0,164

Tablica 4.2. Rezultati testiranja modela

Rezultati testiranja (Tablica 4.2) provedeni su na skupu podataka koji se sastoji od 108 slika, pri čemu je model odradio 150 klasifikacija. Umjerena preciznost (0,557) i osjetljivost (0,638) ukazuju na to da model ponekad pogrešno detektira logotipe ili propušta neke od njih. Također, model ima bolju točnost pri kriteriju mAP@.5, ali njegova točnost u detekciji opada kad se razmatra stroži kriterij mAP@.5:.95. Slično kao i kod treniranja, jednostavni logotipovi BMW, Adidas, Porsche, Texaco imaju vrlo visoku preciznost i osjetljivost, što znači da ih model gotovo uvijek prepoznaje (Slika 4.3, 4.4).



Slika 4.3. Uspješna detekcija logotipa BMW (preciznost 0,7)



Slika 4.4. Uspješna detekcija logotipa Adidas (preciznost 0,94)

S druge strane, RedBull, Sprite, McDonald's i Coca Cola pokazuju lošije rezultate. Primjerice, RedBull ima preciznost i osjetljivost vrijednosti 0, što znači da model nije uspio ni jednom uspješno prepoznati i klasificirati ovaj logotip. Razlog tomu je, kao i kod treniranja modela, kompleksnost logotipa, ali i vizualna sličnost s drugim logotipima (Slika 4.5., 4.6,4.7).



Slika 4.5. Neuspješna detekcija logotipa RedBull (pogrešna klasifikacija Coca Cola, preciznost 0,6)



Slika 4.6. Neuspješna detekcija logotipa Citroen (pogrešna klasifikacija Coca Cola, preciznost 0,57)



Slika 4.7. Neuspješna detekcija logotipa Volkswagen (pogrešna klasifikacija Unicef, preciznost 0,5)

Nadalje, propusti u detekciji pojedinih logotipa mogu biti i rezultat kombinacije faktora kao što su kut i orijentacija logotipa. Primjerice, model nema problema s prepoznavanjem UNICEF logotipa ako je on jasno izražen npr. na uniformi volontera (Slika 4.8). No, ako je na slici logotip postavljen pod kutom i djelomično zakrenut, model jednostavno nema dovoljno primjeraka logotipa pod sličnim uvjetima da bi ga mogao ispravno detektirati (Slika 4.9).



*Slika 4.8. Uspješna detekcija logotipa UNICEF
(preciznost 0,54, 0,48)*



*Slika 4.9. Neuspješno prepoznavanje logotipa
UNICEF s obzirom na njegovu orijentaciju*

Model ponekad propušta detekciju određenih logotipa zbog nedostatka dovoljnog broja uzoraka tog logotipa u trening skupu. Na primjer, iako Nike postiže maksimalnu preciznost (1), njegova je osjetljivost vrlo niska (0). Ovakav rezultat ukazuje na to da je model siguran u to da se radi o Nike brendu, ali nema dovoljno uzoraka da bi to mogao konzistentno potvrditi. Također, neki logotipovi nisu prepoznati u testiranju jer se na slikama nalaze zastarjele verzije koje se razlikuju u dizajnu u odnosu na specifičnu verziju na kojoj je model treniran (Slika 4.10, 4.11). To iziskuje uključivanje više raznolikijih primjera u skup za treniranje. Takvi koraci proširenja baze podataka, uz fino podešavanje modela, mogli bi poboljšati rezultate u detekciji.



Slika 4.10. Neuspješno prepoznavanje svih verzija logotipa Heineken



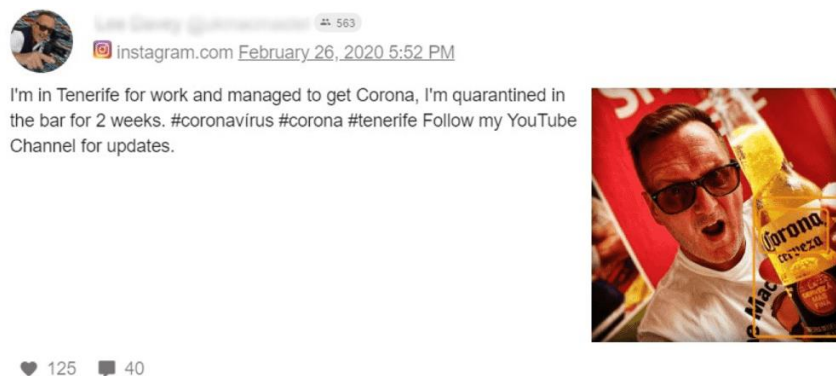
Slika 4.11. Primjer uspješnog prepoznavanja različitih verzija logotipa Pepsi

