



Univerzitet u Beogradu - Elektrotehnički fakultet

Katedra za signale i sisteme



## **DIPLOMSKI RAD**

# **Detekcija automobila korišćenjem neuralnih mreža i histograma boja**

### **Kandidat**

Jakša Jovičić, 2016/0384

### **Mentor**

dr Željko Đurović, redovni profesor

Beograd, *jul* 2020. godine

## REZIME RADA

Detekcija automobila na slici je vrlo zahtevan i bitan problem, danas se za njegovo rešavanje najčešće koriste metode mašinskog učenja i veštačke inteligencije. Ovaj diplomski rad pokušava taj problem da reši konkretno korišćenjem klasičnih neuralnih mreža kao klasifikatora i tronivoskih hitograma boja kao obeležja za prepoznavanje. Model je treniran na združenom skupu podataka iz tri odvojena skupa podataka sa interneta i daje vrlo dobre rezultate kada samo treba sliku da klasifikuje da li je data slika automobil ili ne. Problem nastaje kod detekcije, korišćen je klizeći prozor različitih dimenzija koji prolazi kroz celu sliku i daje svoju procenu za sigurnost da je trenutni prozor automobil za svaki prozor tako da može da se desi da prozor "promaši" automobil zbog činjenice da je veoma računarski zahtevno prozor pomerati piksel po piksel pa se dodaje određeni preskok pri svakom pomeraju. Takođe još jedan problem je to što je model treniran na slikama celih automobila a pri prolasku kroz sliku može da se desi da prozor samo uhvati dobar deo automobila ali nedovoljno za detekciju. Ti problemi uz glavni problem ovakvog pristupa a to je nedostatak prepoznavanja morfoloških karakteristika automobila dovode do toga da rezultati nisu na nivou na kakvom bi bili korišćenjem konvolucionih neuralnih mreža na celu sliku.

## ZAHVALNICA

Izražavam zahvalnost svojim mentorima u izradi diplomskog rada:

- Profesoru Željku Đuroviću, na pomoći u izradi diplomskog rada i na svim nezaboravnim predavanjima tokom studija.
- Profesoru Bogdanu Pavkoviću, na nezamenjivom praktičnom znanju koje sam stekao tokom stipendijskog programa pod njegovim mentorstvom, kao i na velikom strpljenju i predusretljivosti.

## SADRŽAJ

REZIME RADA .....	2
ZAHVALNICA .....	3
SADRŽAJ .....	4
1 UVOD .....	5
2 METODOLOGIJA RADA .....	6
2.1 Teorijski uvod u Neuralne Mreže i njihovo korišćenje u klasifikaciji .....	6
2.2 Teorijski uvod u osnove obrade slike i korišćenje histograma boja .....	10
2.3 Prikupljanje podataka .....	12
2.4 Treniranje modela .....	15
2.5 Pretraga slike kliznim prozorom .....	19
3 REZULTATI .....	25
4 DISKUSIJA .....	30
5 ZAKLJUČAK .....	32
6 LITERATURA .....	33

## 1 UVOD

Problem kojim se bavi ovaj diplomski rad je automatska detekcija automobila na slici dobijenoj sa kamere koja se nalazi u automobilu i snima put ispred sebe. Ovaj problem je vrlo zahtevan i do sada je najbolje rešen primenom konvolucionih neuralnih mreža i *YOLO* (eng. *You Only Look Once*) metode, dok ovaj rad pokušava isti problem da reši na drugi način: Prolaskom kroz datu sliku sa više prozora raznih veličina i izdvajanjem obeležja iz slike u vidu tronivoskih histograma boje i klasifikacija trenutnog prozora korišćenjem prethodno treniranje neuralne mreže. Nakon prolaza kroz celu sliku traže se maksimumi sigurnosti da je automobil u određenom prozoru i oni se vraćaju kao rezultat algoritma.

Cilj rada je napraviti sto bolji model za klasifikaciju kao i sto bolji algoritam za primenu tog modela na celoj slici, i ukoliko to ne ometa rad sistema napraviti da on bude što vremenski efikasniji.

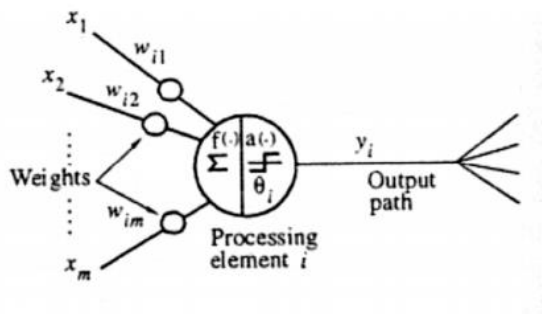
Prvo poglavlje se bavi teorijskim uvodom u pojam veštačkih neuralnih mreža i njihovim korišćenjem u klasifikaciji. Drugo poglavlje objašnjava algoritam koji je korišćen za pretragu slike. Treće i četvrto poglavlje objašnjavaju postupak prikupljanja podataka i samog obučavanja mreže. U petom poglavlju opisan je prvo opstupak prolaska kroz sliku prozorom jedne veličine a onda to preneseno na više prozora. U nastavku rada su rezultati i kratka diskusija i zaključak.

