

# Klasifikacija prometnih znakova korištenjem konvolucijskih neuronskih mreža

---

**Matoš, Ivona**

**Master's thesis / Diplomski rad**

**2020**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:723809>

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

*Download date / Datum preuzimanja:* **2025-03-14**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU**

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni studij**

**KLASIFIKACIJA PROMETNIH ZNAKOVA  
KORIŠTENJEM KONVOLUCIJSKIH NEURONSKIH  
MREŽA**

**Diplomski rad**

**Ivona Matoš**

**Osijek, 2020.**

# SADRŽAJ

1.UVOD .....	1
2.PRIMJENA STROJNOG I DUBOKOG UČENJA PRILIKOM PREPOZNAVANJA PROMETNIH ZNAKOVA.....	3
3.PREPOZNAVANJE OBJEKATA KORIŠTENJEM KONVOLUCIJSKIH NEURONSKIH MREŽA.....	9
3.1. Arhitektura konvolucijskih neuronskih mreža.....	9
3.1.1. Ulazni sloj .....	10
3.1.2. Konvolucijski sloj .....	11
3.1.3. Hiperparametri u konvolucijskom sloju.....	12
3.1.4. Opća formula za operaciju konvolucije i formiranje aktivacijske mape .....	13
3.1.5. Nelinearni sloj .....	14
3.1.6. Sloj sažimanja .....	15
3.1.7. Potpuno povezani sloj .....	16
3.2. Metrika u konvolucijskim neuronskim mrežama.....	17
4.IZVEDBA SUSTAVA ZA PREPOZNAVANJE PROMETNIH ZNAKOVA .....	20
4.1. Priprema i obrada slika za kreiranje prilagođene baze podataka .....	20
4.1.1. Alati korišteni za označavanje baze prometnih znakova .....	23
4.1.2. Augmentacija baze slika prometnih znakova.....	24
4.2. Arhitektura RetinaNet modela duboke konvolucijske neuronske mreže.....	33
4.2.1. FPN u RetinaNet modelu .....	34
4.2.2. Klasifikacijska podmreža.....	34
4.2.3. Regresijska podmreža .....	35
4.2.4. Kriterijska funkcija .....	36
4.2.5. Postavljanje parametara za treniranje RetinaNet mreže .....	36
4.3. Programsko rješenje za prepoznavanje prometnih znakova .....	37
4.4. Treniranje RetinaNet konvolucijske neuronske mreže za lokalizaciju i klasifikaciju prometnih znakova u slici .....	38

5.EVALUACIJA KONVOLUCIJSKIH MREŽA KORIŠTENIH ZA PREPOZNAVANJE PROMETNIH ZNAKOVA.....	47
5.1. Testna baza slika .....	47
5.2. Analiza rezultata testiranja RetinaNet konvolucijske neuronske mreže .....	49
6.ZAKLJUČAK .....	55
LITERATURA.....	56
SAŽETAK.....	61
ABSTRACT.....	62
ŽIVOTOPIS .....	63
PRILOZI.....	64

# 1. UVOD

Sustavi za pomoć vozaču (engl. *Advanced driver assistance systems*, ADAS) predstavljaju neizostavan faktor standardne opreme suvremenih vozila te se smatraju izrazito funkcionalnom pomoći pri vožnji, a primarna svrha istih je omogućiti maksimalnu sigurnost sudionika u prometu skretanjem pozornosti vozača na moguće opasnosti u prometu. Uz to, prema [1], unutar samog ADAS sustava može se pronaći mnogo različitih značajki, pri čemu se ističu: sustav za raspoznavanje prometnih znakova, pomoć pri parkiranju, sustav za nadzor budnosti vozača, praćenje horizontalne signalizacije na kolniku te sustav za automatsku prilagodbu brzine vozila. Iako su algoritmi spomenutih sustava za pomoć vozaču danas već u svakodnevnoj uporabi, treba uzeti u obzir kako neki od istih ne postižu dovoljno veliku razinu preciznosti te ih u trenutnom stanju nije moguće integrirati u potpuno autonomna vozila. Tako na primjer, prosječna preciznost jedne od najpoznatijih konvolucijskih neuronskih mreža, YOLOv3, koja se, između ostalog, istražuje i za primjenu u području prepoznavanja prometnih znakova, iznosi približno 92 %, kao što je prikazano u [2]. Sukladno navedenom, prema [3], provode se mnoga istraživanja, naročito u svrhu unaprjeđenja rada sustava za prepoznavanje prometnih znakova jer je isti glavna okosnica na koju se drugi podsustavi, primjerice sustav za regulaciju brzine vožnje, moraju osloniti.

Konvencionalan postupak prepoznavanja prometnih znakova vrši se u nekoliko koraka, koji, prema [4], podrazumijeva postupke obrade slika, detekcije te klasifikacije. Pritom, detekcija objekata vrši se u odnosu na oblik i/ili boju istog, a klasifikacija može uključivati neke od tehnika prepoznavanja znakova, poput umjetnih neuronskih mreža (engl. *Artificial Neural Network*, ANN), metode potpornih vektora (engl. *Support Vector Machine*, SVM) te metode određivanja  $k$  najbližih susjeda (engl. *k-Nearest Neighbour*, KNN). Uzimajući u obzir spomenuto, prema [5], pojam detekcije prometnih znakova može se definirati kao sinergija određivanja položaja određenih objekata unutar slike, odnosno lokalizacije te identificiranja lokaliziranih objekata u odnosu na vrste promatranih objekata (takozvanih klasa), odnosno klasifikacije. Na tragu navedenoga, prema [6, str. 197-198], odvojeno se definiraju izrazi „lokalizacija i klasifikacija“, koji se često koriste kada u slici postoji jedna promatrana klasa objekta za prepoznavanje, te „detekcija objekata“, koja se koristi u slučaju da se u slici nalazi više različitih promatranih klasa objekata za prepoznavanje.

Uzimajući u obzir benefite konvolucijskih neuronskih mreža, kao i potrebu za istraživanjem i doradom postojećih načina prepoznavanja prometnih znakova, ne čudi što je prepoznavanje prometnih znakova jedna od najpopularnijih tema u svijetu modernih vozila. Na

tragu navedenoga, tema ovog rada je evaluacija preciznosti rada konvolucijskih neuronskih mreža koje na standardnim skupovima slika za utvrđivanje uspješnosti algoritma (npr. COCO) postižu konkurentne rezultate u odnosu na trenutno najčešće korištene mreže u području prepoznavanja prometnih znakova. Pritom, navedeno se odnosi na konvolucijske neuronske mreže koje, iako rješavaju brojne probleme vozača, ipak, nisu dovoljno sigurne za primjenu u potpuno autonomnim vozilima. Kako bi se uspješnost metode korištene u ovom radu kvalitetnije verificirala, korištena je prilagođena baza slika prometnih znakova pretežno karakterističnih za europske prometnice te RetinaNet konvolucijska neuronska mreža. Osim toga, kako bi se stvorilo vlastito programsko okruženje unutar kojeg je moguće koristiti prikladne verzije Tensorflow i *Keras* alata, korištena je Anaconda 3 platforma. Štoviše, upotrebom spomenute platforme, omogućena je jednostavnija implementacija i pokretanje *python* skripta koje koriste gore spomenute Tensorflow i *Keras* alate za treniranje i testiranje neuronskih mreža na grafičkim karticama, čime se uvelike smanjuje vrijeme potrebno za obradu podataka. Naime, navedeni alati iznimno su pogodni za rad s algoritmima koji se koriste za strojno i duboko učenje jer sadrže funkcije koje olakšavaju implementaciju neuronskih mreža. Prije implementacije neuronske mreže, potrebno je pripremiti i prilagoditi bazu slika. Obzirom na raznoliku zastupljenost pojedinih prometnih znakova u prometu, prikupljen je nejednak broj slika različitih klasa prometnih znakova. Kako bi se izbalansirao broj slika po klasama, nad postojećim slikama u bazi izvršene su preinake poput primjene rotacije, promjene svjetline, izoštravanja slike itd. Spomenuti proces promjena postojećih slika u svrhu proširenja baze podataka naziva se augmentacija. U ovom radu korištena je *ImgAug* biblioteka za automatsku augmentaciju slika, kojom je, između ostalog, omogućena automatska prilagodba graničnih okvira prometnih znakova u slikama. Obzirom da će se ovaj rad baviti problematikom prepoznavanja prometnih znakova, koristit će se RetinaNet konvolucijska neuronska mreža za lokalizaciju i klasifikaciju prometnih znakova unutar slika. Pritom, prilikom analize rada RetinaNet mreže, stavit će se naglasak na uspješnost klasifikacije prometnih znakova.

Rad je strukturno podijeljen na pet dijelova. Drugo poglavlje daje pregled znanstvenih radova u području prepoznavanja prometnih znakova korištenjem konvolucijskih neuronskih mreža. Nadalje, u trećem poglavlju predstavljena je teorijska podloga konvolucijskih neuronskih mreža, a u četvrtom poglavlju detaljno su opisane etape razvijanja praktičnog dijela rada. Osim toga, u petom poglavlju navedeni su i razjašnjeni postignuti rezultati testiranja rada upotrebljavane konvolucijske neuronske mreže. Naposljetku, šesto poglavlje bavi se glavnim zaključcima donesenim uslijed korištenja konvolucijskih neuronskih mreža za rješavanje problema lokalizacije i klasifikacije prometnih znakova.

## 2. PRIMJENA STROJNOG I DUBOKOG UČENJA ZA PREPOZNAVANJE PROMETNIH ZNAKOVA

U ovom poglavlju razmotrit će se i поближе opisati metode i modeli korišteni prilikom detekcije i klasifikacije prometnih znakova. Radovi navedeni u ovom poglavlju usmjereni su na istraživanja metoda klasifikacije i/ili detekcije prometnih znakova te primjenu istih u autonomnim vozilima. Pritom, u nekim radovima fokus je stavljen na razvoj novog modela mreže. Nasuprot tome, neki radovi bave se unaprjeđenjem već postojećih modela mreže, a u nekim radovima stavljen je naglasak isključivo na primjenu modela konvolucijskih neuronskih mreža za prepoznavanje znakova u ugradbenim sustavima. Međutim, zajedničko svim radovima je implementacija modela koji točno i brzo može lokalizirati i/ili klasificirati prometne znakove u slici.

Autori u [7] rješavaju problem klasifikacije prometnih znakova koristeći europsku bazu podataka, koja se, u navedenom slučaju, sastoji od slika prometnih znakova iz sljedećih europskih država: Belgije, Hrvatske, Francuske, Njemačke, Nizozemske i Švedske. Pritom, za rješavanje problema klasifikacije korišteno je sljedećih nekoliko konvolucijskih neuronskih mreža: LeNet-5, IDSIA, URV, CNN asymmetric i autorska konvolucijska mreža s osam slojeva. Testiranja su izvođena na NVIDIA GeForce GTX1080Ti grafičkoj kartici s 11 GB memorije. Rezultati dobiveni treniranjem poboljšani su pomoću tehnika predobrade slika i augmentacije te su, pritom, ostvarili preciznost prepoznavanja prometnih znakova u iznosu od 99,37 % za vlastitu prilagođenu europsku bazu slika i 98,99 % za GTSRB (engl. *German Traffic Sign Recognition Benchmark*) bazu slika na konvolucijskoj neuronskoj mreži s osam slojeva. Unatoč uspješnosti algoritama, autori su zaključili kako je do najviše pogrešaka došlo uslijed prepoznavanja prometnih znakova obavijesti te prometnih znakova koji sadrže tekst. Navedene pogreške autori pripisuju problematici prepoznavanja teksta te činjenici da određeni prometni znakovi variraju izgledom u različitim državama. Nasuprot tome, u ovom radu se, uz postupak klasifikacije, vrši i postupak lokalizacije kako bi se testirala uspješnost primjene RetinaNet mreže na realnim slikama prometa. Usprkos primjeni različitih modela mreža, ovim radom potvrđuje se zaključak autora o problematici prepoznavanja prometnih znakova koje variraju izgledom u pojedinim državama. Na tragu, toga, dovodi se u pitanje mogućnost stvaranja univerzalnog sustava za prepoznavanje prometnih znakova širom svijeta.

Autori u [8] predstavili su sustav za prepoznavanje prometnih znakova korištenjem YOLO konvolucijske neuronske mreže i mobilne grafičke kartice. Pritom, korištena je NVIDIA Jetson

TK1 platforma te GTSRB baza slika za klasifikaciju prometnih znakova, kao i prilagođena baza slika koju su autori samostalno izradili. U dijelu rasprave rezultata navedene metode autori su se fokusirali na mjerenje potrebnog vremena za prepoznavanje prometnih znakova na testnim slikama te su priložili nekoliko slika s rezultatima detekcije i prepoznavanja prometnih znakova. Iako su svi priloženi prometni znakovi bili ispravno lokalizirani, postoje primjeri koji nisu bili ispravno klasificirani, pri čemu se ističe prometni znak koji označava kružni tok prometa. Govoreći o metrici, dobiveni rezultati na NVIDIA platformi su znatno lošiji u usporedbi sa korištenjem grafičke kartice u računalima. Na tragu spomenutoga, vrijeme potrebno za obradu jedne slike na NVIDIA Jetson TK1 platformi iznosi 1,7 sekundi, a na GTX950 grafičkoj kartici 0,04200 sekunde. Ipak, potrebno je istaknuti važnost ovog rada jer su autori uspješno implementirali složenu konvolucijsku mrežu na najjeftinijem Jetson modelu platforme. Naime, ovim radom pokazano je kako se sustav za prepoznavanje prometnih znakova može uspješno implementirati na mobilnu platformu te kako se korištenjem mobilnih platformi s nešto više resursa, kao na primjer Jetson model treće generacije, može postići rad spomenutog sustava u realnom vremenu.

Autori u [9], također, opisuju uporabu konvolucijskih neuronskih mreža u svrhu klasifikacije prometnih znakova na primjeru GTSRB baze slika. Pritom, slično kao i u ovom radu, naglašava se važnost pripreme skupa za treniranje te se vrši augmentacija slika kako bi se stvorio veći skup slika za učenje, ali i ujednačio broj rasprostranjenosti uzoraka prometnih znakova po klasama baze. Arhitektura navedene konvolucijske neuronske mreže podrazumijeva prosljeđivanje izlaza iz pojedinog sloja samom klasifikatoru, za razliku od tipičnih unaprijednih neuronskih mreža u kojima se izlazni podatak nekog sloja prosljeđuje sljedećem sloju. Osim toga, slično kao i u ovom radu, korištene su sljedeće tehnike regularizacije: isključivanje čvorova (engl. *dropout*), L2 gubitak (engl. *L2 loss*) i rano zaustavljanje treniranja mreže (engl. *early stopping*) praćenjem gubitka unakrsne entropije (engl. *cross entropy loss*) validacijskog skupa slika. Prilikom treniranja konvolucijske neuronske mreže korištene su dvije baze slika, proširena i balansirana. Iste su kreirane augmentacijom svih postojećih slika prometnih znakova, međutim, kod proširene baze slike vršena je augmentacija svih slika originalnog skupa slika, a kod balansirane baze, augmentacija je vršena u odnosu na pojedine klase kako bi se stvorio podjednak broj slika u svakoj klasi. Treniranje je izvršeno u dva stadija, pri čemu se u prvom stadiju koristi proširena baza slika i stopa učenja (engl. *learning rate*) u iznosu 0,001, a u drugom stadiju, tzv. finom podešavanju (engl. *fine tuning*), koristi se balansirana baza slika te stopa učenja u iznosu od



0,0001. Korištenjem spomenute metode postignuta je točnost prepoznavanja prometnih znakova na testnom skupu slika u iznosu od 99,33 %.

Primjer primjene dubokih neuronskih mreža za klasifikaciju prometnih znakova opisan je i u [10]. Pritom, primarni fokus autora je na izradi modela primjerenog za implementaciju u mobilne ugradbene sustave. Spomenuti model izveden je pomoću dvije konvolucijske neuronske mreže. Na tragu navedenoga, formira se jedna složenija mreža, tzv. učiteljska mreža te druga, jednostavnija mreža, tzv. učenička mreža. Također, prilikom treniranja navedenih mreža vrši se preneseno učenje, odnosno izlazne vrijednosti iz učiteljske mreže koriste se kao početne vrijednosti ulaznih veličina u učeničku mrežu. Spomenuta tehnika prijenosa znanja s veće na manju neuronsku mrežu predstavlja glavno obilježje primjene tehnike destilacije znanja (engl. *knowledge distillation*). Na tragu spomenutoga, učiteljska mreža sastoji se od više konvolucijskih slojeva, između kojih je omogućeno paralelno dijeljenje aktivacijskih mapa, što podrazumijeva prosljeđivanje aktivacijskih mapa svim slojevima ispred određenog konvolucijskog sloja, a ne samo narednom konvolucijskom sloju. Osim toga, potrebno je napomenuti kako se učenička mreža sastoji od pet konvolucijskih slojeva te jednog potpuno povezanog sloja. Navedeni model treniran je i testiran na GTSRB i BTSC (engl. *Belgian Traffic Sign Classification*) bazama slika prometnih znakova, pri čemu treba uzeti u obzir kako je učiteljska mreža prethodno trenirana i na CIFAR-10 bazi podataka te je dobivena prosječna točnost prepoznavanja prometnih znakova učiteljske mreže u iznosu od 99,23 %, dok točnost prepoznavanja prometnih znakova učeničke konvolucijske neuronske mreže iznosi 99,61 %.

Autori u [11] predlažu korištenje R-CNN konvolucijskih neuronskih mreža za detekciju i prepoznavanje prometnih znakova. Pritom, izrađena je vlastita baza podataka DFG koja sadrži 200 klasa prometnih znakova, a za evaluaciju rješenja uspoređeni su rezultati dobiveni prilagođenim modelom Mask R-CNN, uz augmentaciju podataka, sa standardnom Mask R-CNN mrežom i Faster R-CNN mrežom. Osim DFG baze podataka, za evaluaciju modela predstavljenog u navedenom radu korištena je i švedska baza slika prometnih znakova. Kako bi se Mask R-CNN prilagodila za detekciju i prepoznavanje prometnih znakova, autori su koristili sljedeće tehnike:

- OHEM (engl. *Online Hard Example Mining*), koji je implementiran u klasifikacijski dio korištenog modela mreže kako bi se osiguralo učenje modela na uzorcima prometnih znakova koji se najteže prepoznaju, što se mjeri pomoću iznosa kriterijske funkcije klasifikacije (engl. *classification loss*)

- distribucija odabranih uzoraka za trening, kako bi se ujednačio broj uzoraka malih i velikih objekata (prometnih znakova) u slikama
- postavljanje težina u odnosu na uzorke, čime se veće težine postavljaju na objekte koji se nalaze prednjem planu u odnosu na pozadinu
- prilagođavanje broja regija od interesa koje se prosljeđuju klasifikacijskoj mreži tijekom detekcije objekata.

Naposljetku, potrebno je istaknuti kako su korištenjem prethodno objašnjenog prilagođenog modela mreže postignuti sljedeći rezultati: 95,5 % točnosti za prilagođenu Mask R-CNN mrežu s ResNet-50 okosnicom (engl. *backbone*) te 95,2 % točnosti za prilagođenu Mask R-CNN mrežu s ResNet-101 okosnicom.

Nadalje, u [12] predstavljen je poboljšani model Faster R-CNN mreže za detekciju prometnih znakova. Pritom, korištene su njemačka i kineska baza slika, koje su spojene u jednu bazu kako bi se povećao broj uzoraka prometnih znakova koje je potrebno detektirati. Kako bi se smanjilo vrijeme procesuiranja slika korištenjem spomenute mreže, autori uvode korištenje dodatne mreže koja predlaže najvjerojatnija područja od interesa unutar slike te istu koriste kao primaran izvor informacija. Štoviše, navedenim postupkom uspješno je reduciran broj graničnih okvira (engl. *anchor*) koji nastaju uslijed primjene originalne Faster R-CNN mreže. Nadalje, korištena je metoda kombiniranja značajki trećeg, četvrtog i petog sloja VGG16 mreže, čime se pospješuje mogućnost detekcije manjih objekata u slici. Također, korišteni su rezultati sekundarnih područja od interesa prilikom postupka klasifikacije znaka kako bi spomenuti postupak klasifikacije bio točniji. Govoreći o postignutim rezultatima, točnost detekcije prometnih znakova navedene metode iznosi 99,01 %, a moguće je procesuirati prosječno 9,3 okvira u sekundi, što zadovoljava zahtjeve izvođenja predikcije u stvarnom vremenu.