5. Diskusija

Kako se razvija tehnologija detekcije logotipa tako raste i njezina primjena, posebice u digitalnom marketingu, gdje je s vremenom postala ključni alat za analizu tržišta i potrošača. U narednim poglavljima diskusije pruža se detaljan uvid u to kako se detekcija logotipa primjenjuje u digitalnom marketingu za izradu učinkovite marketinške strategije te se analiziraju ključni tehnički izazovi s kojima se ova tehnologija suočava. Na posljetku, razmatra se smjer razvoja detekcije logotipa u digitalnom okruženju u nadolazećim godinama.

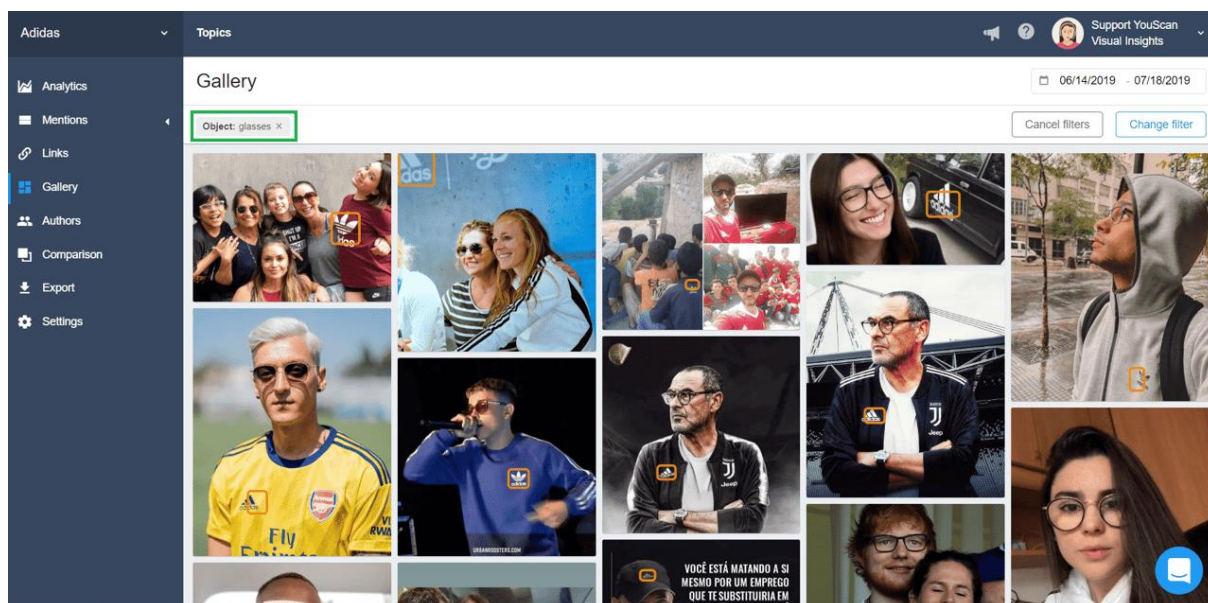
5.1. Primjena detekcije logotipa u digitalnom marketingu