## 2 METODOLOGIJA RADA

### 2.1 Teorijski uvod u Neuralne Mreže i njihovo korišćenje u klasifikaciji

Neuralna mreža je struktura napravljena od više povezanih slojeva veštačkih Neurona, oni su napravljeni tako da imitiraju ljudski ili životinjski mozak. U ovom radu korišćena je samo osnovna struktura neuralne mreže za klasifikaciju već izdvojenih obeležja ali postoje i komplikovanije strukture koje su pravljene za rešavanje nekih specifičnih problema.

Osnovna jedinica građe nuralne mreže je neuron sa svojim ulazima i izlazima, njegova jedina funkcija je da primi ulaze, sa njima uradi određenu računsku operaciju i napravi izlaz. Funkcije koje obavlja neuron mogu biti razne (linearna suma, kvadratna funkcija...) ali se najčešće koristi jednostavna težinjena suma ulaza. Pored standardnih ulaza neuron ima i specijalni ulaz koji se zove pomeraj, on je uvek jednak minus jedan a njegov težinski faktor kontroliše koliko on učestvuje u računanju i on služi da se suma ulaza centrira oko nule da bi aktivaciona funkcija uvek bila ista.

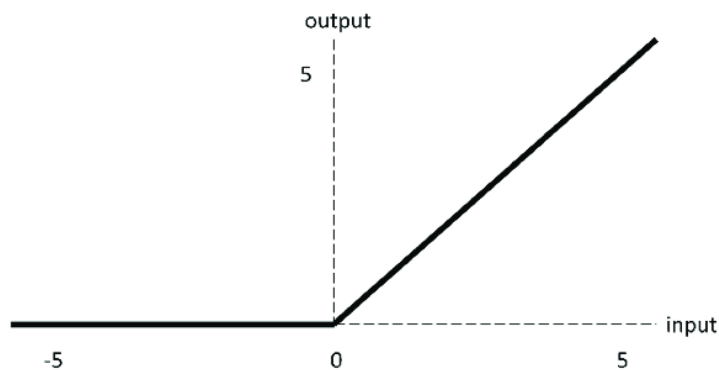


Slika 1 . Veštački neuron. Slika je preuzeta iz [1].

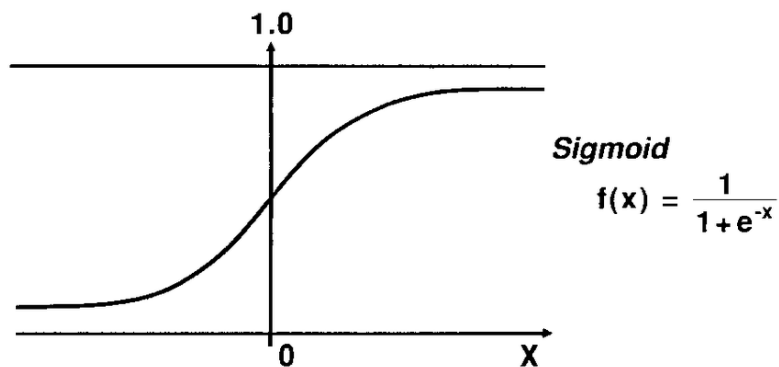
$$f_i \triangleq \text{net}_i = \sum_{j=1}^m w_{ij} x_j - \theta_i$$

Slika 2 . Prenosna funkcija veštačkog neurona. Slika je preuzeta iz [1].

Aktivacione funkcije neurona određuju način funkcionisanja neuralne mreže, najčeće korišćene su step, sgn, ReLU, sigmoid... U ovom radu korišćene su funkcije ReLU - Rectified Linear Unit (za ulazni i skriveni sloj) i sigmoid (za izlazni sloj). Slede grafici ovih aktivacionih funkcija:



Slika 3. ReLU aktivaciona funkcija veštačkog neurona.

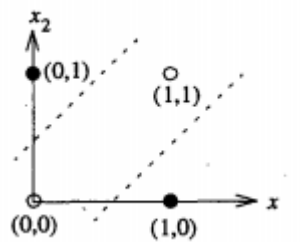


Slika 4. Sigmoid aktivaciona funkcija veštačkog neurona.

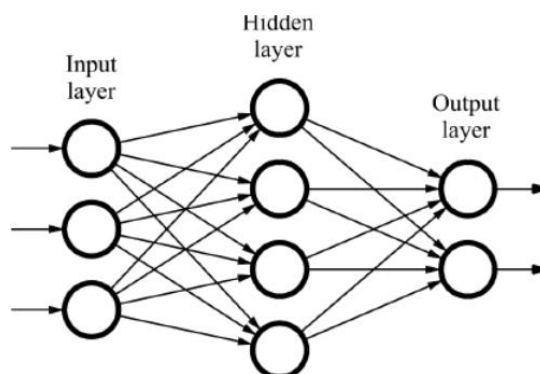
Za izlazni sloj korišćena je sigmoid funkcija jer ona daje rezultate u opsegu od nula do jedan pa se lako jednim pragom koristi za klasifikaciju: Ako je vrednost iznad praga ulaz pripada datoj klasi a ako je ispod ne.