Prepoznavanje prometnih znakova unutar slike snimanih sa većih udaljenosti, također, je jedan od velikih izazova prilikom istraživanja primjene računalnog vida u autonomnim vozilima. Ipak, riječ je o iznimno bitnoj odrednici prilikom govora o kvaliteti i funkcionalnosti sustava za prepoznavanje prometnih znakova jer se navedenim osigurava identifikacija prometnih znakova sa većih udaljenosti. Na tragu navedenoga, u [13], opisana je uporaba MR-CNN mreže za prepoznavanje udaljenih prometnih znakova. Treniranje i testiranje mreže izvedeno je na bazi slika prometnih znakova Tsinghua-Tencent 100K, a implementacija modela temelji se na kombiniranju značajki konvolucijskih slojeva i informacijama o kontekstu unutar slike. Prilikom detekcije dolazi do formiranja prijedloga o području od interesa na temelju fuzije aktivacijskih mapi. Spomenuta

fuzija izrađena je pomoću operacija dekonvolucije, normalizacije i kompresije. Nadalje, prilikom klasifikacije kreira se fuzija značajki za potpuno povezani sloj te se vrši određivanje regija konteksta unutar slike. Navedena metoda pokazala se uspješnom za prepoznavanje prometnih znakova koji variraju veličinom unutar same slike (obzirom na udaljenost snimanja) te je izmjerena vrijednost F1 mjere za male prometne znakove iznosila 86 %, za srednje velike prometne znakove 93,5 %, a za velike prometne znakove 90,1 %.

Osim navedenog, zanimljiv pristup prepoznavanju znakova predstavljen je u [14], u kojem se kombiniraju Viola-Jones algoritam i duboko učenje. Za potrebe navedenog istraživanja, korištena su tri rješenja za prepoznavanje prometnih znakova. Ponajprije, implementiran je Viola-Jones algoritam za prepoznavanje prometnih znakova. Nadalje, konstruirana je konvolucijska neuronska mreža kojom se vrši detekcija i klasifikacija prometnih znakova u slici. Konačno, treće rješenje bavi se implementacijom rješenja u kojem se međusobno integriraju prethodno spomenuti modeli. Pritom, cilj je iskoristiti glavne prednosti svakog modela ponaosob, stoga je izrađen model koji kombinira izvrsnu sposobnost uspješne detekcije prometnog znaka kod Viola-Jones algoritma sa sposobnošću konvolucijske neuronske mreže da s visokom točnošću klasificira detektirane objekte. Konkretno, navedeno je ostvareno korištenjem kaskada, koje predstavljaju glavno obilježje Viola-Jones algoritma, kako bi se izdvojila područja od interesa unutar kojih se nalaze prometni znakovi, a koja se, zatim, predaju konvolucijskoj neuronskoj mreži na provođenje klasifikacije nad određenim znakom. Navedeni modeli validirani su na njemačkoj i indijskoj bazi slika te su postignuti sljedeći rezultati: 90 % točnosti prepoznavanja znakova za njemačku i 92 % za indijsku bazu slika, uz iznos F1 mjere 88,75 % za njemačku bazu i 92,34 % za indijsku bazu podataka.

Prilikom kreiranja modela dubokog učenja, jedna od ključnih stavki na koju treba obratiti pozornost je brzina primijenjenog algoritma. Sukladno navedenom, ističu se rezultati postignuti u [15], u kojem je predložen model detekcije prometnih znakova pogodan za rad u stvarnom vremenu te primjenu u ugradbenim sustavima. Spomenuti model testiran je na prilagođenoj bazi podataka nastaloj prikupljanjem slika prometnih znakova u Kini te sadrži 10 000 slika raspodijeljenih na 73 klase. Naime, riječ je o uporabi MobileNet konvolucijske neuronske mreže za izdvajanje značajki unutar slike te klasifikaciju detektiranih objekata i dodatne konvolucijske neuronske mreže koja ima dva izlaza. Pritom, prvi izlaz predstavlja rezultat djelovanja binarnog klasifikatora, pomoću kojeg se određuje postoji li znak u slici, dok drugi izlaz predstavlja rezultat regresije kojom se utvrđuju koordinate graničnog okvira (engl. *bounding box*). Navedena konvolucijska neuronska mreža sastoji se od tri konvolucijska sloja, *softmax* sloja te sloja linearne

regresije. Testiranjem prethodno objašnjenog modela dobiveni su sljedeći rezultati: točnost prepoznavanja prometnih znakova iznosi 84,22 % s preciznošću preko 80 % po objektu, a brzina inferencije iznosi 0,07 sekundi po slici, što nalaže da navedeni model može obraditi 15 okvira u sekundi. Također, važno je napomenuti kako primijenjeni model zauzima 3,38 puta manje memorije u odnosu na YOLOv3 konvolucijsku neuronsku mrežu, što, uzevši u obzir spomenutu brzinu obrade podataka, istog čini pogodnim za primjenu u ugradbenim računalnim sustavima.

Autori u [16] predlažu sljedeću metodu za efektivnu detekciju i klasifikaciju slika: korištenje HSV sustava boja za segmentaciju slike i filtriranje označenih područja od interesa unutar slike. Pritom, pomoću pragova odgovarajućih boja prometnih znakova izdvaja se područje od interesa. Nakon toga, primjenjuje se filtriranje slike u odnosu na specifične konture uslijed čega dolazi do izbacivanja potencijalnih područja koja su segmentirana, a koja ne sadrže prometni znak. Nakon spomenutog postupka, vrši se detekcija po obliku. Zatim, koristi se poboljšana verzija LeNet-5 konvolucijske neuronske mreže za klasifikaciju detektiranih znakova koja se sastoji od dva konvolucijska sloja, dva sloja sažimanja, dva potpuno povezana sloja te izlaznog sloja. Kako bi poboljšali učinkovitost klasične LeNet-5 mreže, autori koriste Gabor jezgru (engl. *kernel*) kao početnu jezgru konvolucije te uvode BN (engl. *batch normalization*) normalizaciju podataka, kojom se standardiziraju ulazi grupa podataka (engl. *batch*) u svaki sloj mreže. Navedeno se izvodi korištenjem srednje vrijednosti odstupanja i standardne devijacije svih vrijednosti u promatranoj grupi podataka. Osim toga, korištena je ReLU aktivacijska funkcija, kao i Adam metoda optimizacije, koja potpomaže rješavanju problema spore konvergencije te velike varijabilnosti iznosa kriterijske funkcije. Navedeni model testiran je na GTSRB njemačkoj bazi slika. Kako bi se poboljšao postupak treniranja mreže, izvršena je augmentacija baze slika, kojom se broj uzoraka unutar svake, od 43 klase prometnih znakova, izbalansirao. Opisanim postupkom detekcije i prepoznavanja znakova dobiveni su sljedeći rezultati: preciznost prepoznavanja prometnih znakova utvrđena na testnom skupu slika iznosi 99,75 %, a prosječno vrijeme procesuiranja jednog okvira iznosi 5,4 milisekunde.

### 3. PREPOZNAVANJE OBJEKATA KORIŠTENJEM KONVOLUCIJSKIH NEURONSKIH MREŽA

Konvolucijske neuronske mreže (engl. *Convolutional Neural Networks*, CNN) primarno su bile upotrebljavane za prepoznavanje rukom pisanih znamenki 80-ih godina prošlog stoljeća. Prema [17], iste su razvijene kako bi neuroni u nižim slojevima mreže izdvajali lokalne značajke iz slike, koje bi, zatim, neuroni u višim slojevima mreže kombinirali te na taj način stvarali složenije značajke višeg reda.

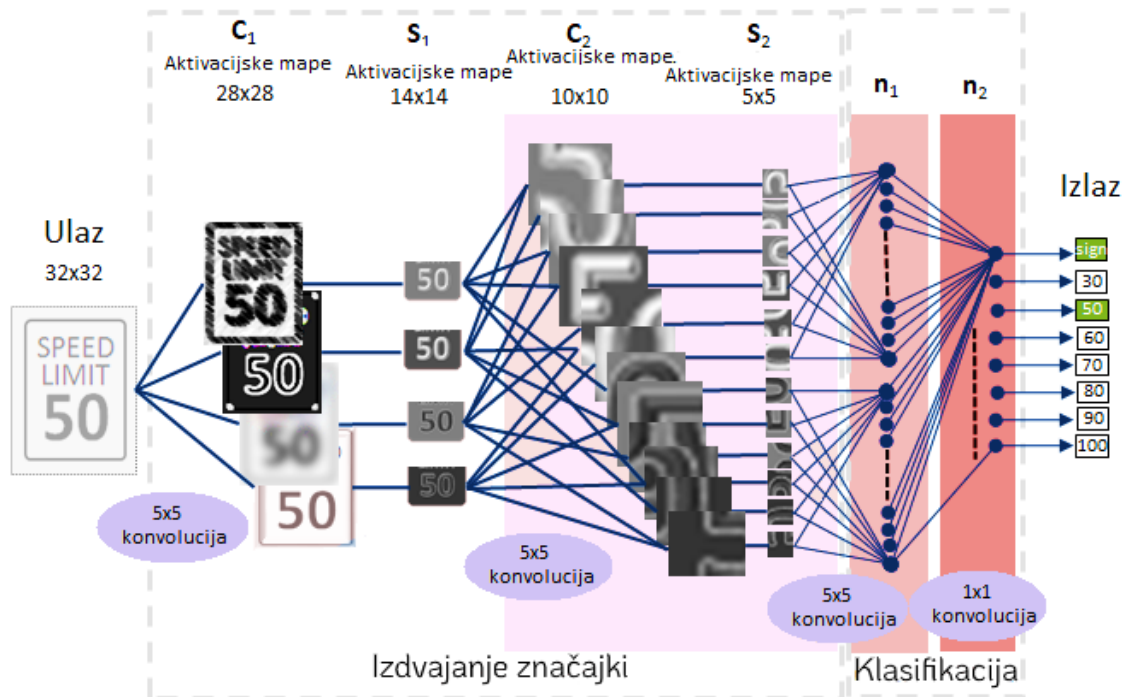
Pojam lokalnih značajki odnosi se na ograničen manji skup susjednih elemenata slike, kao na primjer dijelove krivulja ili linija. U slučaju prepoznavanja znakova, složenije značajke koje bi se dobile pomoću jednostavnijih značajki, odnosile bi se na značajke koje predstavljaju određeni oblik prometnog znaka (npr. karakteristični oblici trokuta ili šesterokuta). Pritom, neuroni u završnim slojevima konvolucijske mreže, pak, koriste opisane složenije značajke kako bi se mogla izvršiti klasifikacija samog prometnog znaka.

Tako se, kao što je navedeno u [17], zaključuje da srž neuronskih mreža za prepoznavanje objekata u slikama počiva na pronalasku lokalnih značajki karakterističnih za specifični objekt. Pri čemu se na temelju prisutnosti, odnosno odsutnosti, lokalnih značajki određuje sadrži li slika specifični objekt. Na tragu navedenoga, sljedeća potpoglavlja opisivat će neizostavne sastavnice konvolucijskih neuronskih mreža, kojima će se pobliže objasniti prethodno spomenuti postupak izdvajanja značajki.

#### 3.1. Arhitektura konvolucijskih neuronskih mreža

Kao jedna od najvećih prepreka za postizanje kvalitetnih i preciznih rezultata prilikom korištenja konvolucijskih neuronskih mreža, navodi se dizajn arhitekture istih. Pritom se očekuje da spomenute mreže mogu identificirati određeni objekt bez obzira u kojem se dijelu slike objekt nalazi. Kako bi se pobliže objasnilo rješenje navedenog problema, prikazat će se i objasniti općenita struktura konvolucijskih neuronskih mreža.

Strukturu konvolucijskih neuronskih mreža čine četiri tipa sloja, a to su: konvolucijski sloj (engl. *convolutional layer*), nelinearni sloj (engl. *nonlinearity layer*), sloj sažimanja (engl. *pooling layer*) te potpuno povezani sloj (engl. *fully connected layer*), kako je i opisano u [18]. (Sl. 3.1.)

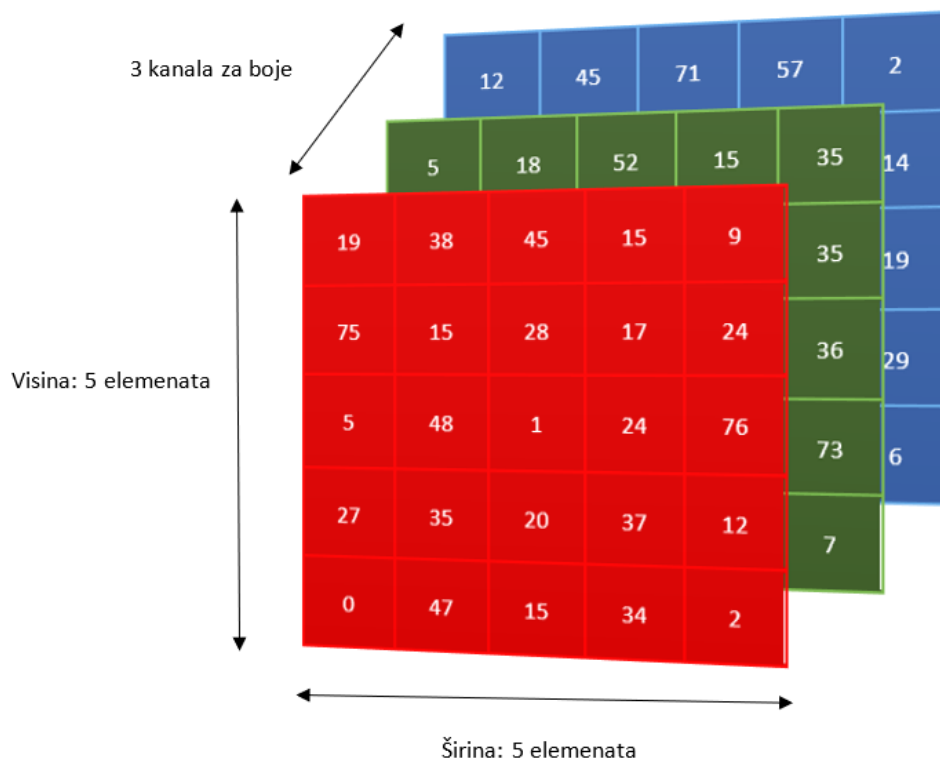


**Slika 3.1.** *Primjer strukture i primjene konvolucijske neuronske mreže nad prometnim znakom ograničenja brzine 50 mph , iz [19].*

### 3.1.1. Ulazni sloj

Prema slici 3.1. moguće je uvidjeti kako konvolucijska neuronska mreža, uz prethodno navedene slojeve, sadrži i ulazni sloj u kojem se učitava slika u obliku trodimenzionalnog niza podataka koji sadržava vrijednosti elemenata slike.

Pritom, svaka dimenzija niza odnosi se na određeni kanal boje (crveni, zeleni ili plavi, engl. *red, green, blue*) RGB slike, kako je prikazano u [20]. Primjer sadržaja spomenutog trodimenzionalnog niza s navedenim vrijednostima elemenata slike za svaki kanal boje prikazan je na slici 3.2. Također, zbog jednostavnijeg prikaza u primjeru na slici 3.2. nalazi se prikaz elemenata slike veličine 5x5 elemenata (engl. *pixels*).



**Slika 3.2.** *Primjer RGB slike veličine 5x5 s prikazom vrijednosti elemenata slike razdijeljenih po kanalima boja.*

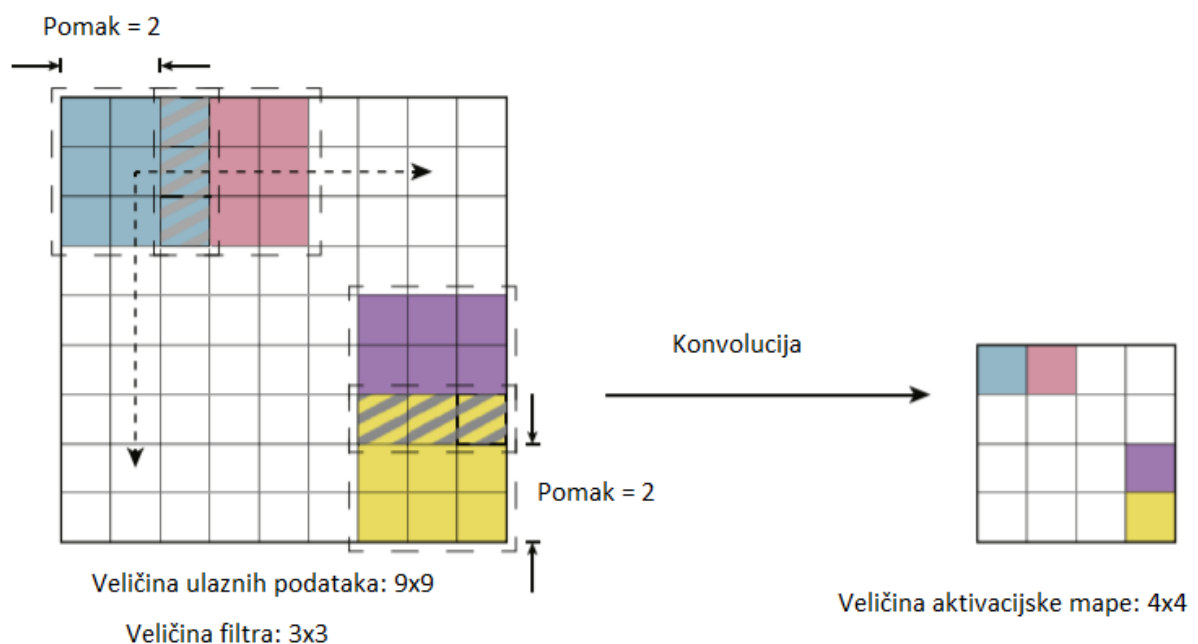
### 3.1.2. Konvolucijski sloj

Konvolucijski sloj je najvažniji sloj konvolucijskih neuronskih mreža, a prema [18], u istom se obavlja najveći dio zahtjevnih računalnih radnji cjelokupne mreže. Također, potrebno je napomenuti kako je glavna zadaća navedenog sloja izvođenje konvolucije nad elementima slike. Uz navedeno, treba napomenuti kako je operacija konvolucije razlog zbog kojeg je konvolucijska neuronska mreža invarijantna na prostor unutar slike, tj. ista može detektirati objekt neovisno o tome gdje se točno u slici određeni objekt nalazi.

Naime, prema [6], unutar konvolucijskih slojeva nalazi se skup filtera, koji se u postupku konvolucije postavljaju u lijevi gornji kut slike te se vrijednosti spomenutih filtera množe s pripadajućim elementima na slici, a potom se navedeni umnošci zbroje. Štoviše, navedeni postupak vrši se u slici na svim elementima tako što se filteri, nakon svake izvršene operacije konvolucije pomiču za prethodno određeni broj elemenata udesno. Kada se dođe do posljednjeg desnog elementa slike, postupak se nastavlja pomicanjem filtera na skup krajnjih lijevih elemenata, koji su za prethodno određeni broj elemenata pomaknuti vertikalno prema dolje. Rezultati svih

operacija konvolucija pohranjuju se u, tzv. aktivacijsku mapu (engl. *feature map*), Navedena operacija konvolucije prikazana je slikom 3.3. Osim toga, vrijednosti koje se nalaze unutar filtera čine niz parametara, tzv. težina (engl. *weights*) koje se tijekom procesa učenja konvolucijske neuronske mreže ažuriraju u svakoj iteraciji.

Također, na slici 3.3. može se vidjeti da za ulazni podatak veličine 9x9, uz pomak elemenata za 2, filter veličine 3x3 te primjenu konvolucije, nastaje aktivacijska mapa dimenzija 4x4. Pritom, iznimno je važno da filter ima jednaku dubinu kao i ulaz (slika) na koju se isti primjenjuje. Kako bi se poboljšalo razumijevanje postupka konvolucije i primjene filtera, prethodno navedeni parametri bit će objašnjeni u daljnjem razvoju teksta.