U kontekstu digitalnog marketinga, detekcija logotipa se sve više koristi za optimizaciju raznih marketinških strategija. Ona omogućava precizno praćenje i analizu prisutnosti nekog brenda u digitalnom okruženju kako bi se bolje razumjela njegova percepcija i osigurala zaštita od potencijalne zlouporabe. Praćenje brenda (engl. *brand monitoring*) podrazumijeva sustavno i kontinuirano nadgledavanje prisutnosti brenda na različitim digitalnim platformama, uključujući društvene mreže, web stranice, blogove i video platforme [28]. Zahvaljujući automatiziranoj detekciji logotipa, brendovi mogu učinkovitije identificirati svoje logotipe u korisnički generiranom sadržaju (engl. *user generated content, UGC*), odnosno u objavama koje korisnici postavljaju, dijele ili s kojima komuniciraju (slike, videozapisi, prijenos uživo). Ovi podaci su izrazito vrijedni marketinškim stručnjacima jer im omogućuju analizu učinka marketinških kampanja kao i prepoznavanje novih prilika za promociju. Prikupljanjem takvih podataka, brend može analizirati koliko se često njegov logotip pojavljuje u objavama korisnika na društvenim mrežama tijekom ili nakon kampanje. Na ovaj način može kvantificirati utjecaj kampanje i prilagoditi svoje strategije u budućnosti [29]. Analizom sadržaja na društvenim mrežama, brendovi nadalje mogu identificirati specifične demografske grupe ili lokacije u kojima se često pojavljuje njihov logotip, što doprinosi boljem razumijevanju tržišta, segmentaciji korisnika i prilagodbi marketinških poruka za različite ciljane skupine. Također, detekcija logotipa pruža uvid u kontekst u kojem se logotip pojavljuje. Ovi podaci omogućuju marketinškim stručnjacima brzo reagiranje na pozitivne ili negativne pojave brenda u medijima, čime se poboljšava reputacija brenda i njegova prisutnost na tržištu [30]. Zahvaljujući autentičnom prikazu proizvoda u raznim objavama korisnika, brendovi mogu saznati preferencije korisnika, što otvara vrata personaliziranim marketinškim strategijama temeljenim na tim preferencijama, osiguravajući bolju povezanost s ciljanom publikom (Slika 5.1).



Slika 5.1. Primjer korisnički generirane objave na Instagramu koja sadrži Corona proizvod (<https://youscan.io/blog/logo-identification-in-marketing/>, 2024.)

Odličan primjer toga jest nedavno istraživanje koje je proveo *YouScan*, platforma koja se bavi praćenjem i analizom sadržaja na društvenim mrežama (Slika 5.2). Oni su, potpomognuti detekcijom logotipa, proučavali ciljne publike glavnih pivskih brendova poput *Bud Light*, *Miller Lite*, *Budweiser*, *Corona*, *Guinness*, *Coors Light*, *Heineken* i *Stella Artois*. Analizom načina na koji su se ovi logotipi pojavljivali u objavama korisnika, *YouScan* je otkrio jedinstvene karakteristike potrošača povezanih s pojedinim brendovima. Tako su primjerice, američki potrošači Corone osobe koje se povezuju sa skijanjem, plovidbom i obiteljskim interakcijama, dok osobe koje piju Stella Artois često vole Wimbledon, jer imidž ovog brenda i događaja odgovara njihovoj sklonosti prema sofisticiranim i elitnim aktivnostima [31]. Identificiranjem takvih karakteristika, brendovi mogu kreirati relevantne i zanimljive kampanje koje jače odjekuju kod njihovih ciljanih demografskih skupina.

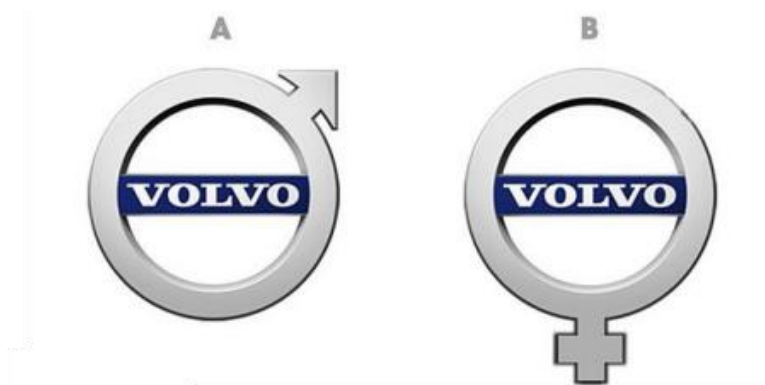


Slika 5.2. Sučelje *YouScan* platforme (<https://youscan.io/blog/top-10-social-media-scanners-with-logo-monitoring-capabilitie/>, 2024.)