Jedan neuron u mreži predstavlja samo objekat koji vrši određenu računsku operaciju nad ulazima koji su mu dati. Informacije u neuralnoj mreži čuvaju se u težinskim koeficijentima veza između neurona i celih slojeva.

Neuralna mreža je organizovana u slojeve, slojevi neuralne mreže mogu biti ulazni (input), skriveni (hidden) i izlazni (output). Neuroni u ulaznom sloju kao svoje ulaze direktno primaju ulaz u mrežu i svoje izlaze prosleđuju na skrivene slojeve. Broj skrivenih slojeva i broj neurona u svakom sloju zavisi od strukture problema, premalo neurona može da dovede do toga da mreža ne uspeva da isprati strukturu problema. Na primeru sledeće slike vidi se da nije moguće napraviti dobar klasifikator korišćenjem jedne prave, dakle problem je taj koji diktira strukturu mreže.



Slika 5. Primer neadekvatne strukture klasifikatora. Slika je preuzeta iz [1].

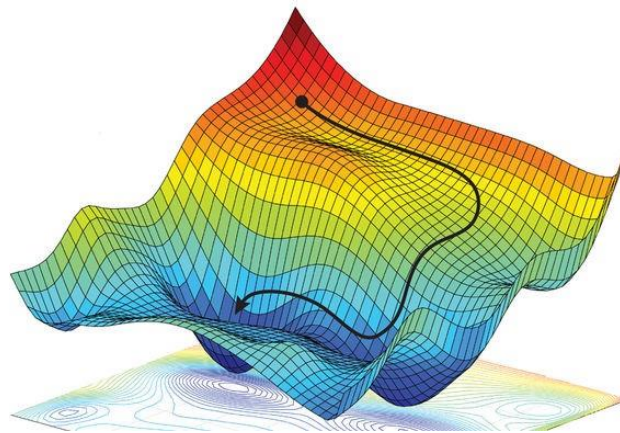


Slika 6. Osnovna struktura veštačke neuralne mreže.

Neuralna mreža se koristi za klasifikaciju i modelovanje tako što koristi koncepte mašinskog učenja, u ovom slučaju učenje sa nadgledanjem (eng. *supervised learning*). Neuralna mreža se obučava tako što joj se daje veliki broj primera za koje je poznato kojoj klasi pripadaju. Težine svih veza su na početku inicijalizovane na nasumične vrednosti, mreža na svom ulazu dobija jedan primer, računa vrednosti svih slojeva do kraja i dobija nekakav izlaz, onda se iz dobijenog i tačanog izlaza računa kriterijumska funkcija. Primenjuje se algoritam propagiranja greške unazad (backpropagation algorithm) i težinski koeficijenti u celoj mreži menjaju tako da se ta kriterijumska funkcija minimizuje. Neuralne mreže su u opštem slučaju mnogodimenzionalni sistemi i nije moguće vizualizovati vrednosti kriterijumske funkcije za sve moguće vrednosti ulaza. Nalbolja analogija za ovaj algoritam je čovek koji silazi niz planinu ali da može da se vidi samo metar oko sebe tako da ne može da



zna koji mu je optimalan put već samo može da vidi u kom smeru treba da napravi korak da bi se spustio što niže.

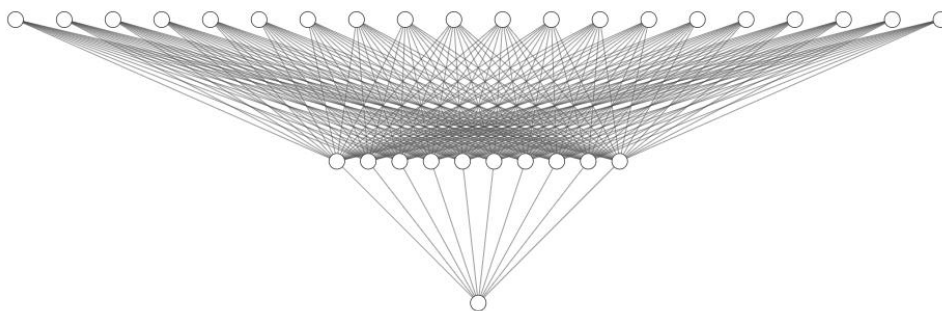


*Slika 7. Pojednostavljen primer minimizacije kriterijumske funkcije.*

Ovakav proces propagacije ulaza do kraja mreže i onda propagacija greške unazad i ažuriranje koeficijenata ponavlja se više puta za svaki od ulaza sve dok neki od kriterijuma za završetak ne bude ispunjen (broj epoha, vreme, greška ispod neke granice...).

Dakle za korišćenje neuralne mreže za klasifikaciju potrebna nam je baza podataka za trening i to u obliku ulaznih vektora brojeva i jednog izlaza koji označava da li data slika pripada klasi koju želimo ili ne. Sledeće poglavlje se bavi dobijanjem gotovih vektora brojeva spremnih za neuralnu mrežu iz baze slika.