**Slika 3.3.** Konvolucija nad ulaznom matricom 9x9, iz [18].

### 3.1.3. Hiperparametri u konvolucijskom sloju

Kako bi se operacija konvolucije mogla uspješno izvesti, prema [6, str. 81-105] i [18], koriste se sljedeći hiperparametri:

- veličina filtera (engl. *filter size*), koja predstavlja visinu i širinu filtera koji se koristi nad elementima slike (na slici 3.3. veličina filtera iznosi tri)
- pomak (engl. *stride*), koji predstavlja broj koraka (elemenata slike) za koji se filter pomiče u horizontalnom i vertikalnom smjeru po slici



- dopunjavanje (engl. *padding*), kojim se dodaju redovi i stupci oko slike čiji elementi imaju zadanu vrijednost 0, a koriste se kako bi se održala potrebna dužina i širina slike koja se uslijed primjene konvolucije smanjuje
- dubina (engl. *depth*), koja označava broj slojeva koji nastaju uslijed primjene filtera i konvolucije nad slikom.

Navedeni parametri zadaju se za svaki sloj konvolucijske neuronske mreže te mogu biti različiti za svaki konvolucijski sloj.

### 3.1.4. Opća formula za operaciju konvolucije i formiranje aktivacijske mape

Nakon objašnjenja hiperparametara koje je potrebno odrediti za uspješnu provedbu konvolucije, mogu se navesti osnovni izrazi po kojima se ista izvodi, a koje su navedene u [6, str. 85].

$$m_{konačno} = \frac{m_{početno} - m_{filter}}{p} + 1, \quad (3-1)$$

gdje  $m_{konačno}$  označava dimenziju matrice koja se dobiva nakon konvolucije,  $m_{početno}$  predstavlja dimenzije ulazne matrice nad kojom se vrši konvolucija,  $m_{filter}$  označava dimenziju filtera, a  $p$  označava zadani pomak.

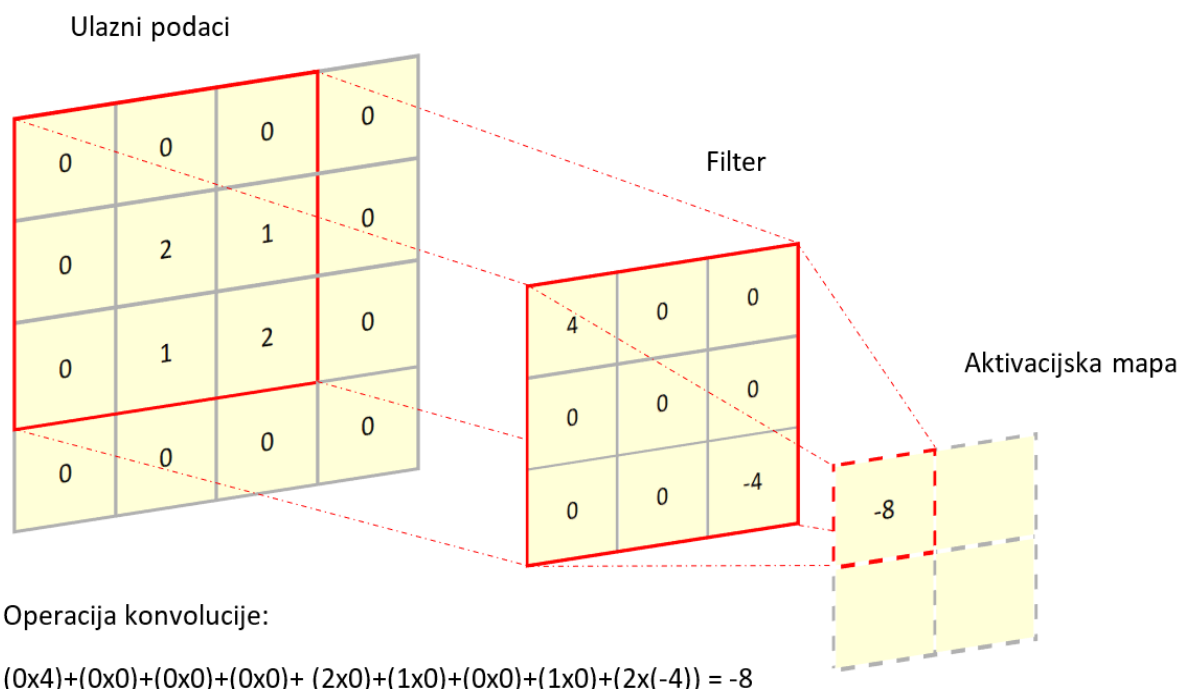
Osim toga, na slici 3.4., prikazan je primjer i postupak računanja vrijednosti jednog elementa aktivacijske mape za pomak po jedan element u matrici i to prema sljedećem izrazu, navedenom u [21]:

ako vrijedi  $U \in \mathbb{R}^{V \times D}$  i  $f \in \mathbb{R}^{D_f \times \check{S}_f}$ , onda je:

$$(U * f)(v, d) = \sum_{i=0}^{V_f-1} \sum_{j=0}^{D_f-1} U(v+i, d+j)(i, j) \quad v = 0, \dots, V-1, d = 0 \dots, D-1, \quad (3-2)$$

gdje  $U$  predstavlja ulaznu sliku,  $f$  filter,  $V$  visinu slike,  $D$  dužinu slike,  $V_f$  visinu filtra, a  $D_f$  dužinu filtra.

Pritom svakako treba naglasiti kako (3-2) vrijedi isključivo za vrijednost pomaka jedan. U slučaju da je pomak veći, primjerice ako se pretpostavi da vrijednost istog iznosi dva, onda vrijedi  $v = 0, 2, 4, \dots, V-1$ ,  $d = 0, 2, 4, \dots, D-1$ . Također, treba napomenuti kako su vrijednosti pomaka rijetko kada veće od dva, kao što je navedeno u [21].



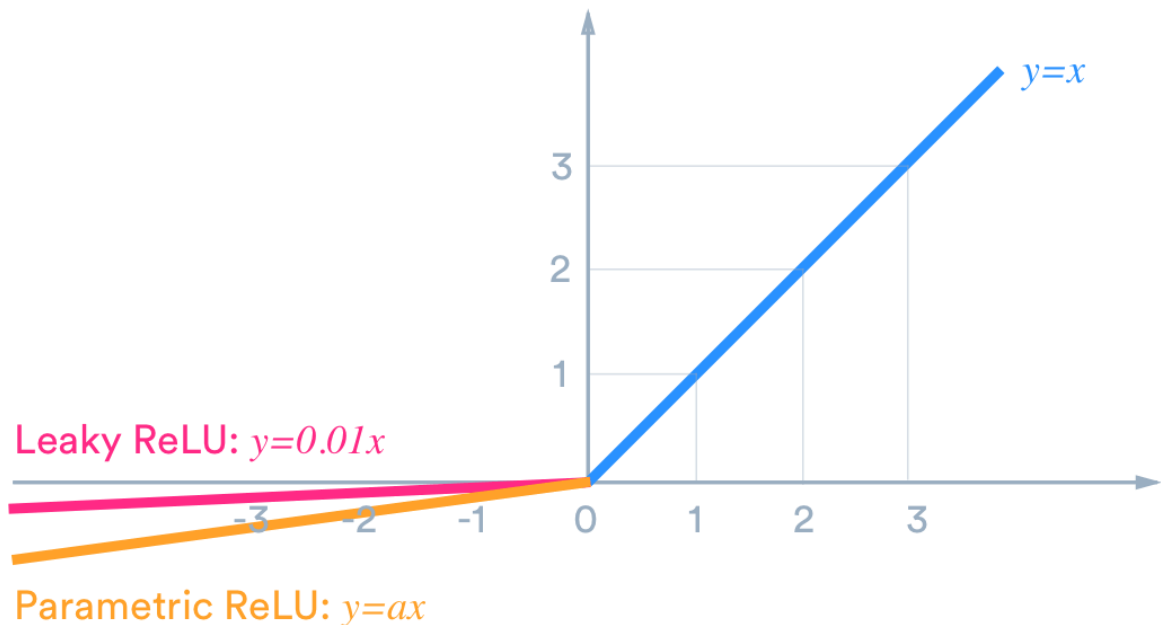
**Slika 3.4.** Prikaz izračuna elementa aktivacijske mape primjenom filtera veličine 3x3 i konvolucije.

### 3.1.5. Nelinearni sloj

Iako se najčešće aktivacijske funkcije poput ReLU, Leaky ReLU te Sigmoid funkcije implementiraju unutar samog konvolucijskog sloja, prema [18], postoje slučajevi kada se iste integriraju u neuronsku mrežu kao zasebni slojevi. Pritom, spomenuti zasebni slojevi nazivaju se nelinearni slojevi. Navedeno se ponekad ugrađuje u strukturu konvolucijske neuronske mreže kako bi se povećala fleksibilnost iste. Glavna svrha aktivacijske funkcije jest propustiti određene vrijednosti u idući sloj mreže. Spomenute vrijednosti moraju po svom iznosu odgovarati rezultatu upotrebljavane aktivacijske funkcije. Obzirom da je ReLU aktivacijska funkcija jedna od najpoznatijih i najupotrebljavanijih u suvremenim istraživanjima dubokog učenja, ista dolazi u nekoliko varijanti, koje su prikazane slikom 3.5.

Naime, ReLU je linearna za sve pozitivne vrijednosti, a iznosi nula za sve negativne vrijednosti. Time su znatno olakšane računalne radnje koje je potrebno izvoditi prilikom obrade neuronskih mreža. Osim toga, omogućena je brža konvergencija te se prorjeđuju neuroni što pospješuje mogućnost aktivacije ispravnog neurona prilikom detekcije značajki objekata. Slično tome Leaky ReLU i Parametric ReLU aktivacijske funkcije su također linearne za sve pozitivne vrijednosti, međutim, negativne vrijednosti Leaky ReLU aktivacijske funkcije računaju se prema

formuli:  $y = 0,01x$ , a negativne vrijednosti Parametric ReLU aktivacijske funkcije prema formuli:  $y = ax$ .

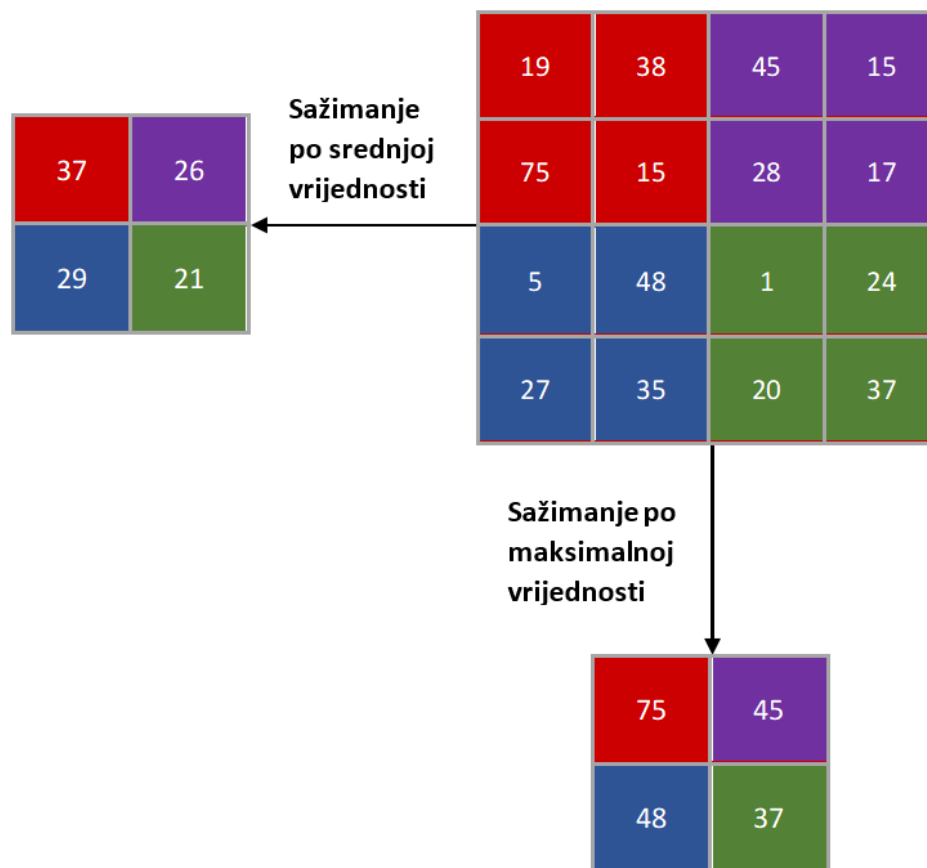


**Slika 3.5.** Prikaz Leaky ReLU, ReLU i Parametric ReLU aktivacijskih funkcija, iz [22].

### 3.1.6. Sloj sažimanja

Sloj sažimanja služi za smanjivanje dimenzija ulaznih podataka u konvolucijskom sloju, kako je i pojašnjeno u [21]. Navedeni postupak često se izvodi kako bi smanjenje dimenzija neposredno uzrokovalo smanjenje računalnih resursa potrebnih za obradu danih podataka. Ipak, potrebno je naglasiti kako se korištenjem spomenute metode ne gube podaci. Štoviše, izdvajaju se podaci koji su najznačajniji u nizu podataka, što, posljedično, omogućava smanjenje šuma u samim podacima.

Također, prema [21], postoji nekoliko vrsta provođenja sažimanja podataka, a najpoznatiji su: sažimanje po maksimalnoj vrijednosti (engl. *Max Pooling*), sažimanje po srednjoj vrijednosti (engl. *Average Pooling*) i stohastičko sažimanje. Kako se naslućuje iz samog naziva, sažimanjem po maksimalnoj vrijednosti izdvaja se maksimalna vrijednost unutar skupa promatranih vrijednosti (engl. *kernel*). Sažimanjem po prosječnoj vrijednosti usrednjavaju se sve vrijednosti promatranog *kernela* te se navedeni rezultat smatra reprezentativnim podatkom promatranog skupa podataka. Grafički prikaz prethodno opisanih metoda sažimanja nalazi se na slici 3.6.



**Slika 3.6.** Grafički prikaz metoda sažimanja.

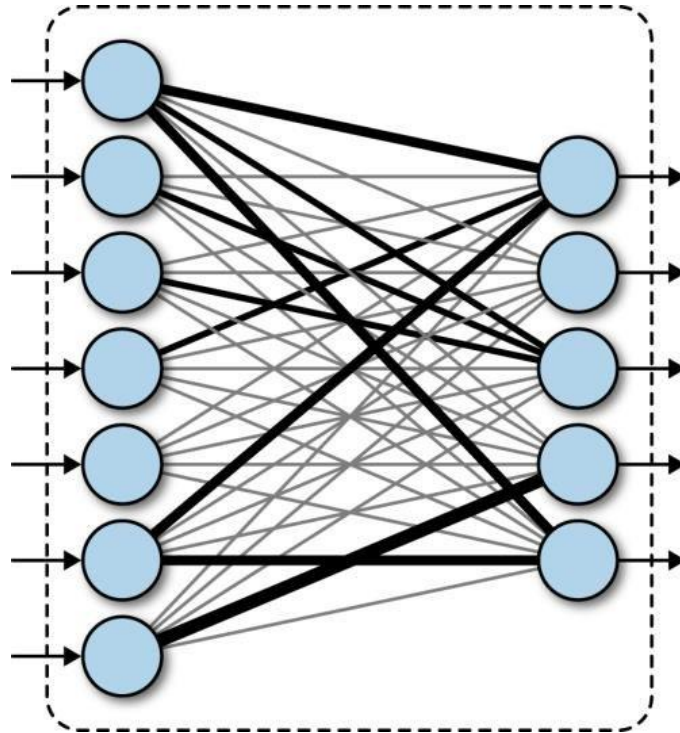
Nadalje, stohastičko sažimanje se, prema opisu u [21], odnosi na nasumičan odabir elementa iz promatranog skupa, ali postoji veća vjerojatnost odabira većih vrijednosti unutar skupa podataka. Obzirom da je eksperimentalno utvrđeno kako je tehnika sažimanja po maksimalnoj vrijednosti najefektivnija, ista se najčešće koristi u praksi prilikom rada s dubokim neuronskim mrežama.

Govoreći o hiperparametrima koji se koriste u sloju sažimanja, prema [23], potrebno je istaknuti veličinu filtera, pomak te dopunjavanje podataka.

### 3.1.7. Potpuno povezani sloj

Potpuno povezani sloj, prema [18], služi za izvedbu klasifikacije nad složenim izdvojenim značajkama nižih slojeva mreže. Svaki neuron navedenog sloja spojen je na svaki neuron narednog sloja, a kako bi se uopće mogla izvesti funkcija potpuno povezanog sloja potrebno je aktivacijske mape pretvoriti u jednodimenzionalni vektor. Također, nakon potpuno povezanog sloja neuronske mreže integrira se izlazni sloj sa *softmax* aktivacijskom funkcijom koja omogućava izvođenje

više-klasne klasifikacije, kako je navedeno u [6, str. 109]. Prikaz potpuno povezanog sloja moguće je vidjeti na slici 3.7.



Slika 3.7. Prikaz potpuno povezanog sloja, iz [24].

### 3.2. Metrika u konvolucijskim neuronskim mrežama

Prilikom uporabe konvolucijskih neuronskih mreža potrebno je obratiti pozornost na mjerenje sljedećih parametara u svrhu testiranja rada korištenog modela konvolucijske neuronske mreže:

- broj točno predviđenih klasa zadanih objekata (engl. *true positive*, TP)
- broj točno predviđenih negativnih klasa zadanih objekata (kada model ispravno predviđa da određeni objekt u slici ne pripada nijednoj od zadanih klasa) (engl. *true negative*, TN)
- broj netočno predviđenih klasa zadanih objekata (engl. *false negative*, FN)
- broj netočno predviđenih negativnih klasa zadanih objekata (kada model svrsta objekte koji ne pripadaju nijednoj od zadanih klasa, u jednu od zadanih klasa) (engl. *false positive*, FP)

Korištenjem prethodno opisanih parametara računaju se statističke veličine pomoću kojih se evaluira podobnost upotrebljavanog algoritma za određeni zadatak, a prema [25] i [26], potrebno je izdvojiti sljedeće:

- preciznost (engl. *precision*)
- odziv (engl. *recall*)
- točnost (engl. *accuracy*).

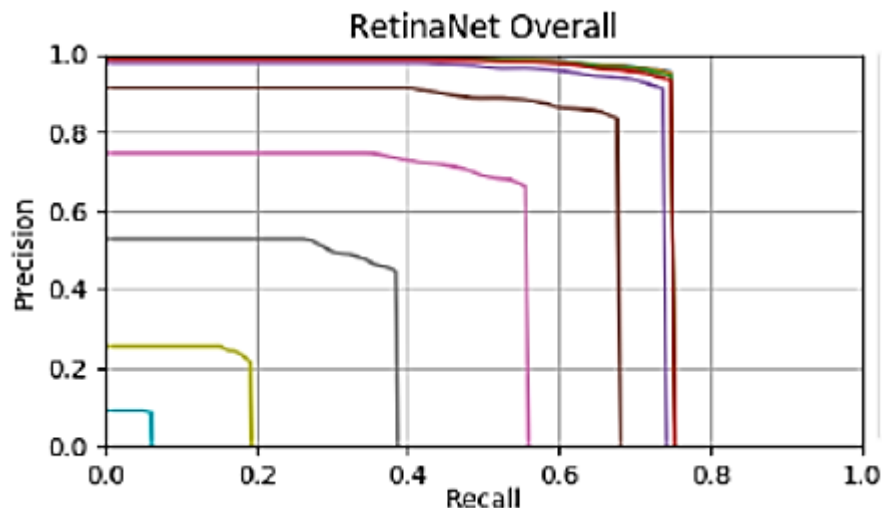
Preciznost se definira kao mjera za točnost korištenog algoritma kojom se iskazuje postotak točno prepoznatih objekata, a računa se prema sljedećem izrazu, kako je navedeno u [25]:

$$\text{preciznost} = \frac{TP}{TP+FP}. \quad (3-3)$$

Nadalje, odzivom se utvrđuje koliko je algoritam uspješan u predviđanju svih objekata koje je potrebno prepoznati, a računa se prema sljedećem izrazu, kako je navedeno u [26]:

$$\text{odziv} = \frac{TP}{TP+FN}. \quad (3-4)$$

Zatim, kako bi se još bolje mogao evaluirati korišteni algoritam, prema [25], preporučljivo je pregledati i krivulju preciznost/odziv, u kojoj je prikazana ovisnost preciznosti modela o odzivu. Primjer navedene krivulje za RetinaNet konvolucijsku neuronsku mrežu dan je na slici 3.8.