Još jedan od ključnih razloga za korištenje detekcije logotipa u digitalnom marketingu jest prevencija prevarantskih aktivnosti i zaštita brenda u digitalnom svijetu. Logotipovi, kao vizualni identiteti brendova, pružaju digitalnom sadržaju osjećaj prepoznatljivosti i potiču povjerenje korisnika. Stoga su nerijetko meta prevaranata koji ih neovlašteno koriste ili modificiraju za potrebe zlonamjernih web stranica u ime renomiranih organizacija, lažnih kampanja ili pak u izradi krivotvorenih proizvoda, s ciljem obmane korisnika i stjecanja financijske dobiti [32]. Zlonamjerna upotreba logotipa dovodi do kršenja intelektualnog vlasništva, što može narušiti reputaciju brenda i uzrokovati značajne financijske gubitke. Sustavi za detekciju logotipa analiziraju sadržaj na digitalnim platformama kako bi utvrdili je li logotip autentičan ili krivotvoren (Slika 5.3, 5.4). Usporedbom detektiranog logotipa s bazom autentičnih logotipa, sustav može brzo identificirati neovlaštenu upotrebu [33]. Također, detekcija logotipa omogućava prikupljanje dokaza potrebnih za pokretanje pravnih postupaka protiv krivotvoritelja, što je ključno za povredu prava intelektualnog vlasništva i odvracanje od budućih povreda.



Slika 5.3. Primjer originalnog BMW logotipa (A) i krivotvorenog, modificiranog logotipa (B).
(Tanniru, 2023.)



Slika 5.4. Primjer originalnog VOLVO logotipa (A) i krivotvorenog, modificiranog logotipa (B).
(Tanniru, 2023.)

5.2. Izazovi u detekciji logotipa

Iako je detekcija logotipa postigla značajan napredak, njezina primjena se i dalje suočava s brojnim tehničkim i praktičnim izazovima. Ove prepreke ističu složenost postizanja točnosti i pouzdanosti detekcije, posebice kad su u pitanju složenije scene iz stvarnog svijeta. Razumijevanje i rješavanje ovih prepreka ključno je za razvoj učinkovitih modela koji se mogu primijeniti u raznim kontekstima stvarnog svijeta. Najčešći izazov s kojim se susreću modeli za detekciju je mala veličina logotipa u odnosu na cjelokupnu sliku. Mala veličina ograničava količinu vizualnih informacija koje su dostupne, što otežava izdvajanje značajki logotipa koje su potrebne za točno detektiranje (Slika 5.5).



Slika 5.5. Primjer pogrešne detekcije logotipa zbog male veličine.

S obzirom na veličinu logotipa, postoji i znatno manji prostor za pogrešku lokalizacije. Često puta se u složenim scenama takvi logotipi stope s pozadinom ili drugim elementima u slici čime se otežava točna lokalizacija u usporedbi s većim objektima. Nadalje, logotipi se pojavljuju na različitim proizvodima i površinama. Svaka takva površina utječe na njegov izgled, čineći ga različitim ovisno o kontekstu pojavljivanja. Tako se isti logotip može naći okružen različitim bojama, teksturama i uzorcima što otežava algoritmima generalizaciju preko ovih varijacija.

Sve češće se pojavljuju pod-brendovi, koji su povezani s matičnim brendom i obično dijele s njim određene karakteristike logotipa. Suptilne i teško uočljive razlike koje se pojavljuju između njih mogu dovesti do zabune u detekciji, posebice kad su prikazani u sličnim kontekstima. Ovakvi zadaci prepoznavanja logotipa zahtijevaju finu klasifikaciju, odnosno visoku razinu preciznosti u prepoznavanju sitnih razlika u dizajnu.



Slika 5.6. Primjer pod-brendova britanskog Virgin konglomerata
(<https://brogan.com/blog/pros-and-cons-sub-branding-and-brand-extension/>, 2024.)