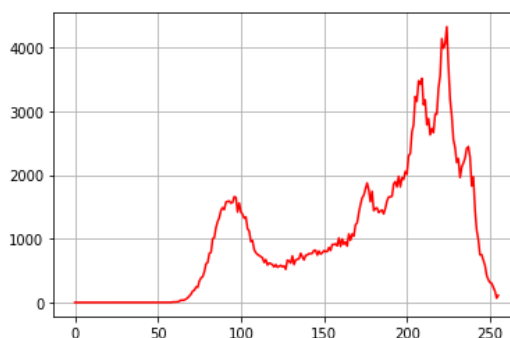
Struktura neuralne mreže koja je korišćena prikazana je na sledećoj slici, 192 ulaza (kasnije objašnjeno) su povezani sa svakim od 20 neurona u prvom sloju i sve tako do izlaza gde je samo jedan neuron. Ova mreža ima ukupno:  $192 \times 20 + 20 \times 10 + 10 = 4050$  parametara za obočavanje od težinskih koeficijenata veza i još  $20 + 10 + 1 = 31$  parametar od koeficijenata pomeraja. Dakle mreža ima 4081 parametar za obučavanje.



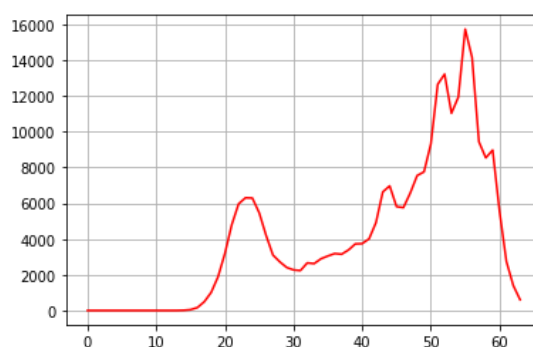
*Slika 8. Struktura korišćene neuralne mreže.*

## 2.2 Teorijski uvod u osnove obrade slike i korišćenje histograma boja

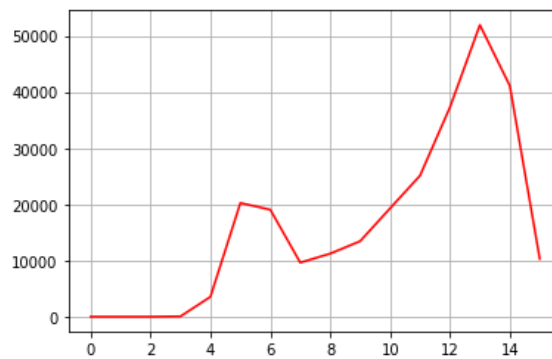
Ovaj rad se bavi klasifikacijom slika u boji, one se u računar predstavljaaju kao tri matrice koje predstavljaju nivoe intenziteta crvene, zelene i plave boje. Dakle slika dimenzije  $N \times M$  ima  $3 \times N \times M$  bita sto je u opštem slučaju previše podataka da bi se koristilo kao ulaz neuralne mreže kao takvo. Potrebno je smanjiti dimenzionalnost tih podataka i što veću količinu informacija iz originalne slike prikazati sa što manje brojeva. Ovaj rad za klasifikaciju koristi histograme boja. Slika u boji ima tri histograma boje za svaku od tri boje koje se koriste za reprezentaciju, histogram je niz brojeva koji predstavljaju broj piksela u slici koji imaju određeno osvetljenje. Pošto se za reprezentaciju slike koriste bitovi, svaki piksel može imati intenzitet boje od 0 do  $2^8 = 256$  pa tako tri histograma boja imaju  $3 \times 256 = 768$  brojeva koji mogu biti i veći od bita. To je već broj koji bi mogao da se koristi za neuralnu mrežu iz razloga sto je konstantan i svaka slika u boji ima toliko obeležja nevezano za njenu dimenziju.



Slika 9. Histogram crvene boje slike bez grupisanja nivoa.

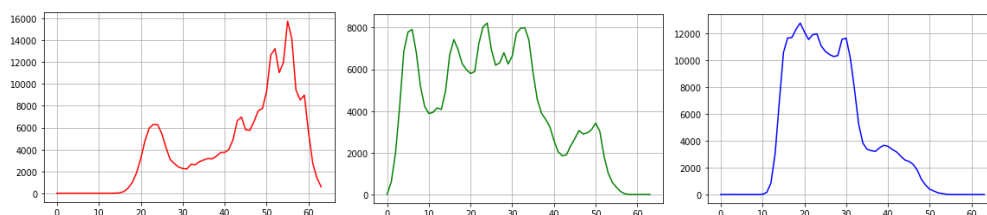
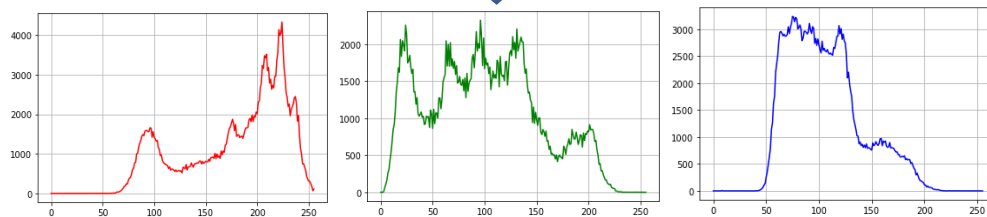
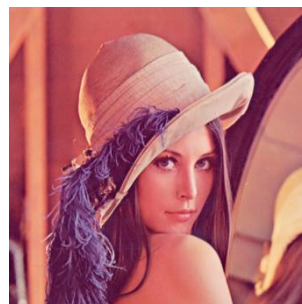


Slika 10. Histogram crvene boje slike grupisan u 64 grupe.