**Slika 3.8.** Prikaz krivulje preciznost/odziv, iz [25].

Iz slike 3.8. mogu se izračunati srednje vrijednosti preciznosti rada modela. Prema [25], prosječna preciznost (engl. *average precision*, AP) za određenu klasu  $n$  računa se kao površina ispod krivulje preciznosti, pri čemu se preciznost  $p$  integrira kao funkcija odziva  $r$ . Na tragu navedenoga, dolazi se do sljedećeg izraza za računanje srednje preciznosti:

$$AP = \int_0^1 p(r)dr. \quad (3-5)$$

Daljnijim usrednjavanjem vrijednosti  $AP$  za sve promatrane klase  $N$ , dobiva se vrijednost glavne prosječne preciznosti (engl. *mean average precision*, mAP), koja se računa prema sljedećoj formuli, opisanoj u [25]:

$$mAP = \frac{1}{N} \sum_{n=1}^N AP(n). \quad (3-6)$$

Osim toga, potrebno je napomenuti kako iznos navedenih veličina može poprimiti vrijednost između 0 i 1. Uz to, gore navedene veličine ujedno su najčešće korišteni parametri za evaluaciju algoritama za prepoznavanje objekata jer se smatraju ključnim pokazateljima uspješnosti izvođenja zadatka klasifikacije i/ili detekcije objekata. Također, prilikom korištenja gotovih modela konvolucijskih neuronskih mreža, navedena statistika je najčešće već implementirana u sam algoritam pomoću postojećih biblioteka i aplikacijskih programskih sučelja (engl. *Application Programming Interface*, API) prilagođenih potrebama razvojnih inženjera strojnog i dubokog učenja.

Sljedeće poglavlje neposredno se nadovezuje na teorijsku podlogu vezanu za konvolucijske neuronske mreže te će se u istom predstaviti i opisati baza podataka te model konvolucijske neuronske mreže korišten u svrhu izvršavanja zadatka ovog rada.

## 4. IZVEDBA SUSTAVA ZA PREPOZNAVANJE PROMETNIH ZNAKOVA

Ovo poglavlje posvećeno je opisu izvedbe sustava za prepoznavanje prometnih znakova izrađenog u okviru ovog diplomskog rada. Pritom će biti opisana prilagođena baza prometnih znakova izrađena za potrebe izvedbe zadatka ovog rada, kao i alati koji su korišteni prilikom implementacije modela konvolucijske neuronske mreže. Za obradu podataka korišten je skriptni jezik *python*, u kojem je, između ostalog, napisana struktura konvolucijske mreže korištene u ovom radu. Uz to, korišten je *Keras, python API*, koji omogućuje znatno lakše rukovanje metodama za implementaciju algoritama dubokog učenja.

Osim spomenutoga, u ovom poglavlju bit će opisana RetinaNet, složena konvolucijska neuronska mreža, čiji je model korišten za treniranje i evaluaciju procesa lokalizacije i klasifikacije prometnih znakova.

### 4.1. Priprema i obrada slika za kreiranje prilagođene baze podataka

Kako bi se uopće moglo provesti prepoznavanje prometnih znakova korištenjem konvolucijskih neuronskih mreža, potrebno je najprije obraditi odgovarajuće slike i stvoriti bazu podataka. Cilj stvaranja kvalitetne baze slika je omogućiti dobru generalizaciju modela tijekom treniranja uslijed čega se očekuje uspješnija validacija i testiranje istog u pogledu rezultata glavne prosječne preciznosti.

Obzirom da broj vrsta prometnih znakova seže do nekoliko stotina, odabrane su 33 najčešće, nad kojima je potrebno trenirati, validirati i testirati konvolucijske neuronske mreže za klasifikaciju i lokalizaciju prometnih znakova. Pritom, riječ je o prometnim znakovima navedenima u tablicama 4.1. i 4.2.

**Tablica 4.1.** *Popis prometnih znakova za klasifikaciju.*

Naziv znaka	
ograničenje brzine 30 km/h	prestanak ograničenja brzine 80 km/h
ograničenje brzine 50 km/h	ograničenje brzine 100 km/h
ograničenje brzine 60 km/h	ograničenje brzine 120 km/h
ograničenje brzine 70 km/h	zabrana obilaska
ograničenje brzine 80 km/h	zabrana obilaska kamiona



**Tablica 4.2.** *Popis prometnih znakova za klasifikaciju.*

Naziv znaka	
raskrižje sa sporednom cestom pod pravim kutom	sklizak kolnik
raskrižje s cestom koja ima prednost prolaska	zabrana ulaska za teretna vozila
cesta s prednošću prolaska	zabrana prometa u jednom smjeru
obavezno zaustavljanje	opasnost na cesti
zabrana prometa u oba smjera	neravan kolnik
suženje ceste s desne strane	prestanak svih ograničenja
radovi na cesti	obavezan smjer desno
nailazak na prometna svjetla	obavezan smjer lijevo
djeca na cesti	obavezan smjer ravno
poledica	dopušteni smjerovi ravno i desno
obavezno obilaženje s desne strane	prestanak zabrane obilaska
kružni tok	

Prethodno navedene vrste prometnih znakova prikazane su na slikama 4.1., 4.2. i 4.3.

Prilikom konstruiranja baze slika prometnih znakova, obraćena je pozornost na prilagođavanje iste prometnicama različitih zemalja svijeta, stoga su korištene slike prikupljene iz sljedećih postojećih baza slika prometnih znakova: belgijske baze slika [29], *Lisa* američke baze slika [30], kineske baze slika, njemačke baze slika [31], slovenske baze slika [11] te švedske baze slika [32], [33]. Uz to, prikupljene su slike iz videa snimanih na osječkim prometnicama za potrebe izvedbe ovog rada.

Iz navedenog može se uvidjeti kako su prometnice Europske unije dobro zastupljene prilikom izrade baze podataka te će, sukladno navedenom, model mreže evaluiran u ovom radu biti prikladan za korištenje širom Europe. Pritom, navedene slike je bilo potrebno ručno označiti, a isto je učinjeno korištenjem *LabelImg* alata opisanog u sljedećem odjeljku. Ručno označavanje vrši se u svrhu lakšeg prikupljanja slika prometnih znakova relevantnih za ovaj rad te stvaranju uniformnih identifikacijskih oznaka prometnih znakova kako bi se isti mogli lakše klasificirati.



a)



b)

**Slika 4.1.** Prikaz prometnih znakova koje je potrebno identificirati korištenjem konvolucijskih neuronskih mreža, a) ograničenje brzine 30 km/h, ograničenje brzine 50 km/h, ograničenje brzine 60 km/h, ograničenje brzine 70 km/h, ograničenje brzine 80 km/h, b) prestanak ograničenja brzine 80 km/h, ograničenje brzine 100 km/h, ograničenje brzine 120 km/h, zabrana obilaska, zabrana obilaska kamiona.

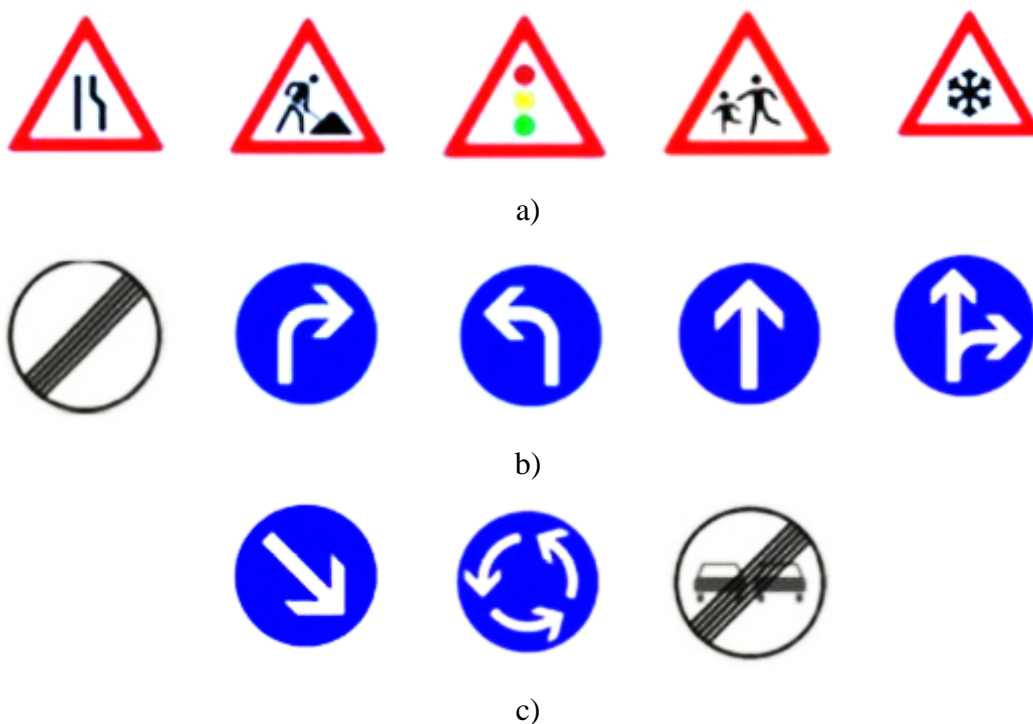


a)



b)

**Slika 4.2.** Prikaz prometnih znakova koje je potrebno identificirati korištenjem konvolucijskih neuronskih mreža, a) raskrižje sa sporednom cestom pod pravim kutom, raskrižje s cestom koja ima prednost prolaska, cesta s prednošću prolaska, obavezno zaustavljanje, zabrana prometa u oba smjera, b) zabrana ulaska za teretna vozila, zabrana prometa u jednom smjeru, opasnost na cesti, neravan kolnik, sklizak kolnik.



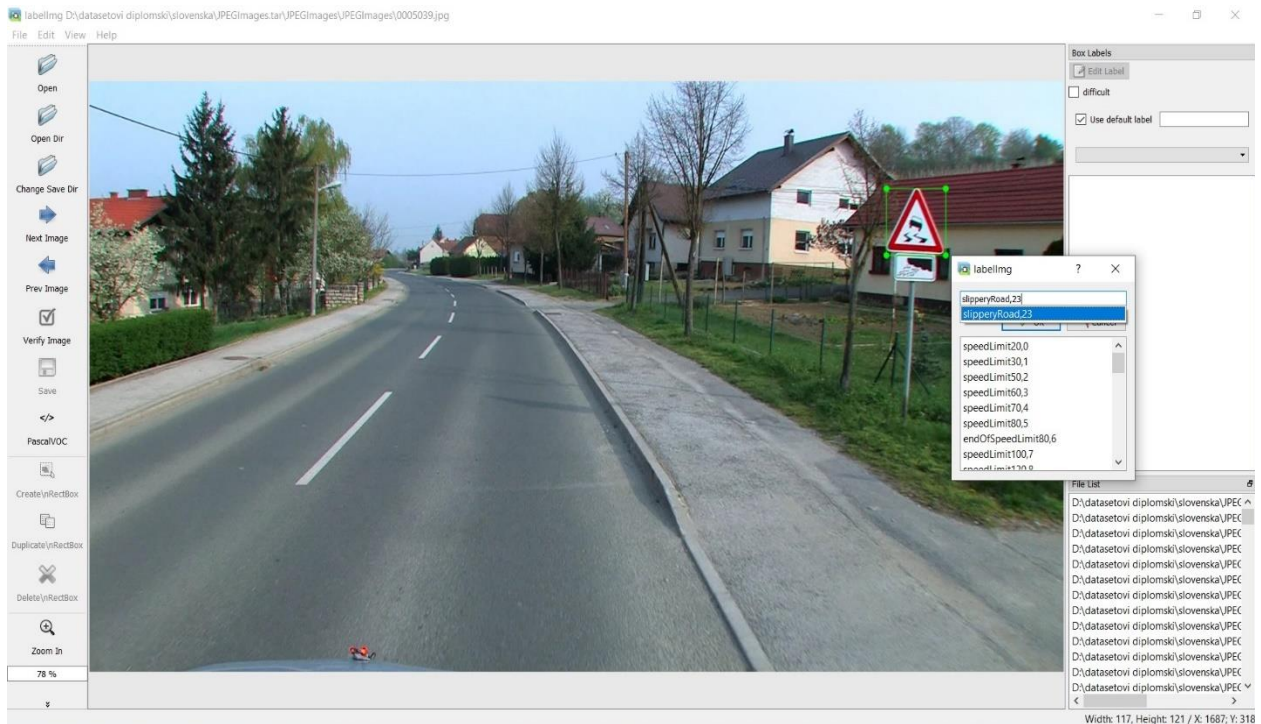
**Slika 4.3.** Prikaz prometnih znakova koje je potrebno identificirati korištenjem konvolucijskih neuronskih mreža, a) suženje ceste s desne strane, radovi na cesti, nailazak na prometna svjetla, djeca na cesti, poledica, b) prestanak svih ograničenja, obavezan smjer desno, obavezan smjer lijevo, obavezan smjer ravno, dopušteni smjerovi ravno i desno, c) obavezno obilaženje s desne strane, kružni tok, prestanak zabrane obilaska.

#### 4.1.1. Alati korišteni za označavanje baze prometnih znakova

Kako bi se uspješno provela evaluacija konvolucijskih neuronskih mreža nad prilagođenom bazom podataka, korišteno je nekoliko alata. Ponajprije, korišten je *LabelImg*, besplatan grafički alat za označavanje podataka, koji sprema podatke o anotacijama na slikama u PASCAL VOC formatu. Podatke je potrebno naknadno obraditi, kako bi se stvorila željena datoteka u CSV formatu. Iz CSV datoteke neuronska mreža može čitati podatke o prisutnosti prometnih znakova na slikama te pripadajućim koordinatama graničnih okvira istih. Inače, *LabelImg* alat napisan je u *python* skriptnom jeziku te koristi *Qt* grafičko sučelje, a isti se otvara pokretanjem *labelImg.py* skripte iz naredbenog retka. Dakako, prije samog pokretanja spomenutog programa, potrebno je preuzeti izvorni kod istog, kako je i navedeno u [28].

Prilikom označavanja slika *LabelImg* alat generira *XML* datoteke, u kojima se nalaze informacije o putanji i nazivu datoteke označene slike, podaci o visini i širini slike te podaci o svim označenim objektima na određenoj slici. Podaci o označenim objektima uključuju naziv klase

kojoj označeni objekt pripada te koordinate graničnog okvira unutar kojeg se isti nalazi. Primjer postupka označavanja objekta unutar slike korištenjem *LabelImg* alata nalazi se na slici 4.4.



**Slika 4.4.** Prikaz grafičkog sučelja *LabelImg* alata i načina označavanja objekta unutar istog na primjeru označavanja prometnog znaka upozorenja na sklizak kolnik.

#### 4.1.2. Augmentacija baze slika prometnih znakova

Sukladno činjenici da postoje prometni znakovi koji se puno češće pojavljuju u prometu, broj određenih znakova u izrađenoj bazi podataka znatno je manji u odnosu na broj potrebnih znakova da mreža što uspješnije nauči lokalizirati i klasificirati iste. Na tragu navedenoga, uvodi se augmentacija slika kojom se raznim tehnikama obrade slike, poput rotacije, translacije, zamučivanja slike, dobivaju novi primjeri postojećih slika. Na taj način se omogućava neuronskoj mreži lakša generalizacija skupa slika za učenje te se, na spomenuti način, smanjuje mogućnost pretjeranog usklađivanja mreže na trening skup slika prilikom treniranja iste (engl. *overfitting*).

Važno je napomenuti da se, u praksi, augmentacija vrši isključivo na skupu slika za treniranje jer se sama bit augmentacije zasniva na poboljšanju učenja modela neuronske mreže. U većini slučajeva, nema smisla upotrebljavati augmentirane podatke prilikom validacije i testiranja modela. Validacija i testiranje se trebaju vršiti na slikama iz stvarnog svijeta na kojima se zaista može utvrditi uspješnost rada modela. Ipak, u posljednje vrijeme pojavljuje se pojam augmentacije tijekom testiranja (engl. *test-time augmentation*), pod kojim se najčešće podrazumijeva nasumična

uporaba nekih od tehnika augmentacije za svaku sliku prilikom postupka predikcije. Prema [34], navedeno se izvodi kako bi se neuronskoj mreži u postupku predikcije pružilo više verzija promatrane slike i kako bi različite instance korištenog modela glasale, na osnovu navedenih slika, kojoj od potencijalnih izlaznih klasa promatrani objekt unutar slike pripada.

U svakom slučaju, prije same augmentacije potrebno je izvršiti podjelu baze slika na trening, validacijski i testni dio. U ovom radu, navedena podjela izvršena je tako da se 70 % ukupnog broja slika spremio u skup za trening, 20 % slika u validacijski te 10 % ukupnog broja slika u testni skup. Prije postupka augmentacije, također, izvršena je promjena veličina svih slika na rezoluciju 800x600 kako bi se smanjilo memorijsko opterećenje korištene grafičke kartice tijekom treninga mreže te ubrzao sam postupak treniranja. Obzirom na visoku rezoluciju originalnih slika, prilikom smanjenja iste, ne dolazi do gubitka važnih podataka u slici.

Također, ako se podjela baze podataka izvršava nakon augmentacije, vrlo lako se može narušiti vjerodostojnost validacijskog i testnog skupa jer augmentirane slike mogu dospjeti u skupove slika za validaciju i testiranje. Uslijed navedenog, spomenuti skupovi slika više ne bi bili jedinstveni jer bi sadržavali inačice slika na kojima je vršeno učenje modela.

Skup svih znakova iste vrste čine jednu klasu, a uz navedene klase uvedena je i klasa tzv. negativnih primjera slika, na kojima se ne nalaze prometni znakovi ili se nalaze prometni znakovi koji ne pripadaju nijednoj klasi navedenoj u ovom radu.

Na slici 4.5. prikazani su ukupni brojevi uzoraka promatranih klasa prometnih znakova. Moguće je uočiti kako baza podataka ima vrlo različit broj znakova u svakoj klasi. Razlog tomu leži u već spomenutoj rasprostranjenosti prometnih znakova na prometnicama, ali i po državama općenito. Tako je, na primjer, znak prestanka ograničenja brzine 80 km/h iznimno karakterističan za njemačke prometnice, ali isti je rjeđe moguće vidjeti na prometnicama drugih država Europe.



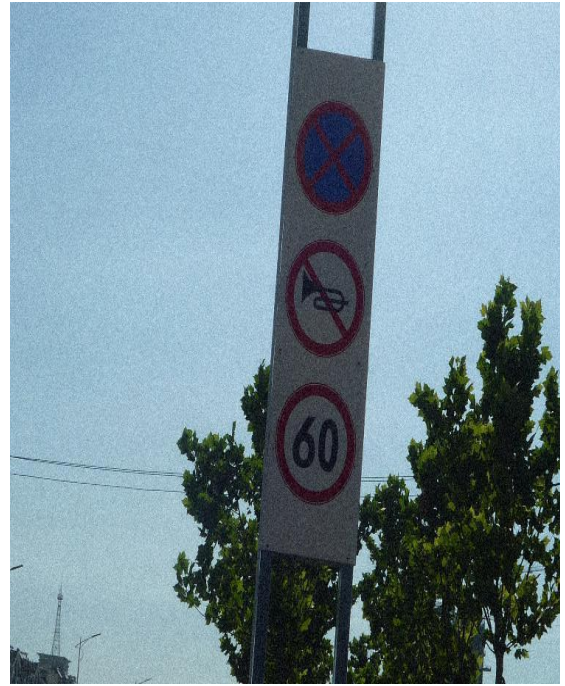
**Slika 4.5.** Prikaz broja uzoraka prometnih znakova u skupu za učenje prije augmentacije.

Na tragu navedenoga, očito je da je potrebno izvršiti augmentaciju prikazanog skupa za trening. Za potrebe augmentacije skupa slika za učenje, u ovom radu korištena je *ImgAug python* biblioteka za automatsku augmentaciju slika. Prema [35], spomenuta biblioteka nudi mnogo augmentacijskih tehnika, a neke od istih su: mijenjanje kontrasta, dodavanje Gaussovog šuma u slike, odbacivanje pojedinih dijelova slike i tako dalje. Osim toga, augmentacije nad slikama moguće je kombinirati i nasumično primjenjivati. Štoviše, *ImgAug* biblioteka ima mogućnost automatskog podešavanja graničnih okvira unutar slike u skladu s promjenama izvođenima tijekom same augmentacije.