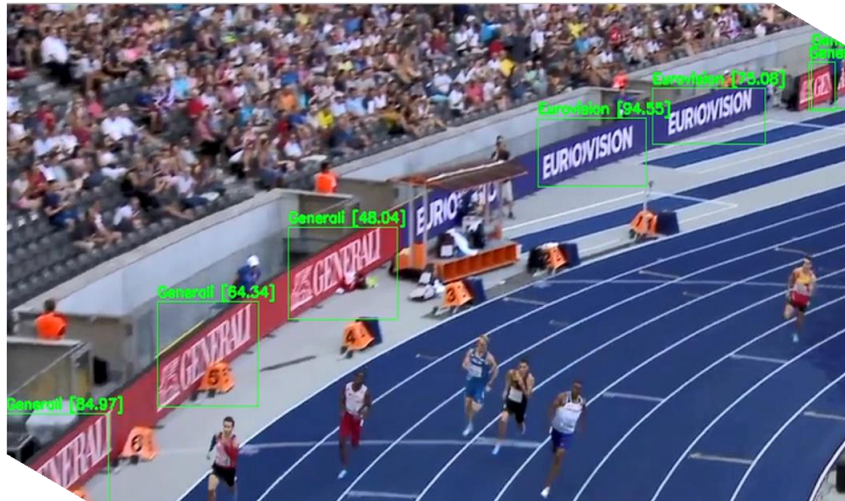
U stvarnim okolnostima, neki su brendovi i logotipi popularniji te se zbog toga pojavljuju češće od drugih. Neuravnotežena distribucija uzoraka može uzrokovati probleme prilikom treniranja algoritama za detekciju, jer modeli mogu postati pristrani prema takvim logotipima i lošije prepoznavati rijetke logotipe. Ovaj izazov je posebno izražen u primjenama gdje je ključno pouzdano detektirati sve vrste logotipa, bez obzira na njihovu učestalost. Također, modeli se moraju suočiti i s prepoznavanjem novih, neviđenih logotipa. Većina modela je obučena na zatvorenoj bazi podataka, pri čemu su klasifikacije logotipa unaprijed poznate. Takvi modeli, kada se suoče s nepoznatim logotipima, imaju problem s klasifikacijom te često griješe u detekciji.

5.3. Budućnost detekcije logotipa

Budućnost detekcije logotipa u digitalnom marketingu usko je povezana s inovacijama u tehnologiji računalnog vida. Detekcija logotipa je s vremenom postala sastavni dio digitalnog marketinga. Vizualni sadržaj i podaci koji se prikupljaju detekcijom, ne samo da povećavaju učinkovitost marketinških kampanja, već i preoblikuju samu prirodu marketinške komunikacije s potrošačima. U kombinaciji s naprednim analitičkim alatima, detekcija logotipa postaje ključni alat za stvaranje učinkovitih i personaliziranih marketinških strategija.

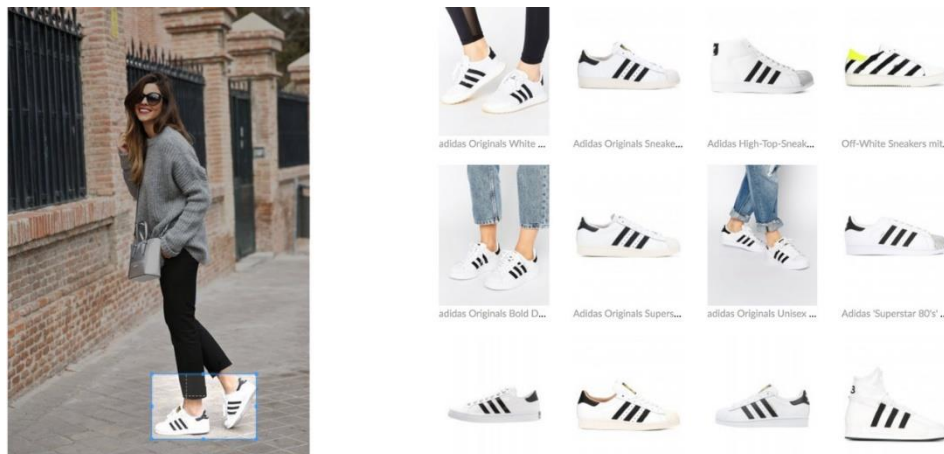
Kad je riječ o samoj detekciji logotipa, u prednjem planu su naponi za razvoj modela koji će brže i učinkovitije prepoznati sitne logotipe na proizvodima i ambalaži. Primjerice, kozmetički proizvodi često sadrže male etikete, pa je ključno razviti tehnologiju koja može precizno prepoznati i sitne logotipe unatoč lošijoj vidljivosti. Također, nastoji se razviti model koji uspješno prepoznaje potpuno nove ili rijetke logotipe koji su slabije zastupljeni u bazama korištenim za treniranje modela.