Slika 11. Histogram crvene boje slike grupisan u 16 grupa.

U ovom radu taj broj je još malo smanjen tako sto je histogram svake boje grupisan u pregrade koje predstavljaju malo duži niz boja tako da nemamo po broj za svaki intenzitet boje već se brojevi piksela svake boje iz opsega sabiraju u jedan broj. Na slikama se vidi efekat kada je histogram grupisan u 64 grupe i 16 grupa. Izabrano je grupisanje u 64 grupe jer ono dovoljno dobro prenosi informaciju a značajno je smanjen broj bitova. Tako da se svaka slika predstavlja nizom od  $3 \times 64 = 192$  broja.



Slika 12. Proces dobijanja gotovih obeležja od slike.

## 2.3 Prikupljanje podataka

Za treniranje neuralne mreže potreban je što veći i što raznovrsniji obeležen skup podataka. U ovom radu korišćena su tri odvojena skupa podataka preuzeta sa interneta i dodatan skup koji je dobijen slikanjem saobraćaja mobilnim telefonom.

Skup podataka 1 - [CompCars dataset \[6\]](#): Ovaj skup sadrži preko 100000 slika celih automobila i njihovih delova sa parovima koordinata koje određuju gde se automobil nalazi za svaku sliku. Pored toga svaka slika je obeležena oznakom perspektive iz koje je slikan automobil (1 – napred, 2 – pozadi, 3 – sa strane, 4 – napred ukoso, 5 – pozadi ukoso). Eksperimentalno je zaključeno da sistem radi dosta bolje kada se za treniranje koriste samo slike automobila označene sa 1 i 2 što je i u redu jer su i slike iz drugih skupova iz tih perspektiva pa je time funkcionalnost sistema ograničena samo na to prepoznavanje. Slede primeri slika iz ovog skupa:



Slika 13. Primeri slika iz skupa 1.

Skup podataka 2 – [Motion-based Segmentation and Recognition Dataset \[8\]](#): Ovaj dataset je prvobitno namenjen za semantičku segmentaciju slike tako da nema obeležene klase na slikama. Na ovom sajtu mogu se naći tri duga snimka napravljena kamerom u toku vožnje, u ovom radu oni su korišćeni tako što su iz njih izvučeni prozori i iz njih delovi slike pa su ti delovi korišćeni za treniranje kao nije-automobil klasa. Delovi ovih snimaka koji nisu korišćeni za treniranje su korišćeni za testiranje sistema. Sledi primer prozora iz jednog od snimaka:



Slika 14. Primer prozora iz kog su izvučene slike za trening.

Iz ovakve slike izvučene su slike veličina onih prozora koji će se koristiti u pretrazi slike i takve podslike su korišćene za trening kao klasa nije-automobil. Primeri takvih slika:



Slika 15. Primeri podslika iz skupa 2 koje su korišćene za trening.

Skup podataka 3 – [Vehicle Image Database \[7\]](#): Ovaj skup sadrži oko 3000 slika automobila dobijenih iz raznih video snimaka sa interneta, ali i slika klase nije-automobil. Pošto su ove slike već u dobrom formatu, isečeni su samo automobili, nije potrebno raditi nikakvu obradu nad ovim skupom. Slike su u formatu  $64 \times 64$  tako da im je rezolucija dosta lošija od ostalih skupova ali to može dobro da doprinese sistemu da bolje klasifikuje

automobile koji su daleko. Slike su i ovde podeljene u 4 različite perspektive: Far, Left, MiddleClose, Right.



*Slika 16. Primeri podslika iz skupa 3 koje su korišćene za trening redom Far, Left, MiddleClose, Right.*

Skup podataka 4 – Dobijen slikanjem saobraćaja u toku vožnje: Ovaj skup nije upload-ovan na internet ali nije ni neophodan za pravilno funkcionisanje sistema. Iz ovog skupa je korišćeno oko 100 slika za klasu automobil i oko 500 slika za klasu nije-automobil. Ove slike su pažljivo izabrane i dosta su bolje rezolucije nego slike iz drugih skupova, povećanje broja ovakvih slika bi verovatno poboljšalo sistem u svakom smislu:



*Slika 17. Primeri slika iz skupa 4 koje su korišćene za trening.*

Modeli zasnovani na mašinskom učenju povećavaju svoju preciznost i korisnost sa povećanjem skupova za treniranje i povećanjem kvaliteta tih skupova tako da se rezultati ovog rada uvek mogu poboljšati na taj način.