Za potrebe ovog rada definirano je nekoliko vrsta tehnika augmentacija koje se primjenjuju na slikama, a to su:

- nasumična rotacija od minimalno  $-5^\circ$  do maksimalno  $5^\circ$  kako ne bi došlo do promjene značenja prometnog znaka
- translacija, smicanje (engl. *shear*) do maksimalno osam elemenata slike
- povećanje i smanjivanje slika do najviše 50 % cijele slike
- korištenje najbližeg susjeda bilinearne interpolacije za popunjavanje elemenata prilikom skaliranja slike
- množenje svih elemenata slika nasumičnim brojem između 0,5 i 1,5 kako bi se slika posvijetlila ili potamnila
- dodavanje šuma primjenom srednje vrijednosti elemenata slike (engl. *Average Blur*)
- dodavanje šuma pomoću Gaussovog filtra (engl. *Gaussian blur*)
- uklanjanje šuma pomoću Median filtra (engl. *Median blur*)
- izoštravanje slike
- dodavanje linearnog kontrasta
- dodavanje ispupčenja u slici (engl. *emboss*).

Prethodno navedene tehnike augmentacije ne mijenjaju značenje prometnih znakova u slici, već omogućavaju davanje različitih uvjeta u slici koji se potencijalno mogu pojaviti u stvarnosti. Tako na primjer, povećanjem/smanjenjem kontrasta simulira se pojava različite osvijetljenosti tijekom vožnje. Nadalje, rotacijom simulira se znak na cesti koji je zbog loših vremenskih uvjeta ili nezgoda nakrivljen. Pritom, ne upotrebljavaju se svi gore navedeni postupci augmentacije na svakoj slici, već se, nasumičnim odabirom upotrebljavaju dvije od navedenih metoda augmentacije na svakoj slici. Neki od primjera primjene navedenih nasumičnih augmentacija prikazani su na slikama 4.6., 4.7., i 4.8., a uz augmentirane slike priloženi su i pripadajući originali istih. Augmentacije su provođene za sve klase koje broje manje od 1000 znakova, a glavni cilj bio je stvoriti bazu slika u kojoj svaka od 33 klasa ima minimalno 1000 uzoraka prometnih znakova.



a)



b)

**Slika 4.6.** Prikaz nasumično augmentiranih slika prometnih znakova (desno) i pripadajućih originala (lijevo) na primjeru prometnog znaka: a) ograničenja brzine 60 km/h, b) obaveznog smjera kretanja ravno.





**Slika 4.7.** *Prikaz nasumično augmentiranih slika prometnih znakova (desno) i pripadajućih originala (lijevo) na primjeru prometnog znaka skliskog kolnika.*

Kao što je spomenuto u Poglavlju 2, identifikacija prometnih znakova snimljenih s različitih udaljenosti jedan od najvećih izazova danas. Na tragu toga, prilikom izrade baze slika za potrebe izvođenja zadatka ovog rada prikupljene su slike prometnih znakova snimane s različitih udaljenosti.

Trenirajući mrežu na prometnim znakovima čije veličine variraju u slici čine istrenirani model robusnijim. Uz to, veća je šansa za uspješnijim radom odabranog modela u realnim uvjetima ako se isti istrenira na kvalitetnoj i robusnoj bazi.

Na slici 4.8. može se vidjeti kako se, zumiranjem slike kao tehnikom augmentacije, povećava prometni znak ograničenja brzine 120 km/h unutar slike. Navedenim se simulira snimanje istog prometnog znaka s različitih udaljenosti te dodatno pospješuje kvaliteta baze.



a)



b)

**Slika 4.8.** Prikaz nasumično augmentiranih slika prometnih znakova (a) i pripadajućih originala (b) na primjeru prometnog znaka ograničenja brzine 120 km/h.

Nakon augmentacije baze slika za trening, ukupan broj slika u bazi podataka iznosi 39 357 slika za trening i validaciju te 2 317 slika za testiranje, a broj uzoraka po klasama nakon augmentacije moguće je vidjeti na slici 4.9.



**Slika 4.9.** Prikaz broja uzoraka prometnih znakova u skupu za učenje nakon augmentacije.

Postoje određeni prometni znakovi, kao npr. prometni znakovi ograničenja brzine 70 km/h i ceste s prednošću prolaska, koji imaju veću zastupljenost na prometnicama u odnosu na druge prometne znakove. Nasuprot tome, postoje znakovi koji se pojavljuju rjeđe u prometu, kao npr., prometni znakovi prestanka ograničenja brzine 80 km/h i obaveznog smjera kretanja lijevo. Uz to, često se zastupljeniji znakovi u prometu pojavljuju u kombinaciji s nekim prometnim znakovima

koje se može rjeđe vidjeti u prometu. Zbog navedenog je neposredno došlo do značajnog porasta broja čestih prometnih znakova, što se jasno može uočiti sa slike 4.9.

Obzirom da RetinaNet prima podatke prilagođene baze slika prema točno određenim pravilima kako bi se baza podataka mogla uspješno učitati, bilo je potrebno navoditi spomenute podatke na sljedeći način:

- svaki prometni znak unutar slike mora stajati u jednom redu datoteke (u slučaju pojave više označenih prometnih znakova unutar jedne slike, potrebno je navesti svaki od spomenutih u poseban red)
- svaki redak sastoji se od šest podataka odvojenih zarezima, a to su: naziv datoteke (uključujući putanju do iste ako je potrebna),  $x1$  koordinata graničnog okvira,  $y1$  koordinata graničnog okvira,  $x2$  koordinata graničnog okvira,  $y2$  koordinata graničnog okvira te naziv klase (sl. 4.10. a)
- ako je riječ o slici iz negativne klase, potrebno je navesti naziv datoteke (uključujući putanju do iste ako je potrebna) te ostatak retka odvojiti zarezima bez navođenja drugih informacija (sl. 4.10. b)
- potrebno je izraditi posebnu datoteku, koja sadrži popis svih klasa i pripadajućih indeksa koji se dodjeljuju, slijedno, svakoj klasi, prema redu navođenja unutar datoteke (sl. 4.10. c).

```
resizeImg_/sweden/resizeImg1277381674Image000001.jpg,258,353,272,368,keepRight
```

a)

```
sweden/1277104235Image000037.jpg,,,,,
```

b)

```
speedLimit30,0
```

```
speedLimit50,1
```

c)

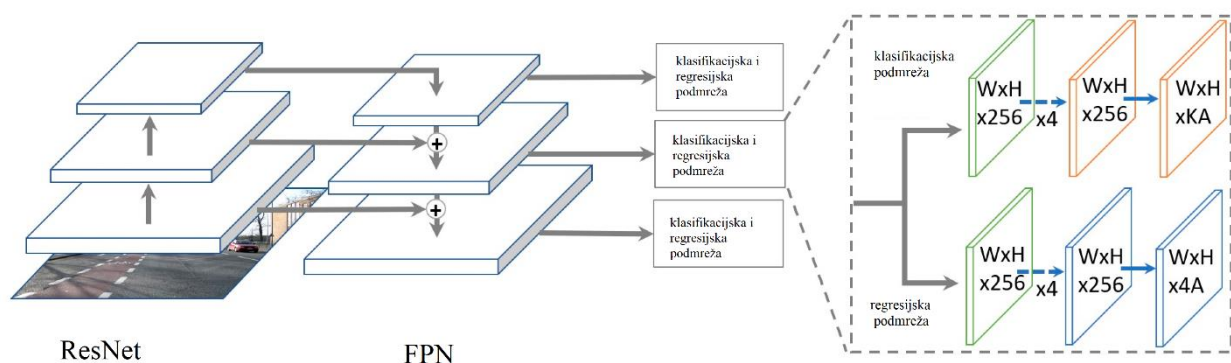
**Slika 4.10.** Prikaz primjera zapisa podataka unutar CSV datoteka za RetinaNet mrežu, a) zapis označenog prometnog znaka obaveznog obilaska desno, b) zapis datoteke koja pripada klasi s negativnim primjerima slika, c) nazivi klasa prometnih znakova ograničenja brzine 30 km/h i 50 km/h te pripadajućih indeksa.

Na tragu navedenoga, korištenjem posebne skripte pisane u *pythonu*, izvršena je pretvorba prethodno spomenutog PASCAL VOC formata u gore opisani način zapisa, koji je, zatim, spremljen u CSV format datoteke.

## 4.2. Arhitektura RetinaNet modela duboke konvolucijske neuronske mreže

RetinaNet je duboka konvolucijska neuronska mreža koja se, kao što je navedeno u [36], sastoji od FPN-a (engl. *Feature Pyramid Network*), podmreže za klasifikaciju i podmreže za regresijsko određivanje graničnih okvira objekata. Njezina detaljna arhitektura prikazana je na slici 4.11., a u daljnjem razvoju teksta ista će biti detaljnije objašnjena.

Prema [37], RetinaNet pripada tzv. detektorima s jednom fazom (engl. *Single-shot detector*, SSD), koji se inače sastoje od okosnice (engl. *backbone*) te SSD zaglavlja (engl. *SSD head*). Osim toga, umjesto klizajućeg prozora za detekciju objekata, RetinaNet koristi mrežu ćelija (engl. *grid cell*), od kojih je svaka odgovorna za detekciju objekata u vlastitoj regiji. Pritom, svakoj ćeliji moguće je pridijeliti više graničnih okvira (engl. *anchor boxes*), čije su veličine predefinirane te se prilikom detekcije vrši usporedba spomenutih graničnih okvira s graničnim okvirima detektiranog objekta. Granični okvir određene ćelije koji se najbolje poklapa s graničnim okvirom objekta zadužuje se za predikciju položaja i klase istog. Navedene karakteristike iznimno su praktične jer omogućuju sigurniju klasifikaciju objekta na slici, ističući pritom i točan položaj istog. Uostalom, proces klasifikacije u slici koja ne sadrži isključivo sam objekt koji je potrebno prepoznati mora se zasnivati i/ili neposredno nadovezivati na određivanju položaja određenog objekta, kako bi se isti uopće mogao ispravno klasificirati.



Slika 4.11. Arhitektura RetinaNet konvolucijske neuronske mreže, iz [26].

#### 4.2.1. FPN u RetinaNet modelu

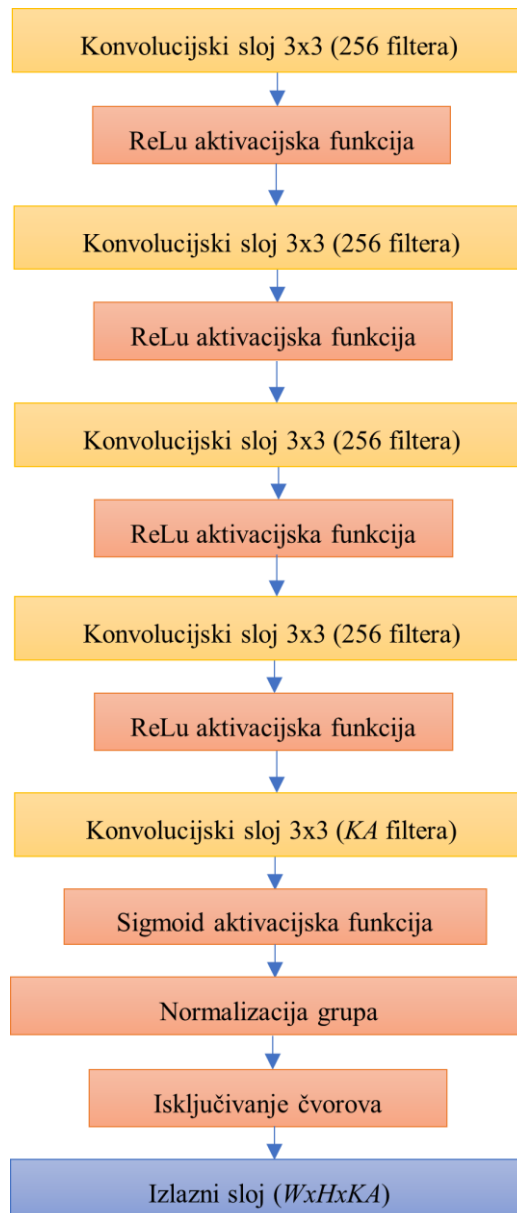
FPN se, kako je navedeno u [38], smatra osnovnom komponentom sustava za prepoznavanje objekata jer omogućuje detekciju istih u različitim dijelovima slike te u različitim veličinama. Riječ je, zapravo, o konvolucijskoj neuronskoj mreži, koja na ulazu prima sliku na kojoj se nalazi objekt koji je potrebno prepoznati te na osnovu istog kreira izlaznu mapu značajki na više razina, što na kraju tvori piramidu značajki (engl. *feature pyramid*). Navedena višerazinska piramida značajki omogućava detekciju objekata različitih veličina u slici jer se svaka razina piramide koristi za detekciju objekata drugačije veličine. Osim toga, u RetinaNet mreži, FPN predstavlja mrežu okosnicu koja se nadovezuje na ResNet-50 duboku konvolucijsku neuronsku mrežu, koja se koristi za izdvajanje značajki u slici, kako je opisano u [36]. Nadalje, u RetinaNet mreži konstruirana je piramida koja se sastoji od razina  $P3$  do  $P7$ . Putanja u samoj piramidalnoj strukturi može se kretati odozgor prema dolje i obrnuto, odozdol prema gore. Pritom, putanja odozdol prema gore omogućava generiranje hijerarhije značajki na osnovu aktivacijskih mapi različitih razina piramide značajki. Nasuprot tome, putanja odozgor prema dolje ima tzv. lateralne veze, kojima se omogućava stvaranje mapi značajki različitih veličina.

#### 4.2.2. Klasifikacijska podmreža

Prema [36] i [38], klasifikacijska podmreža zadužena je za predikciju vjerojatnosti pojave određenog objekta za svaki od graničnih okvira. Riječ je o maloj potpuno povezanoj konvolucijskoj neuronskoj mreži koja je spojena sa svakom razinom FPN mreže, a sastoji se od četiri  $3 \times 3$  konvolucijska sloja s 256 filtera. Iza svakog od navedenih slojeva primjenjuje se ReLU aktivacijska funkcija. Uz navedeno, dodaje se još jedan konvolucijski sloj, čiji broj filtera odgovara umnošku ukupnog broja prisutnih graničnih okvira i broja klasa objekata za prepoznavanje. Naposljetku, dodaje se Sigmoid aktivacijska funkcija povezana na izlaz iz same mreže, čije dimenzije iznose  $W \times H \times KA$ , kako je i prikazano na slici 4.11. Pritom, vrijednosti  $W$  i  $H$  su proporcionalne širini i visini ulazne aktivacijske mape, a  $KA$  predstavlja spomenuti umnožak broja klasa objekata za prepoznavanje i ukupnog broja prisutnih graničnih okvira. Na tragu navedenoga, izlaz iz konvolucijske podmreže je, zapravo, predikcija klase po prostornoj lokaciji na slici. Također, treba napomenuti kako se parametri cijele mreže opisane u ovom odjeljku dijele sa svim razinama piramide značajki.

Za potrebe ovog rada na opisani model klasifikacijske podmreže nadodana je normalizacija grupa (engl. *batch normalization*) s momentom 0,8 te isključivanje čvorova iz mreže

s vjerojatnošću 0,5. Navedeno je dodano nakon implementacije Sigmoid aktivacijske funkcije, a prikaz opisane klasifikacijske podmreže korištene u ovom radu dan je na slici 4.12.



**Slika 4.12.** Prikaz arhitekture klasifikacijske podmreže RetinaNet mreže.

#### 4.2.3. Regresijska podmreža

Regresijska podmreža spojena je na FPN paralelno s klasifikacijskom podmrežom, a glavna uloga iste je određivanje razmaka (engl. *offset*) između postojećih graničnih okvira i lokaliziranih objekata u blizini istih. Konstrukcija spomenute mreže jednaka je konvolucijskoj podmreži opisanoj u prethodnom odjeljku. Razlika se nazire u izlazu iz mreže gdje se umjesto broja klasa prosljeđuju  $4A$  linearni izlazi po prostornoj lokaciji, pri čemu se za svaki od graničnih okvira predviđaju četiri izlaza, koji predstavljaju razmake između graničnih okvira i lokaliziranih

objekata u blizini istih, kako je i pojašnjeno u [38]. Slično kao i kod klasifikacijske podmreže, za potrebe ovog rada, na opisani model regresijske podmreže nadodana je normalizacija grupa s momentom 0,8 te isključivanje čvorova iz mreže s vjerojatnošću 0,5.

#### 4.2.4. Kriterijska funkcija

Iznosi kriterijske funkcije u RetinaNet mreži dobivaju se pomoću iznosa kriterijskih funkcija klasifikacije i lokalizacije i to prema sljedećem izrazu, kako je navedeno u [36]:

$$L = \lambda L_{loc} + L_{class}, \quad (4-1)$$

gdje  $\lambda$  predstavlja balansirajući faktor između iznosa klasifikacijske i lokalizacijske kriterijske funkcije,  $L_{loc}$  predstavlja iznos kriterijske funkcije lokalizacije, a  $L_{class}$  iznos kriterijske funkcije klasifikacije. Glavna svrha prilikom treniranja neuronske mreže je minimizirati navedenu kriterijsku funkciju. Nadalje, prilikom računanja iznosa kriterijskih funkcija klasifikacije koristi se tzv. *focal loss* funkcija, predložena u [37], kojom se uspješno prevladavaju problemi nebalansiranih klasa (određeni primjeri sa lako odvojivom pozadinom počinju dominirati učenjem mreže, što često uzrokuje lošiju točnost prepoznavanja objekta). Spomenuta funkcija, prema [36], prikazuje se pomoću:

$$L_{class} = -\sum_{i=1}^C \{a_i y_i (1 - p_i)^y \log(p_i) + (1 - a_i)(1 - y_i) p_i^y \log(1 - p_i)\}, \quad (4-2)$$

gdje  $C$  označava broj klasa,  $p_i$  vrijednost predikcije za pojedinu klasu  $i$ ,  $y$  označava parametar fokusiranja (u iznosu od 0 do  $+\infty$ ),  $a_i$  faktor težine za pojedinu klasu  $i$  (u iznosu 0-1). Pritom, prema [36], uvođenjem faktora modulacije  $(1 - p_i)^y$  u kriterijsku funkciju unakrsne entropije omogućava se smanjivanje iznosa težina jednostavnijih uzoraka prilikom treniranja konvolucijske mreže, uslijed čega dolazi do fokusa na teže uzorke (negativne klase) prilikom izvođenja treniranja. U slučaju da vrijednost predikcije točno klasificiranog uzorka teži ka jedan, faktor modulacije postaje 0, a iznos kriterijske funkcije klasifikacije se, prema [36], smanjuje.

#### 4.2.5. Postavljanje parametara za treniranje RetinaNet mreže

U ovom odjeljku bit će razjašnjeni ključni pojmovi, čije je razumijevanje potrebno za izvedbu uspješnog pokretanja i izvršavanja preinaka na postojećem modelu RetinaNet mreže.

Ponajprije, valja spomenuti preneseno učenje (engl. *transfer learning*), čijom se uporabom omogućava korištenje težina već istreniranih modela neuronskih mreža, kako je pojašnjeno u [39]. Pritom se spomenute težine uzimaju kao početne vrijednosti težina treniranja mreže na



prilagođenoj bazi podataka. U slučaju da se učenje mreže vrši bez početnih težina, iste se inicijaliziraju na nulu te se, tijekom procesa učenja mreže, postepeno ažuriraju. Navedeni proces je znatno sporiji u odnosu na primjenu težina prethodno istrenirane mreže. Također, uzimaju se obzir isključivo težine koje su trenirane na velikom skupu podataka s dovoljno općenitim klasama koje je potrebno prepoznati, kao što su *ImageNet* i *COCO* skupovi slika za klasifikaciju i detekciju objekata. Osim toga, proces prenesenog učenja može se izvoditi tzv. izdvajanjem značajki (engl. *feature extraction*), pri čemu se vrši ažuriranje težina klasifikacijske i regresijske podmreže tijekom treninga, a okosnica modela čitave mreže se zamrzne (engl. *freeze backbone*), što znači da se tijekom treninga težine iste neće ažurirati. Suprotno tome, trening prenesenim učenjem može se vršiti i finim podešavanjem (engl. *fine-tuning*), pri čemu se odmrzava okosnica ili samo gornji slojevi iste te se ažuriranje težina u navedenom slučaju vrši i na odmrznutim dijelovima mreže, kako je i opisano u [40].