S obzirom na sve veću popularnost prijenosa uživo, posebice tijekom sportskih događaja, razvijaju se modeli koji automatski prepoznaju logotipe u stvarnom vremenu. Na taj način bi se učinkovito mogla analizirati izloženost brenda sponzora tijekom velikih sportskih događaja [20]. Nastavno s tim, fokus unutar digitalnog marketinga jest u unapređenju vremena potrebnog za analizu podataka dobivenih detekcijom logotipa. Detekcijom logotipa u stvarnom vremenu, brendovi bi postali agilniji i brži u reagiranju na promjene kod ponašanja potrošača, kod novih trendova na društvenim mrežama te u praćenju razvoja kampanja i njihovoj prilagodbi.



Slika 5.7. Primjena logo detekcije u prijenosu utrke za detektiranje sponzora (<https://www.onlinecontrol.eu/logo-detection-control/>, 2024.)

Daljnji razvoj detekcije logotipa je važan i za vizualnu pretragu, koja je u međuvremenu postala uobičajeni alat za otkrivanje i kupovinu proizvoda u online trgovinama. Potrošači umjesto da upisuju ključne riječi i fraze u tražilice, prijenosom slike proizvoda odmah dobivaju relevantne rezultate, uključujući slične proizvode i brendove.



Slika 5.7. Vizualna pretraga proizvoda (<https://www.linkedin.com/pulse/visual-search-zalando-asos-matthias-dantone>, 2024.)

Unapređenje točnosti i brzine prepoznavanja logotipa iz različitih kutova i pod različitim uvjetima, u konačnici će omogućiti precizno pojavljivanje brendova u rezultatima pretraga te povezivanje s potrošačima u trenutku kada su najspremniji za kupovinu. S obzirom na to da potrošači postaju sve svjesniji načina na koji se njihovi podaci prikupljaju i koriste na digitalnim platformama, neprestano raste i zabrinutost za privatnost. Brendovi su svjesni da povjerenje potrošača uvelike ovisi i o tome koliko transparentno i odgovorno postupaju s njihovim podacima. Zbog toga se u posljednjim godinama naglašava problematika vezana uz etičku praksu u korištenju podataka korisnički generiranog sadržaja pri detekciji logotipa. Modeli za detektiranje logotipa se dizajniraju sa strožim principima zaštite korisničkih informacija što uključuje transparentnost u prikupljanju, obradi i pohranjivanju podataka u skladu s najvišim standardima etičkog postupanja.

6. Zaključak

U ovom diplomskom radu obuhvaćeno je cjelovito istraživanje dosadašnjih pristupa u detekciji logotipa s posebnim naglaskom na njezinoj primjeni u digitalnom marketingu. Kroz pregled razvoja različitih arhitektura modela dubokog učenja, kao što su konvolucijske neuronske mreže (CNN), dvofazni (R-CNN, brzi R-CNN, brži R-CNN) i jednofazni modeli (SSD i YOLO), analizirani su ključni elementi koji omogućuju detekciju logotipa.

Na temelju tih saznanja, odabrana je arhitektura modela YOLOv7 za praktični zadatak u kojem je uz pomoć Flick27 baze za učenje, izvedeno treniranje i testiranje modela za detekciju logotipa. Evaluacija rezultata pokazala je da dobro odabrana arhitektura modela u kombinaciji s kvalitetnom bazom može unaprijediti performanse detekcije logotipa u stvarnim slikama. Međutim, u radu se ističu i ključni izazovi s kojima se detekcija logotipa redovito suočava, poput slabije preciznosti u prepoznavanju malih i rijetkih logotipa, kao i složenosti scene i konteksta u kojoj se logotip nalazi.

Unatoč izazovima, ukazuje se na veliki potencijal za daljnji razvoj, posebice u kontekstu primjene u digitalnom marketingu, gdje detekcija logotipa postaje ključni alat za analizu tržišta, ponašanja potrošača i optimizaciju marketinških strategija.

Očekuje se da će daljnji razvoj detekcije logotipa omogućiti brendovima bolje razumijevanje gdje se sve i u kojem kontekstu pojavljuju njihovi logotipovi kako bi mogli lakše pratiti tržište i personalizirati sadržaj koji se prikazuje potrošačima, odnosno korisnicima na društvenim mrežama, web stranicama i video platformama.