## 2.4 Treniranje modela

Ukupno je iz svih skupova podataka dobijeno 23830 slika klase nije-automobil i 12190 slika klase automobil, što je dobra količina podataka za trening. Pri eksperimentalnom isprobavanju raznih arhitektura i parametara tih 36020 slika se deli na trening i test skup u odnosu 80:20 i model se trenira samo na trening skupu a test skup se koristi za proveru kvaliteta rada sistema. Nakon utvrđivanja konačne strukture mreže svih 100% podataka se koristi za trening. Ukupan skup podataka se često deli na još jedan skup – validacioni skup, on se izdvaja iz test skupa i služi da se u svakoj epohi proverava stanje modela na podacima koji nisu korišćeni za samo obučavanje.

Kriterijumska funkcija je ona funkcija koja će se minimizovati u toku obučavanja mreže, ona je uglavnom funkcija željenog izlaza i tačnog izlaza koji je poznat iz trening podataka. Izbor kriterijumske funkcije koja će se koristiti pri obučavanju zavisi od strukture problema i umnogome određuje rad modela. Osnovni oblik kriterijumske funkcije je MSE (eng. *Mean Squared Error*) ali pored njega postoji dosta funkcija koja su pravljene za specifične primene, na primer za probleme klasifikacije najčešće se koristi *Cross-Entropy* funkcija. U ovom modelu s obzirom na to da imamo samo dve klase koristi se funkcija *Binary Cross-Entropy*. Slede formule nekih kriterijumskih funkcija.

$$MSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}$$

Slika 18. MSE kriterijumska funkcija.

$$CE = - \sum_i^C t_i \log(s_i)$$

Slika 19. Cross-Entropy kriterijumska funkcija.

$$CE = - \sum_{i=1}^{C'=2} t_i \log(s_i) = -t_1 \log(s_1) - (1 - t_1) \log(1 - s_1)$$

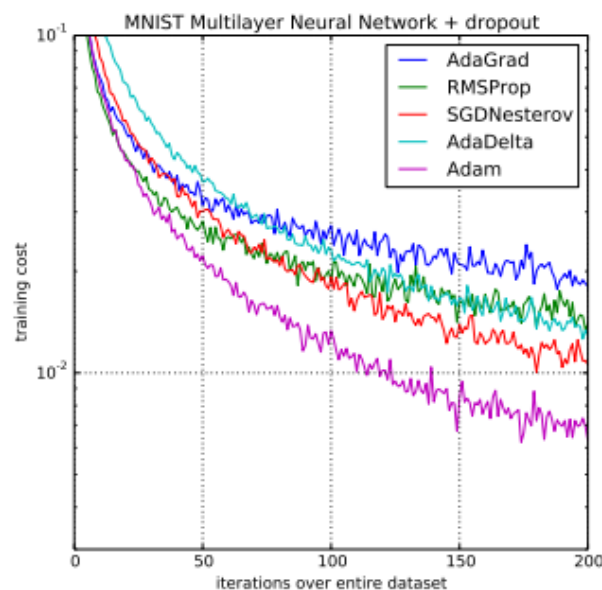
Slika 20. Binary Cross-Entropy kriterijumska funkcija.

Treniranje neuralne mreže radi se u epohama, jedna epoha odgovara jednom prolasku kroz sve trening primere. Dodatno, epoha se izvršava u delovima, pri prolasku kroz trening primere parametri mreže se ne ažuriraju na svakom primeru već se primeri grupišu u serije (eng. *batch*) i tek na kraju kraju serije se na osnovu ukupne greške menjaju parametri mreže.



Ovim se postiže dosta manji broj računskih operacija ali može da dođe do manje preciznog konvergiranja modela. Broj primera u seriji se najčešće stavlja da bude umnožak broja dva, u ovom slučaju u jednoj seriji nalazi se 128 trening primera.

Postoji više različitih algoritama za optimizaciju obučavanja neuralne mreže (*Momentum*, *Nesterov accelerated gradient*, *Adam*, *Adagrad*, *RMSprop*...). Osnovni algoritam se zove *SGD* (eng. *Stochastic Gradient Descent*) koji ažurira prametre mreže u smeru u kom se nalazi trenutno najbolje sledeće rešenje, ovaj algoritam bez dodatnih unapređenja nailazi na probleme zbog lokalnih minimuma kriterijumske funkcije, sedlastih tačaka, različitih osetljivosti promena različitih parametara... Ovi problem se rešavaju unapređenjima osnovnog algoritma: *AdaGrad* (eng. *Adaptive Gradient Algorithm*) uvodi promenljivu konstantu obučavanja kroz vreme i time postiže manju zavisnost od osetljivosti promene pojedinačnih parametara ali ne rešava problem sedlastih tačaka. *RMSProp* (eng. *Root Mean Square Propagation*) koji pored promenljive konstante obučavanja ima i konstantu promene za svaku promenljivu i time ne dozvoljava velike preskoke pri traženju minimuma. U ovom radu korišćen je *Adam* algoritam, on kombinuje dve bitne nadogradnje iz *AdaGrad* i *RMSProp* i eksperimentalno se pokazuje da radi bolje od oba algoritma.