Nadalje, pojava u kojoj mreža tijekom treninga prođe kroz cijelu bazu podataka naziva se epoha (engl. *epoch*). Obzirom da se ne može cijela baza podataka odjednom prosljediti mreži na obradu, ista se dijeli na manje grupe (engl. *batch*), koje se, zatim, prosljeđuju mreži. Sukladno tome, mreža mora proći kroz određeni broj iteracija (engl. *steps*), potrebnih da obradi sve grupe podataka kako bi uopće prošla kroz cijelu bazu podataka. Na tragu spomenutoga, može se definirati izraz za računanje broja iteracija koje je potrebno postaviti RetinaNet mreži da bi ista u jednoj epohi prošla kroz cijelu bazu slika za trening:

$$\text{broj iteracija} = \frac{\text{broj uzoraka prometnih znakova}}{\text{broj uzoraka prometnih znakova u jednoj grupi}} \quad (4-3)$$

Tehnika isključivanja čvorova (engl. *dropout*) često se koristi kako bi se izbjeglo pretjerano usklađivanje modela na podatke prilikom treninga, kako je i navedeno u [41]. Pod pojmom pretjeranog usklađivanja podrazumijeva se da je model koji se trenira jako dobro naučio skup slika za treniranje. Međutim, kada isti treba prepoznati objekte iz slika u testnom skupu, postiže puno lošije rezultate, nego prilikom treniranja. Naime, riječ je o postupku kojim se određeni čvorovi tijekom treninga ispuštaju, odnosno ne uzimaju se u obzir za daljnji tijek treninga mreže. Pritom, potrebno je zadati vjerojatnost (u iznosu 0 - 1) da se određeni čvor isključi iz mreže. Prilikom isključivanja čvorova iz mreže, ispuštaju se i poveznice istog s prethodnim i sljedećim čvorovima.

### 4.3. Programsko rješenje za prepoznavanje prometnih znakova

Kako bi se pokrenula i testirala RetinaNet, korišten je Anaconda 3 naredbeni terminal. Anaconda 3 znatno olakšava rukovanje potrebnim okruženjima i paketima za uspješan rad svih

komponenti sustava. Pritom, ističe se *TensorFlow* platforma kojom je omogućeno odvijanje treninga mreže na grafičkoj kartici računala. Navedenim se značajno ubrzava proces treniranja i evaluacije rada konvolucijske neuronske mreže. Također, uz *TensorFlow*, korišten je *Keras*. *Keras* je aplikacijsko programsko sučelje pisano u *python* skriptnom jeziku, primarno stvoreno za dodatno olakšavanje rukovanja modelima strojnog i dubokog učenja.

Uza sve navedeno, stvoreno je *python* okruženje koje je potrebno aktivirati unutar Anaconda 3 naredbenog terminala te unutar kojeg se vrši instalacija paketa potrebnih za pokretanje neuronske mreže na grafičkoj kartici. Tako su instalirani: *CUDA toolkit 10.1*, koja sadrži biblioteke, funkcije, C++ prevoditelj i druge alate koji omogućavaju optimalno korištenje resursa grafičke kartice, *cudnn 7.6.5*, biblioteka za prilagođeni rad s modelima dubokih neuronskih mreža, također, optimizirana za rad s grafičkom karticom, *TensorFlow-GPU 2.1.*, *Keras 2.3*, *openCV*, biblioteka potrebna za obradu slika, *pandas*, biblioteka za rad i obradu datoteka CSV formata te, prethodno opisana, *ImgAug*, biblioteka za augmentaciju podataka.

Osim toga, bilo je potrebno instalirati prethodno spomenute verzije *CUDA* i *cudnn* alata na samo računalo izvan Anaconda 3 okruženja, kao i najnoviju verziju upravljačkog programa za korištenu grafičku karticu. Na tragu navedenoga, za potrebe izvedbe modela konvolucijske neuronske mreže u ovom radu, korištena je Nvidia RTX2070 MQ grafička kartica, koja ima 8GB memorije.

Kako bi se mogao pratiti tijek učenja mreže, može se upotrebljavati *Tensorboard* alat. Inače, riječ je o grafičkom alatu prilagođenom za rad s *Tensorflow* platformom, koji omogućava praćenje i vizualizaciju metrike, kao na primjer iznose kriterijske funkcije i preciznost prepoznavanja, vizualizaciju modela, uvid u histograme težina te ima mogućnost prikazivanja slika, teksta i audio podataka.

#### **4.4. Treniranje RetinaNet konvolucijske neuronske mreže za lokalizaciju i klasifikaciju prometnih znakova u slici**

Implementacija RetinaNet konvolucijske neuronske mreže izvršena je korištenjem *keras-retinanet 0.5.1* modela, iz [42], u kojem su se za potrebe što uspješnijeg prepoznavanja prometnih znakova opisanih u ovom radu izvršile blage promjene. Navedeni model koristi se u mnogim projektima, ali svakako valja istaknuti CRFNet mrežu koja se temelji na *keras-retinanet 0.5.1* modelu. Navedena mreža se koristi za detekciju objekata pomoću radara i kamere za primjenu u

autonomnoj vožnji. Instalaciju navedenog modela moguće je pokrenuti iz Anaconda 3 naredbenog terminala pozicioniranjem u direktorij unutar kojeg se nalazi preuzet model *keras-retinanet 0.5.1*.

Za pokretanje navedene mreže, potrebno je prirediti datoteke sa popisom slika za učenje zapisane na način opisan u odjeljku 4.1.2. te odrediti parametre potrebne za trening.

Ipak, prije početka treninga RetinaNet konvolucijske neuronske mreže za lokalizaciju i klasifikaciju prometnih znakova u slici, preporuča se adaptacija graničnih okvira na pripremljenoj bazi podataka. Uz to izvršava se pregled baze slika i otklanjanje potencijalnih grešaka unutar iste. Na tragu navedenoga, adaptacija graničnih okvira izvršava se pomoću modela *anchor-optimization*, iz [43], koji dopušta ručno postavljanje faktora skaliranja (engl. *scale*) i omjera (engl. *ratio*), daje mogućnost uzimanja u obzir pomaka, promjene veličine slike i drugo. Međutim, za bazu slika pripremljenu za potrebe izvedbe ovog rada, koristit će se opcija podešavanja omjera graničnih okvira. Navedeni model pokreće se iz naredbenog retka, a potrebno je predati i CSV datoteku, u kojoj se nalaze podaci o slikama iz baze podataka. Također, treba naglasiti kako se model za adaptaciju graničnih okvira direktno nadovezuje na *keras-retinanet 0.5.1*. U slučaju da se ne izvrši adaptacija graničnih okvira, predefiniрана vrijednost u RetinaNet modelu iznositi će 9 graničnih okvira po slici.

Prilikom pokretanja algoritma za adaptaciju graničnih okvira, traže se i ispisuju najbolje vrijednosti u odnosu na predanu bazu podataka te se, po završetku obrade, ispisuju rezultati optimalnih vrijednosti za granične okvire. Navedeni ispis dan je na slici 4.13. Također, u rezultatima podešavanja graničnih okvira nalaze se podaci o faktorima skaliranja i omjerima te broju uzoraka iz baze nad kojima nije moguće pronaći nijedan pripadajući granični okvir, koji u ovom slučaju iznosi 3 251.

```
Optimization ended successfully!  
Final best anchor configuration  
State: 0.1589  
Ratios: [0.453, 0.762, 1.0, 1.312, 2.208]  
Scales: [0.4, 0.5, 0.628]  
Number of labels that don't have any matching anchor: 3251
```

**Slika 4.13.** Rezultati adaptacije graničnih okvira.

Uzimajući u obzir da ukupan broj uzoraka prometnih znakova iznosi 57 920, zaključuje se da baza sadrži dovoljnu količinu znakova koji se mogu dobro lokalizirati. Informacije o omjerima i faktorima skaliranja je, uz standardne iznose veličina i pomaka koje mreža koristi pri učenju i

evaluaciji, potrebno unijeti u posebnu *config.ini* datoteku. Sadržaj datoteke moguće je vidjeti na slici 4.14.

```
[anchor_parameters]
# Sizes should correlate to how the network processes an image, it is not advised to change these!
sizes = 32 64 128 256 512
# Strides should correlate to how the network strides over an image, it is not advised to change these!
strides = 8 16 32 64 128
# The different ratios to use per anchor location.
ratios = 0.453 0.762 1.0 1.312 2.208
# The different scaling factors to use per anchor location.
scales = 0.4 0.5 0.628
```

**Slika 4.14.** Sadržaj *config.ini* datoteke.

Trening RetinaNet uz prilagođenu bazu slika prometnih znakova za potrebe izvedbe ovoga rada izveden je u dva stadija. Pritom, u prvom stadiju tijekom 30 epoha trenirana je cijela mreža, uključujući i okosnicu, kako bi se ažurirale težine čitavog modela sukladno korištenim podacima. Na navedeni način stvorene su početne vrijednosti za trening klasifikatora i regresije prilagođenije zadatku ovog rada. Drugi stadij treninga uključuje zamrzavanje okosnice te treniranje ostatka mreže narednih 29 epoha.

Ekperimentalno je pokazano kako, izravnim prenesenim učenjem uz zamrzavanje okosnice te korištenjem težina ResNet-50 mreže trenirane na *ImageNet* bazi slika, RetinaNet znatno sporije konvergira. Posljedično, konačni rezultati preciznosti i kriterijske funkcije lošiji su u odnosu na primijenjen pristup opisan u prethodnom odlomku. Slično tome, treniranje cijele mreže nakon 30 epoha dovodi do stalnog pada preciznosti nad validacijskim skupom slika te porasta iznosa kriterijske funkcije. Zbog navedenog su težine navedenog modela, nastale nakon 30. epohe treniranja cijele mreže, iskorištene kao početne težine prilikom treniranja mreže uz zamrzavanje okosnice.

Također, prije početka treninga potrebno je odrediti parametre opisane u prethodnim poglavljima ovoga rada. Spomenuti parametri mogu se ručno unijeti u sam kod, ili, jednostavnije, predati u obliku naredbe prilikom pokretanja modela iz naredbenog retka. Na tragu navedenoga, određen je broj iteracija svake epohe koji iznosi 3 620 uz veličinu grupa od 16 u slučaju zamrznute okosnice. Govoreći o treningu čitave mreže, uključujući okosnicu, broj iteracija iznosi 9 652 uz veličinu grupa od 6 uzoraka. Navedeni parametri određeni su sukladno (4-2) te su prikazani u tablici 4.3.

**Tablica 4.3.** Parametri za trening mreža.

RetinaNet	Broj iteracija	Veličina grupe	Broj epoha	Stopa učenja
Zamrznuta okosnica	3620	16	29	0,00146
Nezamrznuta okosnica	9652	6	30	

Iako je mreža stabilnija ako grupe imaju što više uzoraka, memorijska ograničenost grafičke kartice ne dopušta korištenje grupa s više uzoraka od prethodno navedenih. Pritom, treba uzeti u obzir da navedeni parametri vrijede isključivo ako je riječ o okosnici ResNet-50. U suprotnom, npr., ako je riječ o ResNet-101 ili ResNet-152, koje imaju dvostruko i trostruko više slojeva u odnosu na ResNet-50, navedeni broj uzoraka po grupi je potrebno još više smanjiti. Razlog tome je što spomenute okosnice imaju puno više parametara u odnosu na korištenu ResNet-50 okosnicu te zahtijevaju puno više memorije. Također, broj epoha se u ovom slučaju određuje praćenjem rezultata validacije na kraju svake epohe i/ili korištenjem *Tensorboard* vizualnog prikaza te tehnikom ranog zaustavljanja. Pritom se trening mreže okončava ako se postigne broj unaprijed zadanih epoha ili ako se preciznost prepoznavanja validacijskog skupa počne smanjivati. Ponekad se navedeno smanjivanje preciznosti može popraviti smanjivanjem stope učenja, što se, u korištenom modelu može omogućiti, kao jedan od kriterija za praćenje uspješnosti sustava nad validacijskim skupom slika.

Osim toga, prilikom treniranja RetinaNet mreže može se omogućiti validacija iste nakon svake epohe, što uvelike olakšava praćenje iznosa točnosti prepoznavanja i iznosa kriterijske funkcije, kako trening, tako i validacijskog skupa slika. U slučaju da se iznos kriterijske funkcije ne smanjuje tijekom nekoliko epoha, model će pomoću *callback* funkcije smanjiti stopu učenja deset puta. Govoreći o stopi učenja, ako se ista ne odredi prije početka treninga, bit će postavljena na predefiniranu vrijednost u modelu te će iznositi  $10^{-5}$ . Obzirom da je preporuka postaviti veću stopu učenja uz grupe s više uzoraka, postavljena je vrijednost stope učenja na 0,00146. Prikaz pokretanja naredbe za trening može se vidjeti na slici 4.15.

```
python train.py --backbone resnet50 --freeze-backbone --config
putanja_do_konfiguracijske_datoteke --no-resize --lr 0.01 --batch-size 16 --steps 3620
--epochs 100 --weights putanja_do_h5_datoteke_s_težinama --snapshot-path
putanja_do_direktorija_za_pohranjivanje_spremljenih_modela --tensorboard-dir
putanja_do_direktorija_za_pohranjivanje_spremljenih_modela_grafova --compute-val-loss
csv putanja_do_CSV_datoteke_za_trening putanja_do_CSV_datoteke_s_popisom_klasa --val-
annotations putanja_do_CSV_datoteke_za_validaciju
```

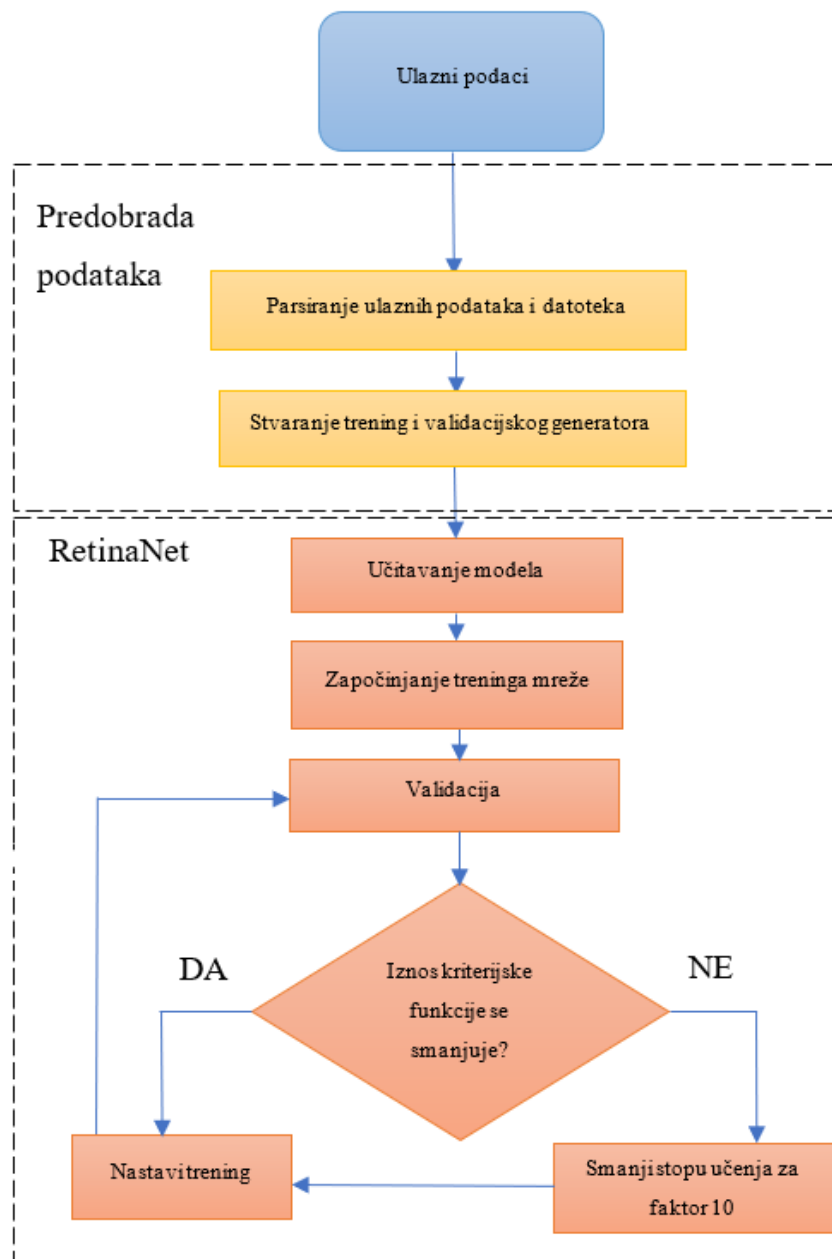
**Slika 4.15.** Naredba za pokretanje treninga RetinaNet mreže.

Prema slici 4.15., može se vidjeti da su svi prethodno opisani parametri proslijeđeni mreži, pri čemu se koriste određene ključne riječi:

- (*--backbone*) definira vrstu okosnice koja se koristi
- (*--freeze-backbone*) označava zamrzavanje okosnice prilikom treninga
- (*--config*) prosljeđuje *config.ini* datoteku s vrijednostima adaptiranih graničnih okvira
- (*--lr*) označava stopu učenja koja će se koristiti tijekom treninga
- (*--batch-size*) označava broj uzoraka u pojedinoj grupi slika
- (*--steps*) označava broj iteracija
- (*--epochs*) označava broj epoha
- (*--weights*) označava putanju na kojoj se nalaze početne težine za trening
- (*--snapshot-path*) označava putanju na koju se spremaju istrenirani modeli nakon svake epohe
- (*--tensorboard-dir*) označava putanju na koju se spremaju *tensorboard* datoteke, koje omogućavaju lakše praćenje treninga mreže konstruiranjem grafova
- (*--compute-val-loss*) govori mreži da mora izračunati vrijednost kriterijske funkcije za validacijski skup podataka nakon svake epohe
- (*csv*) označava da se trening i validacija vrše na prilagođenoj bazi slika, čiji su podaci pohranjeni u CSV formatu te se, nakon *csv* ključne riječi predaju putanje do datoteke s podacima o skupu za trening i datoteke s popisom indeksa i naziva klasa
- (*--val-annotations*) označava putanju do datoteke s validacijskim podacima.

Prethodno opisani postupak treniranja može se svesti na nekoliko generičkih koraka koji su grafički prikazani dijagramom toka na slici 4.16. Riječ je o parsiranju ulaznih podataka i stvaranju trening i validacijskog generatora, učitavanju zadanog modela mreže, trening, validaciju

i provjeru stanja iznosa kriterijske funkcije uspoređujući istu s nekoliko posljednjih epoha na validacijskom skupu.



**Slika 4.16.** Dijagram toka treniranja i validacije keras-retinanet 0.5.1 modela.

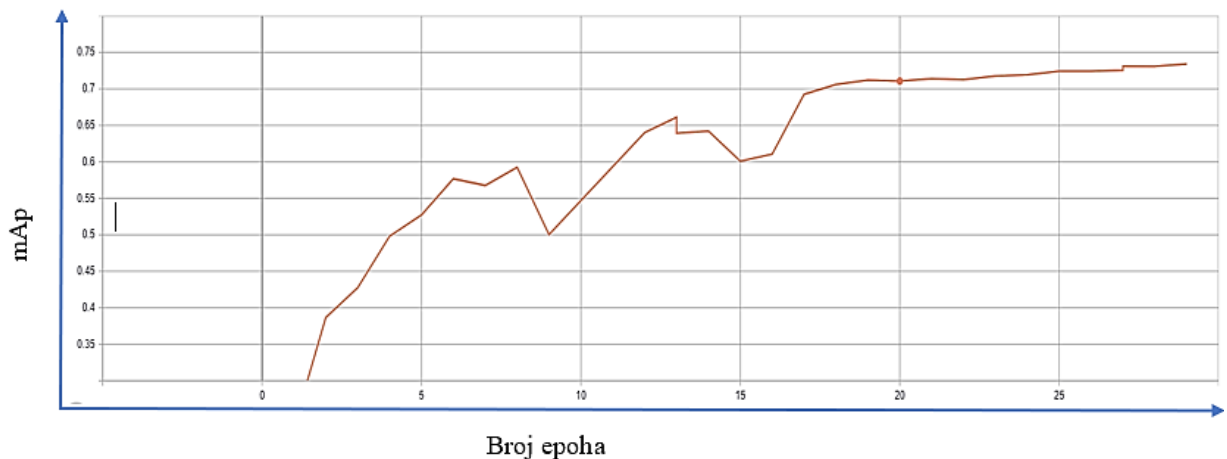
Predobrada podataka uključuje stvaranje trening i validacijskog generatora. Pritom, generator je zapravo definiran posebnom skriptom, a u istom se izvršava parsiranje ulazne datoteke s podacima o bazi za trening i validaciju te učitavanje anotacija i pripadnih graničnih okvira slika. Također, u stadiju predobrade postoji mogućnost za promjenom veličine slika i nasumičnom augmentacijom grupa slika tijekom treniranja. Iako navedeno predstavlja izvrsne funkcionalnosti koje potpomažu procesu učenja mreže, treba uzeti u obzir kako je isto računalno i memorijski

zahtjevno. Usred korištenja spomenutih funkcionalnosti, količina uzoraka prometnih znakova po grupi morala bi se znatno smanjiti, stoga se u ovom radu sve potrebne transformacije slika izvršavaju prije učitavanja modela prema opisu u potpoglavlju 4.2.1.