Literatura

- [1] S. Sačer, "Povezanost vizualnih obilježja slika i osobina ličnosti korisnika pri promociji na društvenim mrežama", Zbornik Ekonomskog fakulteta u Zagrebu, vol. 20, no. 2, 2022., str. 47-60., doi: 10.22598/zefzg.2022.2.47
- [2] G. Bokhua, "Principles of Logo Design: A practical guide to creating effective signs, symbols, and icons", Quarto Publishing Group USA Inc., SAD, 2022.
- [3] J. Sanghvi, J. Rathod, S. Nemade, H. Panchal i A. Pavate, "Logo Detection Using Machine Learning Algorithm : A Survey", 2023 International Conference on Communication System, Computing and IT Applications (CSCITA), Mumbai, India, 2023., str. 136-141, doi: 10.1109/CSCITA55725.2023.10105056.
- [4] A. Geron, "Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow", Second Edition, O'Reilly Media, Inc., Kanada, 2019.
- [5] S. Vlahović, "Vizualizacija rada konvolucijske neuronske mreže", Završni rad, Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje, Zagreb, Hrvatska, 2020., <https://urn.nsk.hr/urn:nbn:hr:235:108019>, dostupno: 12.07.2024.
- [6] L. Mehinović, "Konvolucijske neuronske mreže za klasifikaciju objekata", Diplomski rad, Sveučilište Jurja Dobrile u Puli, Pula, Hrvatska, 2020., <https://urn.nsk.hr/urn:nbn:hr:137:811211>, dostupno: 12.07.2024.
- [7] R. Gandhi, "R-CNN, Fast R-CNN, Faster R-CNN, YOLO – Object Detection Algorithms", <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>, dostupno: 12.07.2024.
- [8] D. Parthasarathy, "A Brief History of CNNs in Image Segmentation: From R-CNN to Mask R-CNN", <https://blog.athelas.com/a-brief-history-of-cnns-in-image-segmentation-from-r-cnn-to-mask-r-cnn-34ea83205de4>, dostupno: 12.07.2024.
- [9] <https://www.geeksforgeeks.org/fast-r-cnn-ml/>, dostupno: 12.07.2024
- [10] R. Girshick, "Fast R-CNN," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Čile, 2015., str. 1440-1448, doi: 10.1109/ICCV.2015.169
- [11] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar i B. Lee, "A survey of modern deep learning based object detection models", Digital Signal Processing, vol. 126, 2022., 103514, doi: 10.1016/j.dsp.2022.103514
- [12] M. Voithofer, "Verwendung von Deep Learning für die Erkennung von Werbung am Straßenrand in Dashcam-Videos", diplomski rad, FH Upper Austria, Hagenberg Campus: Informatics, Communications and Media, Hagenberg, Austrija, <https://theses.fh-hagenberg.at/thesis/Voithofer19>, dostupno: 12.07.2024.
- [13] S.-H. Kang i J.-S. Park, "Aligned Matching: Improving Small Object Detection in SSD", Sensors, vol. 23, no. 5: 2589., 2023., doi: 10.3390/s23052589