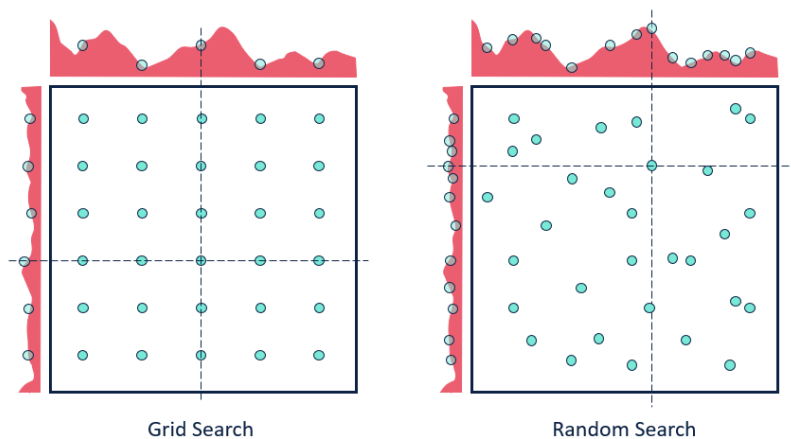


Slika 21. Poređenje algoritama optimizacije. Preuzeto iz [10].

Svi parametri neuralne mreže koji se ne obučavaju zovu se hiperparametri, tu spadaju: arhitektura mreže (broj slojeva, broj neurona u slojevima, aktivacione funkcije...), kriterijumska funkcija, algoritam optimizacije, broj epoha, veličina serije. Pronalazak najboljih hiperparametara nije lak zadatak jer ne postoji algoritam kojim jednoznačno dolazimo do dobrih rezultata, često popravljanjem jednog parametra štetimo nekom drugom pa je taj

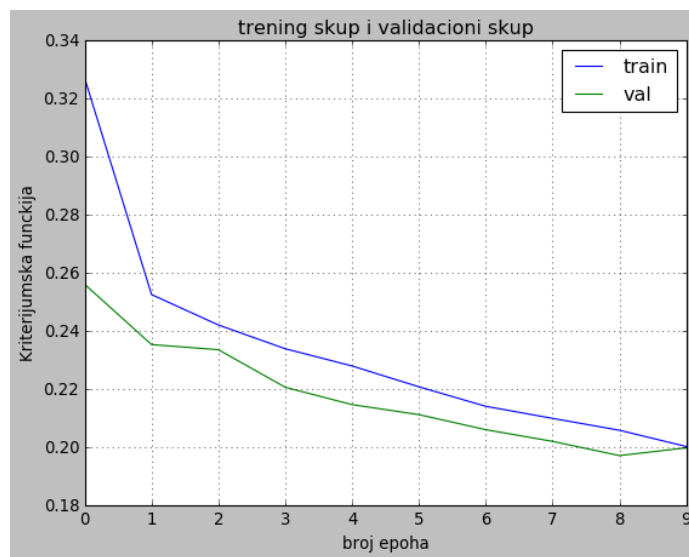


proces dosta vremenski zahtevan. Ipak postoje neki osnovni algoritmi koji mogu da pomognu pri nalaženju najboljih hiperparametara, jedan od njih je mrežna pretraga (eng. *Grid Search*). Grid Search radi tako što proverava model za sve vrednosti (iz nekog skupa) za sve parametre i upoređuje ih međusobno. Ovaj postupak jeste vremenski i računski zahtevan ali je nekad bolji od nasumičnog biranja hiperparametara. Sledeća slika ilustruje primenu Grid Search algoritma na traženje najbolje kombinacije dva parametra. Primenom tog algoritma dobijena je sledeća struktura: Neuralna mreža sa dva skrivena sloja koji imaju redom po dvadeset i deset neurona i jedan neuron u izlaznom sloju. Funkcije aktivacije su *relu* u skrivenim slojevima i *sigmoid* u izlaznom sloju. Kriterijumska funkcija za obučavanje je *binary cross-entropy* i koristi se *Adam* algoritam za optimizaciju obučavanja. Model je treniran kroz deset epoha a podaci za obučavanje su podeljeni u serije od po 128 elemenata.

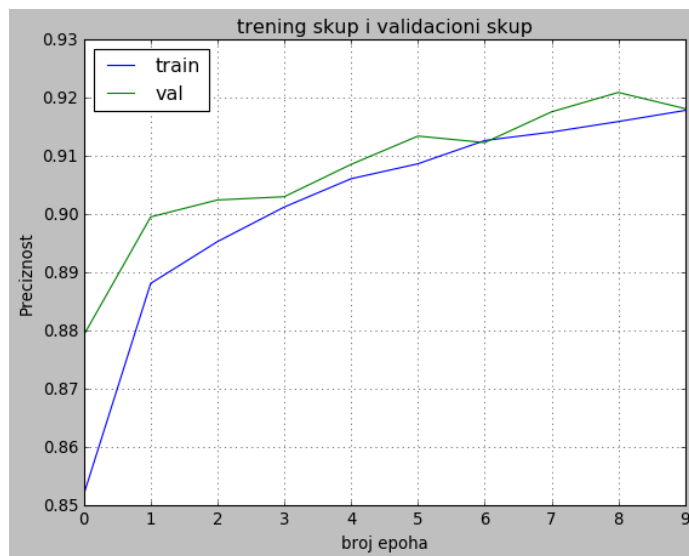


Slika 22. Grid Search algoritam za traženje hiperparametara.