Slika 4.17. prikazuje graf promjene kriterijske funkcije i glavne prosječne preciznosti tijekom treninga mreže. Iz grafa je moguće vidjeti kako mAp konvergira do 29. epohe te tada postiže najveću vrijednost od 74 % na validacijskom skupu slika. Također, vidljivo je kako je u pojedinim epohama došlo do smanjenja iznosa kriterijske funkcije, odnosno smanjenja iznosa mAp, što je znak pretjeranog usklađivanja na testni skup slika. Ipak, navedeni problem je uspješno riješen, što je vidljivo iz slike 4.17, a rješenje istog bit će detaljnije razjašnjeno u nastavku teksta.



a)



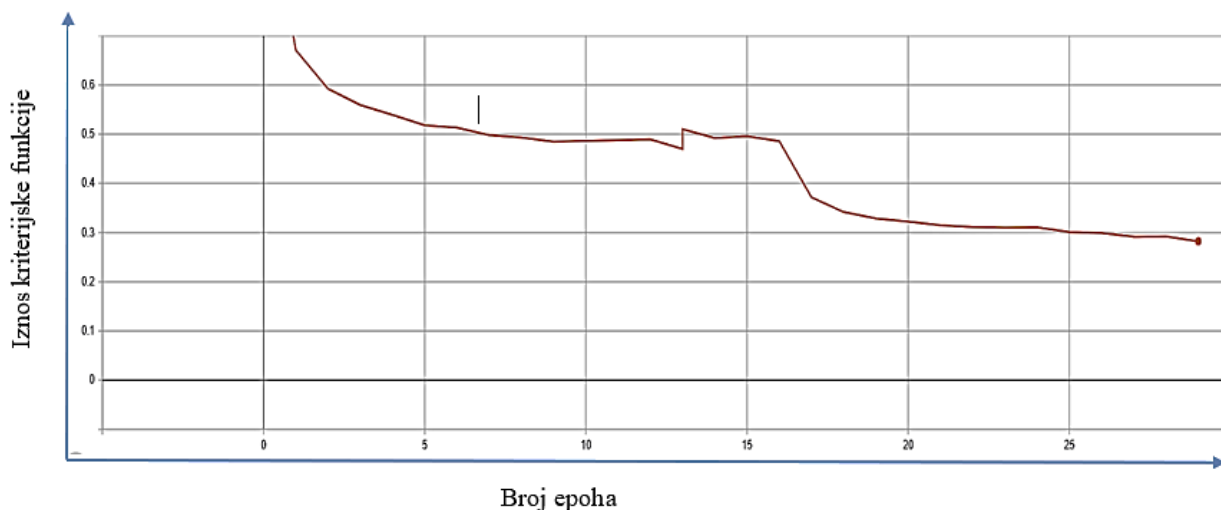
b)

**Slika 4.17.** Prikaz krivulja generiranih tijekom treninga RetinaNet mreže:

a) kriterijske funkcije, b) preciznosti.



Kako je već navedeno u potpoglavlju 4.2., RetinaNet ima odvojene podmreže za klasifikaciju i regresiju, pri čemu svaka zasebno ima vlastitu kriterijsku funkciju koje se sumiraju u konačan iznos kriterijske funkcije cijele mreže. Spomenute promjene kriterijskih funkcija obje mreže moguće je vidjeti na slici 4.18. Pritom, može se uočiti kako su krivulje podosta slične, što je najbolje uočljivo u epohama 14 i 16 kada obje podmreže zabilježavaju porast, odnosno pad, iznosa kriterijske funkcije.



a)

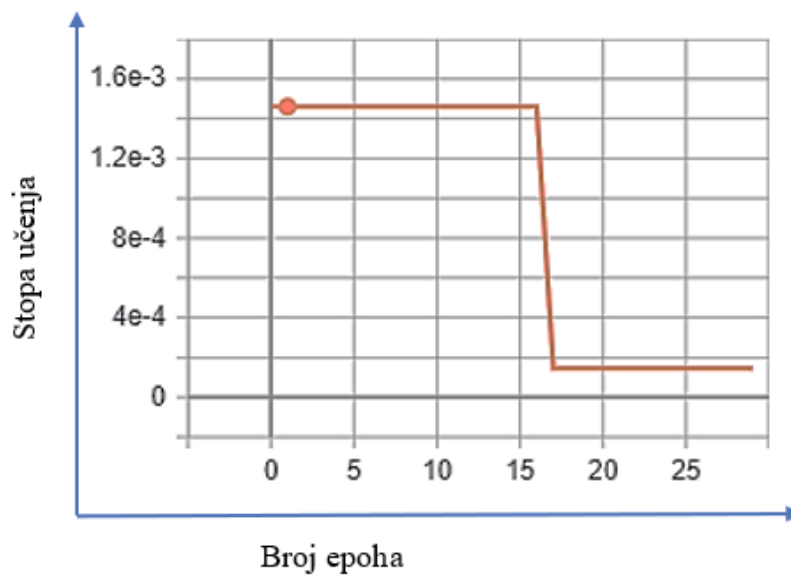


b)

**Slika 4.18.** Prikaz krivulja generiranih tijekom treninga RetinaNet mreže: a) kriterijske funkcije regresijske podmreže, b) klasifikacijske podmreže.

Kao što je već spomenuto, ponekad se zabilježeni porast iznosa kriterijske funkcije može popraviti smanjenjem stope učenja. Navedeno je najbolje vidljivo iz slike 4.19., gdje se u 16. epohi smanjuje stopa učenja na 0,000146 pozivom *callback* funkcije. Isto tako, može se uočiti da je pad

stope učenja u 16. epohi uzrokovao pad svih kriterijskih funkcija te porast preciznosti prepoznavanja na validacijskom skupu slika.



**Slika 4.19.** Prikaz promjene stope učenja tijekom treninga RetinaNet mreže.

Naposlijetku, nakon uspješno istreniranog modela RetinaNet mreže, ista će biti testirana na testnom skupu slika prometnih znakova, o čemu će detaljnije biti riječ u sljedećem poglavlju.

## 5. EVALUACIJA KONVOLUCIJSKIH MREŽA KORIŠTENIH ZA PREPOZNAVANJE PROMETNIH ZNAKOVA

Modeli mreže koji nastaju tijekom procesa treniranja sadrže informacije o dijelovima mreže koji su ključni za trening iste, poput regresijskih i konvolucijskih težina, kako je i navedeno u [42]. Međutim, ako se na slikama želi izvesti detekcija objekata i, uz to, spremi testna baza s prikazanom sigurnošću detekcije i rezultatima klasifikacije, potrebno je pretvoriti postojeći model u inferentni (engl. *inference model*) model. Navedeno je omogućeno u sklopu korištenog RetinaNet modela pozicioniranjem u odgovarajući direktorij i pokretanjem skripte *convert\_model.py* uz prilaganje potrebnih parametara, poput korištene okosnice, putanje do spremljenog treniranog modela i putanje na koju se želi spremi izrađeni inferentni model. Također, ako se prilikom treniranja koristila posebna konfiguracijska datoteka, istu je potrebno priložiti prilikom pretvorbe navedenog modela. Nadalje, nakon pretvorbe u inferentni model, potrebno je pokrenuti skriptu koja će evaluirati rad mreže na testnim slikama. Pritom, pruža se mogućnost za spremanjem testiranih slika s naznakom lokacije i klase objekata generiranima prilikom procesa testiranja. Osim toga, prilikom pokretanja skripte predaju se argumenti koji označavaju korištenu okosnicu, putanju do konfiguracijske datoteke, putanju do datoteke s popisom indeksa i naziva klasa, putanju do datoteke s podacima o testnoj bazi slika, kao i putanju do inferentnog modela stvorenog u prethodnom koraku. Navedena naredba za pokretanje skripte za evaluaciju modela nalazi se na slici 5.1.

```
python evaluate.py --backbone resnet50 --config putanja_do_konfiguracijske_datoteke_csv  
putanja_do_CSV_datoteke_za_testiranje putanja_do_CSV_datoteke_s_popisom_klasa  
putanja_do_h5_datoteke_inferentnog_modela
```

**Slika 5.1.** Naredba za pokretanje skripte za evaluaciju modela nad testnim skupom slika.

### 5.1. Testna baza slika

U svrhu evaluacije uspješnosti RetinaNet mreže, koristi se skup slika za testiranje. Podaci spomenutog skupa slika pohranjeni su u posebnoj CSV datoteci te je iste potrebno priložiti prilikom pokretanja procesa evaluacije modela.

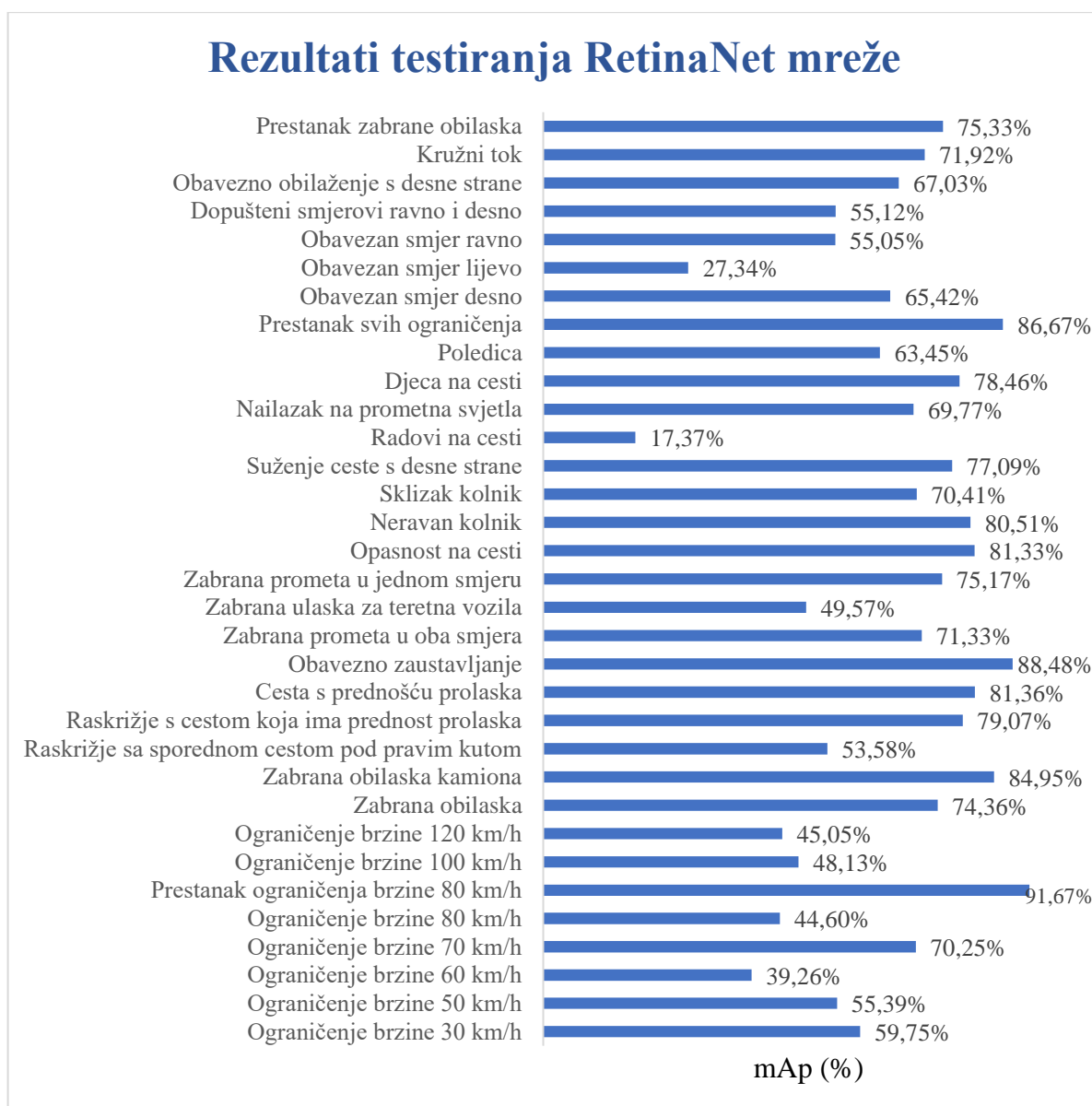
Testna baza broji ukupno 2 222 ručno označenih slika prometnih znakova. Prikaz broja uzoraka pojedinih prometnih znakova raspoređenih po klasama prikazan je u tablici 5.1.

**Tablica 5.1.** Broj uzoraka prometnih znakova za testiranje raspoređenih po klasama.

Naziv znaka	Broj uzoraka u bazi
Ograničenje brzine 30 km/h	82
Ograničenje brzine 50 km/h	141
Ograničenje brzine 60 km/h	65
Ograničenje brzine 70 km/h	226
Ograničenje brzine 80 km/h	88
Prestanak ograničenja brzine 80 km/h	6
Ograničenje brzine 100 km/h	54
Ograničenje brzine 120 km/h	21
Zabrana obilaska	58
Zabrana obilaska kamiona	15
Raskrižje sa sporednom cestom pod pravim kutom	28
Raskrižje s cestom koja ima prednost prolaska	276
Cesta s prednošću prolaska	309
Obavezno zaustavljanje	498
Zabrana prometa u oba smjera	52
Zabrana ulaska za teretna vozila	33
Zabrana prometa u jednom smjeru	94
Opasnost na cesti	60
Neravan kolnik	28
Sklizak kolnik	21
Suženje ceste s desne strane	13
Radovi na cesti	28
Nailazak na prometna svjetla	18
Djeca na cesti	80
Poledica	12
Prestanak svih ograničenja	9
Obavezan smjer desno	21
Obavezan smjer lijevo	8
Obavezan smjer ravno	31
Dopušteni smjerovi ravno i desno	28
Obavezno obilaženje s desne strane	252
Kružni tok	50
Prestanak zabrane obilaska	10
Negativni primjeri	257

## 5.2. Analiza rezultata testiranja RetinaNet konvolucijske neuronske mreže

U ovom potpoglavlju bit će analizirani rezultati dobiveni testiranjem RetinaNet mreže. Prag sigurnosti (engl. *score threshold*) postavljen je na 30 %, a isti određuje kolika mora biti minimalna sigurnost detekcije znaka da bi se ista ubrojila u znakove koji su ispravno prepoznati te bila grafički prikazana na slikama. Eksperimentalno je utvrđeno kako navedeni prag sigurnosti daje najpogodniji mAp na testnoj bazi slika. Rezultati testiranja RetinaNet mreže prikazani su na slici 5.2.



**Slika 5.2.** Rezultati lokalizacije i klasifikacije prometnih znakova korištenjem RetinaNet modela konvolucijske neuronske mreže.

Ukupan mAp izračunat na temelju mAp vrijednosti svih klasa navedenih u tablici 5.2. te uzimajući u obzir težinski prosjek istih po klasama iznosi 71,70 %. Prikaz nekih od ispravno označenih i klasificiranih znakova dan je na slikama 5.3., 5.4., 5.5.

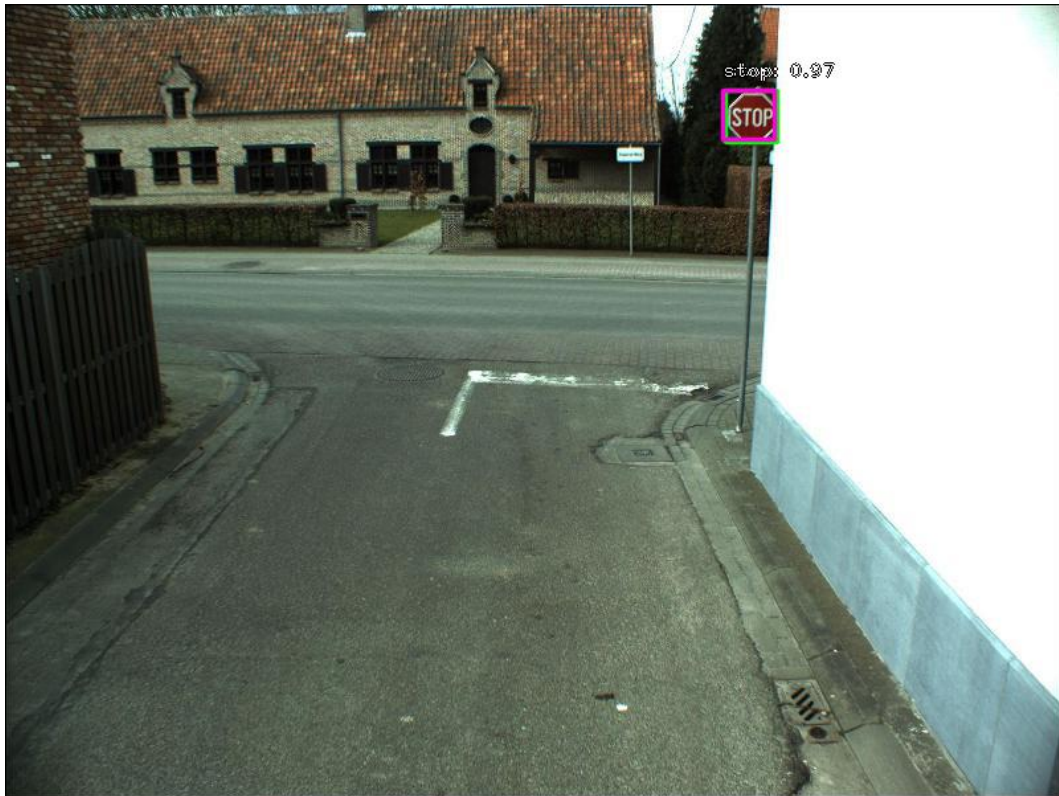


a)



b)

**Slika 5.3.** Prikaz ispravno lokaliziranih i klasificiranih prometnih znakova na primjeru prometnog znaka: a) obaveznog skretanje lijevo (preciznost: 1,00), b) raskrižja s cestom koja ima prednost prolaska (preciznost: 0,87).



a)



**Slika 5.4.** Prikaz ispravno lokaliziranih i klasificiranih prometnih znakova na primjeru prometnog znaka: a) obaveznog zaustavljanja (preciznost: 0,97), b) djece na cesti (preciznost: 1,00).



**Slika 5.5.** Prikaz ispravno lokaliziranih i klasificiranih prometnih znakova na primjeru prometnog znaka opasnosti na cesti (preciznost: 1,00).

U daljnjem razvoju teksta usporedit će se rezultati dobiveni u ovom radu sa rezultatima prepoznavanja prometnih znakova u [36], koji uz primjenu *RetinaNet* modela, iznosi preko 90 % preciznosti. Treba napomenuti kako je u [36] *RetinaNet* testirana na uniformnoj bazi njemačkih prometnih znakova, koja broji svega 600 slika za trening te 300 slika za testiranje. Iako je baza augmentirana, navedeni broj slika za 42 korištene klase ne čini dovoljno velik broj uzoraka prometnih znakova kako bi se moglo reći da je korišteni model dobro generaliziran za općeniti problem prepoznavanja prometnih znakova.