- [14] M. Elshebly, "Autonomous Garbage Detection and Localization using Deep Learning and Computer Vision Techniques", završni rad, Faculty of Engineering and Computer Science, Hamburg, Njemačka, <https://reposit.haw-hamburg.de/handle/20.500.12738/14131>, dostupno: 12.07.2024.
- [15] D. Coiras , "Implementation of a logo detection system based on deep learning strategies for media impact analysis in social networks", diplomski rad, E.T.S.I. Telecomunicación (UPM), Madrid, Španjolska, <https://oa.upm.es/63132/>, dostupno: 12.07.2024.
- [16] J. Redmon, S. Divvala, R. Girshick i A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, SAD, 2016., str. 779-788, doi: 10.1109/CVPR.2016.91
- [17] X. Cong, S. Li, F. Chen, C. Liu i Y. Meng, "A Review of YOLO Object Detection Algorithms based on Deep Learning", FCIS, vol. 4, no. 2, 2023., str. 17–20, doi: 10.54097/fcis.v4i2.9730
- [18] "YOLO: You Look Only Once (The 1st Version)", <https://kikaben.com/yolo-version-1/#yolo-architecture>, dostupno: 12.07.2024.
- [19] W. Kuang, "Fundamentals of deep learning: a step-by-step guide. Chapter 10 Object Detection -- YOLO", <https://faculty.utrgv.edu/weidong.kuang/book/book.html>, dostupno: 12.07.2024.
- [20] S. C.H. Hoi, X. Wu, H. Liu, Y. Wu, H. Wang, H. Xue, Q.Wu. "LOGO-Net: Large-scale Deep Logo Detection and Brand Recognition with Deep Region-based Convolutional Networks", doi: 10.48550/arXiv.1511.02462
- [21] S. Hou, J.Li, "Deep Learning for Logo Detection: A Survey", ACM Transactions on Multimedia Computing, Communications and Applications, 2023., str. 1-23, doi:10.48550/arXiv.2210.04399
- [22] H. Wortmann, "Objekterkennung unter Nutzung von Machine Learning für Augmented Reality Anwendungen", diplomski rad, Fakultät Technik und Informatik, Hochschule für Angewandte Wissenschaften Hamburg, Hamburg, <https://reposit.haw-hamburg.de/handle/20.500.12738/15706>, dostupno: 12.7.2024.
- [23] G. Montorio, "Entwicklung einer Bildverarbeitung mit dem Schwerpunkt Personenerkennung für ein autonomes Logistik-Fahrzeug ", diplomski rad, Bochum University of Applied Sciences, Hochschule Bochum, Bochum, https://www.hochschule-bochum.de/fileadmin/public/Die-BO_Fachbereiche/fb_e/Institute_und_Labore/Antriebstechnik/Abschlussarbeiten/Entwicklung_einer_Bildverarbeitung_mit_dem_Schwerpunkt_Personenerkennung_fuer_ein_autonomes_Logistik-Fahrzeug.pdf , dostupno: 12.7.2024.
- [24] A. Anwar, "What is Average Precision in Object Detection & Localization Algorithms and how to calculate it?", <https://towardsdatascience.com/what-is-average-precision-in-object-detection-localization-algorithms-and-how-to-calculate-it-3f330efe697b> , dostupno: 27.7.2024.
- [25] "Evaluation Metrics", <https://www.neuralception.com/objectdetection-evaluationmetrics/> , dostupno: 27.7.2024.

- [26] C. -Y. Wang, A. Bochkovskiy and 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, doi: 10.1109/CVPR52729.2023.00721
- [27] "PyTorch", <https://www.nvidia.com/en-us/glossary/pytorch/> , dostupno: 27.7.2024.
- [28] B. Donmez, "Logo Detection – YOLOv7", <https://github.com/nuwandda/yolov7-logo-detection?tab=readme-ov-file> , dostupno: 27.7.2024.
- [29] D. Clark, "Brand Monitoring: Importance, Strategies, and Tools", 2024., <https://www.socialchamp.io/blog/brand-monitoring/>, dostupno: 28.8.2024.
- [30] O. Ori, R. Tous, M. Gomez, J. Poveda, L. Cruz, O. Wust, "Real-time Logo Detection in Brand-related Social Media Images", IWANN 2019 Conference, Gran Canaria, Španjolska, 2019., str. 125-136, doi: 10.1007/978-3-030-20518-8_11
- [31] H. Larsson-Broman, P. Siljerud, "AI driven marketing. Opportunities and tools to revolutionize your marketing strategy", Mercuri International Research AB, 2024., https://mercuri.de/wp-content/uploads/sites/2/2024/01/report_eng.pdf, dostupno: 28.8.2024.
- [32] A. Hura, "How we used social media listening to create customer personas for major beer brands", YouScan, 2020., <https://youscan.io/blog/how-we-used-social-media-listening-to-create-customer-personas-for-major-beer-brands/>, dostupno: 28.8.2024.
- [33] Y. Yadav, "Fake Logo Detection using Machine Learning", IRJETS, vol.6, Karnataka, Indija, 2024., https://www.irjmets.com/uploadedfiles/paper//issue_7_july_2024/60438/final/fin_irjmets1721927029.pdf, dostupno: 28.8.2024.
- [34] V. Tanniru, T. Bhattacharya, "Online Fake Logo Detection System", Research Square, 2023., doi: 10.21203/rs.3.rs-2492597/v1