Suprotno tome, u istraživanju problema detekcije i klasifikacije prometnih znakova iz [44], uz korištenje *RetinaNet* modela te *ResNet-50* okosnice, postiže se mAp u iznosu od 79,39 %. Navedeni model treniran je i testiran na kineskoj bazi podataka (Tsinghua-Tencent 100K). Može se primijetiti kako je dobiveni rezultat u ovom slučaju znatno realniji. Razlog tomu je uporaba kineske baze podataka, koja slovi kao jedna od opsežnijih, ali i težih baza slika za učenje jer se sastoji od prometnih znakova, čije veličine u snimljenim slikama značajno variraju.

Iako su, inače, prometni znakovi poprilično standardizirani, postoje određene razlike pojedinih prometnih znakova u državama svijeta, što, svakako, dodatno otežava učenje modela na



određenim klasama. Tako na primjer, znak koji upozorava na rad na cesti razlikuje se bojom i/ili ikonom unutar znaka u Belgiji, Kini, Njemačkoj i Sloveniji (sl. 5.6.). Iako se razlike između navedenih slika ljudskom oku čine neznačajne, iste bi mogle uzrokovati veće probleme konvolucijskoj neuronskoj mreži prilikom klasifikacije određenog prometnog znaka. Razlog tome je neravnomjerna raspoređenost navedenih inačica znaka radova na cesti unutar klase u skupu slika za trening.



a)



b)



c)



d)

**Slika 5.6.** Prikaz znakova upozorenja o radovima na cesti u: a) Belgiji, b) Kini, c) Njemačkoj, d) Sloveniji.

Navedeno se može potvrditi i usporedbom s prometnim znakom upozorenja na neravan kolnik, koji su u Belgiji i Njemačkoj gotovo identični. Pritom, naziru se manje razlike u ikoni spomenutog prometnog znaka u Sloveniji (sl. 5.7.). Za navedenu klasu prikupljen je podjednak broj slika kao i za znak upozorenja o radovima na cesti. Štoviše, u testnom skupu slika isti brojevi jednaki broj uzoraka, ali, ipak, znakovi za radove na cesti postižu svega  $\approx 18\%$  mAp, što je u odnosu na znakove neravnog kolnika, koji postižu  $\approx 80\%$  mAp, znatno manje.



a)



b)



c)

**Slika 5.7.** Prikaz prometnih znakova upozorenja o neravnom kolniku u: a) Belgiji, b) Njemačkoj, c) Sloveniji.

Prosječno vrijeme za lokalizaciju i klasifikaciju prometnih znakova dobiveno u ovom radu iznosi 0,1157 sekundi po slici. Uspoređujući spomenuto vrijeme s [45], u kojem se, također, testira prepoznavanje prometnih znakova korištenjem RetinaNet modela, može se uvidjeti da je ovim radom postignuto brže izvođenje lokalizacije i klasifikacije objekata. Naime, prosječno vrijeme potrebno za obradu slike u [45] iznosi 0,15 sekundi, pri čemu se kao glavni resursi koriste se dvije Nvidia Tesla P100 grafičke kartice. Pritom treba uzeti u obzir kako je vrijeme obrade slika za lokalizaciju i klasifikaciju jedan od najvećih izazova u području prepoznavanja znakova za primjenu u aplikacijama u stvarnom vremenu.

Općenito, dobiveni rezultati mogli bi se poboljšati balansiranjem klasa prometnih znakova, koji se razlikuju u pojedinim državama. Nadalje, povećanjem broja negativnih primjera slika, koje mreža ne bi trebala identificirati, također, bi se mogli pospješiti rezultati rada RetinaNet mreže. Osim toga, poboljšanje preciznosti prepoznavanja znakova bi se, između ostalog, moglo omogućiti korištenjem složenijih okosnica, poput ResNet-101 te ResNet-152.

Konačno, uzimajući u obzir složenost korištene baze slika, u odnosu na tipične radove, koji svoje modele najčešće treniraju i testiraju na uniformnoj bazi slika jedne države, može se zaključiti da RetinaNet model dobro prepoznaje prometne znakove zadanih klasa.

## 6. ZAKLJUČAK

Razvoj autonomnih vozila jedna je od najkontroverznijih tema današnjice te se provode mnoga istraživanja kako bi isti što prije bili realizirani. Sustavi za prepoznavanje prometnih znakova jedni su od ključnih značajki koji predstavljaju glavnu odrednicu vremenskog razdoblja unutar kojeg će doći do pojave potpuno autonomnih vozila te integracije istih u svakodnevicu ljudi.

U ovom radu navedena su i opisana neka od najuspješnijih istraživanja povezanih sa sustavom za prepoznavanje prometnih znakova te je provedeno vlastito istraživanje primjene jedne od posljednjih tehnologija u području konvolucijskih neuronskih mreža na prilagođenoj bazi slika, koje sadrže standardne prometne znakove s međunarodnih prometnica. Za rješavanje problema prepoznavanja znakova korišten je RetinaNet model konvolucijske neuronske mreže. Rezultati testiranja pokazuju dobru lokalizaciju i prepoznavanje prometnih znakova, uzimajući u obzir složenost baze, koja je sastavljena od sedam različitih baza prometnih znakova. Također, navedeni rezultati uspoređeni su sa sličnim istraživanjima koji upotrebljavaju spomenutu tehnologiju te je zaključeno kako, usprkos napretku u području prepoznavanja znakova, još uvijek najviše problema stvaraju mali znakovi koje je teže lokalizirati i klasificirati. Osim toga, definirale su se mogućnosti za unaprjeđenjem rezultata dobivenih u ovom radu, koje se svode na dodatno proširivanje i balansiranje broja uzoraka prometnih znakova unutar pojedinih klasa te uporabu dubljih modela okosnica kao baza za RetinaNet konvolucijsku neuronsku mrežu.

U današnje vrijeme aktualna su mnoga istraživanja kojima se pokušava smanjiti računalna zauzetost sklopovlja uslijed korištenja strojnog učenja za prepoznavanje prometnih znakova. Sudeći prema povijesnom trendu razvoja računala, očekuje se smanjenje cijena i veličina sklopovlja boljih performansi. kako bi se algoritmi, koji prepoznavanje prometnih znakova dovode do samih maksimuma, mogli implementirati u autonomna vozila.

## LITERATURA

- [1] A. Ziebinski, R. Cuperk, D. Grzechca, L. Chruszczyk, Review of advanced driver assistance systems (ADAS), Proceedings of the International Conference of Computational Methods in Sciences and Engineering, Thessaloniki (AIP Conference Proceedings), br. 1906, sv. 1, str. (120002-1)–(120002-4), Thessaloniki Grčka, 2017. godina
- [2] S.P. Rajendran, L. Shine, R. Pradeep and S. Vijayaraghavan, Real-Time Traffic Sign Recognition using YOLOv3 based Detector, 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), sv. 1, str. 1-7, Kanpur, India, 2019. godina
- [3] S.B. Wali, M.A. Abdullah, M.A. Hannan, A. Hussain, S.A. Samad, P.J. Ker, M.B. Mansor, Vision-Based Traffic Sign Detection and Recognition Systems: Current Trends and Challenges, Sensors, br. 19, sv. 9, str. (2093-1)–(2093-28), svibanj 2019. godine
- [4] Y. Satılmış, F. Tufan, M. Şara, M. Karşlı, S. Eken, A. Sayar, CNN Based Traffic Sign Recognition for Mini Autonomous Vehicles, Advances in Intelligent Systems and Computing (Information Systems Architecture and Technology: Proceedings of 39th International Conference on Information Systems Architecture and Technology – ISAT 2018), br. 853, sv. 1, str. 85-94, Nysa, Poljska, 2018. godine
- [5] Z. Zhao, P. Zheng, S. Xu, X. Wu, Object Detection With Deep Learning: A Review, IEEE Transactions on Neural Networks and Learning Systems, br. 30, sv. 11, str. 3212-3232, studeni 2019. godine
- [6] U. Michelucci, Advanced Applied Deep Learning, Convolutional Neural Networks and Object Detection, Apress, New York, 2019. godine
- [7] C.G. Serna, Y. Ruichek, Classification of Traffic Signs: The European Dataset, IEEE Access, br.1, sv. 6, str. 78136-78148, prosinac 2018. godine
- [8] N.S. Artamonov, P. Y. Yakimov. Towards Real-Time Traffic Sign Recognition via YOLO on a Mobile GPU, The IV International Conference on Information Technology and Nanotechnology (Journal of Physics: Conference Series), br. 1096, sv.1, str. (1096-1)–(1096-8), Samara, Rusija, 2018. godine
- [9] A. Staravoitau, Traffic Sign Classification with a Convolutional Network, Pattern Recognition and Image Analysis, br. 28, sv. 1, str. 155–162, ožujak 2018. godine
- [10] J. Zhang, W. Wang, C. Liu, J. Wang, A. K. Sangaiah, Lightweight deep network for traffic sign classification, Annals of telecommunications, str. 1-11, srpanj 2019. godine

- [11] D. Tabernik, D. Skočaj, Deep Learning for Large-Scale Traffic-Sign Detection and Recognition, IEEE Transactions on Intelligent Transportation Systems, br. 21, sv. 4, str. 1427-1440, travanj 2020. godine
- [12] F. Shao, X. Wang, F. Meng, J. Zhu, D. Wang, D. Juying, Improved Faster R-CNN Traffic Sign Detection Based on a Second Region of Interest and Highly Possible Regions Proposal Network, Sensors, br. 19, sv. 10, str. (2288-1)–(2288-28), svibanj 2019. godine
- [13] Z. Liu, J. Du, F. Tian, J. Wen, MR-CNN: A Multi-Scale Region-Based Convolutional Neural Network for Small Traffic Sign Recognition, IEEE Access, br. 7, sv. 1, str. 57120-57128, travanj 2019. godine
- [14] A. Jose, H. Thodupunoori, B.B. Nair, A Novel Traffic Sign Recognition System Combining Viola–Jones Framework and Deep Learning, Soft Computing and Signal Processing, Advances in Intelligent Systems and Computing (Springer), br. 900, sv. 1, str. 507-517, siječanj 2019. godine
- [15] R. Ayachi, M. Afif, Y. Said, M. Atri, Traffic Signs Detection for Real-World Application of an Advanced Driving Assisting System Using Deep Learning, Neural Processing Letters, br. 51, sv. 1, str. 837–851, 2020. godine
- [16] J. Cao, C. Song, S. Peng, F. Xiao, S. Song, Improved Traffic Sign Detection and Recognition Algorithm for Intelligent Vehicles, Sensors (Basel), br. 19, sv. 18, str. (4021-1)–(4021-21), rujan 2019. godine
- [17] J. D. Kelleher, Convolutional and Recurrent Neural Networks, Deep Learning, The MIT Press, str. 159-184, Cambridge, Massachusetts, SAD, 2019. godine
- [18] W. H. Lopez Pinaya, S. Vieira, R. Garcia-Diaz, A. Mechelli, Convolutional neural networks, Machine Learning, Methods and Applications to Brain Disorder, Academic Press, str. 173-190, London, UK, 2019. godine
- [19] J. Folkens, AI in Automotive, Practical deep learning [online], Texas Instruments, Texas, 2018. godine, dostupno na:  
[https://e2e.ti.com/blogs\\_/b/behind\\_the\\_wheel/archive/2018/02/08/ai-in-automotive-practical-deep-learning](https://e2e.ti.com/blogs_/b/behind_the_wheel/archive/2018/02/08/ai-in-automotive-practical-deep-learning) [01.07.2020.]
- [20] S. Saha, A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way [online], Medium, San Francisco, 2018. godine, dostupno na:  
<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> [05.07.2020.]

- [21] H. H. Aghdam, E. J. Heravi, Guide to Convolutional Neural Networks, A Practical Application to Traffic-Sign Detection and Classification, Springer, Cham, Švicarska, 2017. godine
- [22] S. Sivarajkumar, ReLU – Most popular Activation Function for Deep Neural Networks [online], Medium, San Francisco, 2019. godine, dostupno na: <https://medium.com/@sonish.sivarajkumar/relu-most-popular-activation-function-for-deep-neural-networks-10160af37dda> [03.07.2020.]
- [23] S. Doshi, Convolutional Neural Network: Learn and Apply [online], Medium, San Francisco, 2019. godine, dostupno na: <https://medium.com/@sdoshi579/convolutional-neural-network-learn-and-apply-3dac9acfe2b6> [09.07.2020.]
- [24] S. Tammina, Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images, International Journal of Scientific and Research Publications (IJSRP), br. 9, sv. 10, str. 143-150, listopad 2019. godine
- [25] F. E. Thorsson Schöller, M. K. Plenge-Feidenhans'l, J. D. Stets, M. Blanke, Assessing Deep-learning Methods for Object Detection at Sea from LWIR Images, IFAC-PapersOnLine, br. 52, sv. 21, str. 64-71, rujan 2019. godine
- [26] T. M. Hoang, P. H. Nguyen, N. Q. Truong, Y. W. Lee, K. R. Park, Deep RetinaNet-Based Detection and Classification of Road Markings by Visible Light Camera Sensors, Sensors br. 19, sv. 2, str. (281-1)–(281-25), siječanj 2019. godine
- [27] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature Pyramid Networks for Object Detection, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), sv. 1, str. 936-944, Honolulu, HI, 2017 godine
- [28] Tzutalin, LabelImg [online], Git code, 2015. godine, dostupno na: <https://github.com/tzutalin/labelImg> [08.06.2020.]
- [29] R. Timofte, K. Zimmermann, L. van Gool, Multi-view traffic sign detection, recognition, and 3D localisation, Journal of Machine Vision and Applications, Springer, br. 25, sv. 3, str. 633-647, travanj 2014. godine
- [30] A. Mogelmose, M. M. Trivedi, T. B. Moeslund, Vision based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey, IEEE Transactions on Intelligent Transportation Systems, br. 13, sv. 4, str. 1484-1497, prosinac, 2012. godine

- [31] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, C. Igel, Detection of traffic signs in real-world images: The German traffic sign detection benchmark, The 2013 International Joint Conference on Neural Networks (IJCNN), sv.1, str. 1-8, Dallas, TX, 2013. godine
- [32] F. Larsson, M. Felsberg, Using Fourier Descriptors and Spatial Models for Traffic Sign Recognition , Proceedings of the 17th Scandinavian Conference on Image Analysis, SCIA, br. 6688, sv.1, str. 238-249, Ystad, Švedska, 2011. godine
- [33] F. Larsson, M. Felsberg, P. E. Forssen , Correlating Fourier descriptors of local patches for road sign recognition , IET Computer Vision, br. 5, sv. 4, str. 244-254, 2011. godine
- [34] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, The MIT Press, SAD, 2016. godine
- [35] A. B.Jung et al., ImgAug [online], GitHub Inc., 2020. godina, dostupno na: <https://github.com/aleju/imgaug> [04.07.2020.]
- [36] S. P. Rajendran, L. Shine, R. Pradeep, S. Vijayaraghavan, Fast and Accurate Traffic Sign Recognition for Self Driving Cars using RetinaNet based Detector, 2019 International Conference on Communication and Electronics Systems (ICCES), br. 1, sv. 1, str. 784-790, Coimbatore, India, 2019. godine
- [37] T. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal Loss for Dense Object Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, br. 42, sv. 2, str. 318-327, veljača 2020. godine
- [38] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature Pyramid Networks for Object Detection, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), br. 1, sv. 2, str. 936-944, Honolulu, HI, 2017. godine
- [39] Y. Guo, H. Shi, A. Kumar, K. Grauman, T. Rosing, R. Feris, SpotTune, Transfer Learning Through Adaptive Fine-Tuning, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), str. 4800-4809, Long Beach, CA, SAD, 2019. godine
- [40] A. Soni, Fine Tuning & Feature Engineering [online], Medium, San Francisco, 2018. godine, dostupno na: <https://medium.com/@soniaditya922/fine-tuning-feature-engineering-eb57b7d30ce8> [17.06.2020.]
- [41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research, br. 15, sv. 1, str. 1929-1958, lipanj 2014. godine
- [42] Fizyr, Keras-retinanet [online], GitHub Inc., 2019. godine, dostupno na: <https://github.com/fizyr/keras-retinanet> [15.06.2020]
- [43] M. Zlocha, Q. Dou, B. Glocker, Anchor-optimization [online], GitHub Inc., 2020. godine,

dostupno na: <https://github.com/martinzlocha/anchor-optimization> [15.06.2020.]

[44] Jiahao CUI, Simultaneous Traffic Sign Detection and Classification with RetinaNet [online], GitHub Inc., 2019. godine,

dostupno na: <https://github.com/CJHMPower/Simultaneous-Traffic-Sign-Detection-and-Classification-with-RetinaNet> [28.06.2020.]

[45] M. Zhu, J. Hu, Z. Pu, Z. Cui, L. Yan, Y. Wang, Traffic Sign Detection and Recognition for Autonomous Driving in Virtual Simulation Environment, ArXiv, preprint (abs/1911.05626), listopad 2019. godine



## SAŽETAK

Prepoznavanje prometnih znakova jedna je od ključnih odrednica potrebnih za realizaciju autonomnih vozila. U ovom radu evaluiran je rad konvolucijske neuronske mreže na problemu lokalizacije i klasifikacije prometnih znakova korištenjem prilagođene baze podataka. Kako bi se navedeni problem uspješno riješio, implementirana je RetinaNat konvolucijska neuronska mreža. Prometni znakovi koji se trebaju prepoznati podijeljeni su u 33 klase, a za automatsku augmentaciju slika i prilagodbu graničnih okvira unutar istih korištena je *ImgAug* biblioteka. Model RetinaNet mreže implementiran je u *python* skriptnom jeziku korištenjem *Keras* API-ja. Algoritam je testiran na posebnoj bazi slika te se, uzimajući u obzir složenost baze podataka, utvrdilo kako predloženi model mreže dobro prepoznaje prometne znakove postižući mAp u iznosu 71,70 %. Pritom ističu se prometni znakovi poput obaveznog zaustavljanja koji postižu gotovo 90 % mAp. Nasuprot tome, pojavljuju se prometni znakovi sa značajno nižom mAp. Uspješnost prepoznavanja prometnih znakova potencijalno bi se mogla poboljšati dodatnim balansiranjem pojedinih klasa baze podataka te korištenjem dubljih konvolucijskih neuronskih mreža kao okosnica za RetinaNet model.

**Ključne riječi:** automatska augmentacija, *Keras*, klasifikacija prometnih znakova, RetinaNet, sustavi za autonomna vozila

## ABSTRACT

### Traffic sign classification using convolutional neural networks

Traffic sign recognition is one of the key joints needed for realization of autonomous driving. This master's thesis evaluates the problems of localization and classification of traffic signs using convolutional neural network and custom made traffic sign database. As a part of this thesis, a RetinaNet convolutional neural network was implemented. There are 33 classes of traffic signs that need to be recognized. Also, an automatic augmentation was performed over the database using *ImgAug* library, whereas RetinaNet model was implemented using *Python* and *Keras* API. The algorithm was tested on separated test database. Considering complexity of the created database, it was established, that in most cases traffic signs are correctly recognized. The overall mAP of 71,70 % was achieved. There were traffic sign classes, such as stop, that achieved mAp of almost 90 %. On the contrary, there were classes that achieved significantly low mAp. However, the used model could reach better performance by balancing data in several classes of existing database with new images, as well as, using deeper backbones.

**Keywords:** automatic augmentation, *Keras*, traffic sign classification, RetinaNet, autonomous vehicles' systems

## ŽIVOTOPIS

Matoš, Ivona rođena je 27.08.1996. u Zagrebu. Nakon završene Sedme Gimnazije u Zagrebu, 2015. godine, upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Godine 2018. stječe akademski naziv sveučilišni prvostupnik inženjer računarstva te upisuje sveučilišni diplomski studij Automobilskog računarstva i komunikacija. Godinu dana kasnije sudjeluje na IWSSIP konferenciji u Osijeku, na kojoj uspješno prezentira prvi objavljeni znanstveni rad temeljen na dvogodišnjem proučavanju prepoznavanja prometnih znakova. Osim toga, dobitnica je nagrade za uspješnost u studiranju 2020. godine.

---

Ivona Matoš

## PRILOZI

- P.4.1. Baza podataka trening i validacijskog skupa slika (uključujući pripadajuće *CSV* datoteke s informacijama o navedenim skupovima slika), *python* skripte za mijenjanje veličine slika i augmentaciju podataka
- P.4.4. Konfiguracijska datoteka korištena prilikom treninga RetinaNet mreže te izmijenjena *python* skripta RetinaNet arhitekture
- P.5.1. Baza podataka testnog skupa slika (uključujući pripadajuću *CSV* datoteku s informacijama o testnom skupu slika)