Slede slike grafika koji su dobijeni u toku obučavanja konačnog modela, vidi se da preciznost raste i vrednost kriterijumske funkcije opada tokom celog treniranja, što je dobar znak da ne dolazi do preobučavanja:



Slika 23. Vrednost kriterijumske funkcije u toku obučavanja.



Slika 24. Preciznost modela u toku obučavanja.

Preciznost modela na test skupu je preko 90% što je vrlo dobar rezultat, mada se kasnije u primeni na klasifikaciji delova slike dobijaju subjektivno lošije performanse ali to je normalno jer model nije treniran na snimcima saobraćaja kao takvim.

## 2.5 Pretraga slike kliznim prozorom

Sada kada imamo obučenu mrežu koja dovoljno dobro klasifikuje automobile od ostalih slika treba nam sistem koji koristi taj model za pronalaženje automobila na slici saobraćaja. Postoji više načina da se to uradi, dva koja su vrlo slična su korišćenje piramide slika i korišćenje jedne slike ali više različitih prozora za pretragu. Piramida slike se zasniva na tome da se pretraga vrši prozorom konstantne veličine ali da se od ulazne slike dobije više slika koje su iste samo preskaliranih veličina.

Sledeće slike ilustruju primenu piramide slika, svaka slika je od prethodne umanjena 1.5 puta i sve tako dok nije dostignuta minimalna veličina. Veličina slike koja se pretražuje se menja ali crveni kvadrat (prozor koji se klasifikuje) je uvek istih dimenzija, time se postiže mogućnost detekcije automobila različitih veličina na jednoj slici korišćenjem efektivno jednog prozora. Ono što predstavlja problem kod ovakvog pristupa je izbor faktora umanjenja slika i minimalne veličine slike.



*Slika 25. Prikaz piramide slika.*

Piramidu slika je korisno upotrebiti kada je bitno da prozor za klasifikaciju uvek bude istih dimenzija (npr. zbog promenljivog broja obeležja). U ovom radu to nije slučaj jer korišćenjem histograma boja dobijamo uvek isti broj obeležja kolika god da je dimenzija prozora, tako da je korišćen malo drugačiji pristup – pretraga slike konstantne dimenzije sa više prozora različitih dimenzija. U nastavku rada opisan je postupak pretrage slike.

Korišćen je metod kliznog prozora, on se zasniva na tome da pravougaoni prozor prolazi kroz piksele slike sa jednog na drugi, sa početkom u gornjem levom uglu prvo ide horizontalno do kraja slike pa prelazi u sledeći red i sve tako do kraja slike. U svakom koraku ovog algoritma vrši se ekstrakcija obeležja iz trenutnog prozora i poziva se klasifikacija obučenim modelom.

Prvi deo sistema radi samo pretragu slike prozorom jedne veličine, i onda se on koristi za više veličina. Da bi napravili dobar sistem koji vrši pretragu sa jednim prozorom moramo znati veličinu tog prozora, dakle ovaj deo radi uz dodatnu informaciju o veličini prozora što u opštem slučaju nije poznato ali konačni sistem može da radi bez toga.

Za pretragu velike slike usvojeno je nekoliko detalja koji poboljšavaju vremensko i memorijsko zauzeće ali može da se desi da rezultat bude manje precizan. Prva stvar koja je usvojena je da se ulazna slika preskalira pre bilo kakve obrade, ovime se postiže da imamo manji broj piksela za pretragu a ne gubi se prevelika količina informacija. Skaliranje se radi tako što se širina slike uvek postavlja na 500 piksela a visina se računa tako da odnos dimenzija bude isti kao pre skaliranja. Sledeća slika je slikana kamerom sa mobilnog telefona i na njoj će se demonstrirati razvoj sistema, originalna slika je dimenzija  $4608 \times 3456$  a posle skaliranja dobijena je slika dimenzija  $500 \times 375$ , dakle ušteda u broju obrađenih piksela je ogromna (80-90 puta):



*Slika 26. Smanjivanje dimenzija slike zbog uštede vremena.*

Sledeći način uštede je uvođenje preskoka u toku pretrage, time se takođe znatno smanjuje broj piksela koji je potrebno obraditi. Mogu se u opštem slučaju uvesti dva različita preskoka za dva pravca pretrage ali u ovom radu korišćen je jedan isti. Eksperimentalno je dobijeno da preskok 20 ne remeti previše rad sistema a dovoljno smanjuje vreme izvršavanja: faktor uštede je  $20^2$ . Na sledećoj slici se vidi da i pored tako velikog faktora uštede greška nije zabrinjavajuća:



*Slika 27. Greška u detekciji sa preskokom od 20 piksela.*

Konačno, pretraga nekih regija slike je nepotrebna jer se može očekivati da u se u njima neće nalaziti automobil. Ako je slika dobijena sa kamere koja snima samo saobraćaj u uvek istim uslovima, razumno je pretpostaviti da se automobil neće nalaziti u gornjem delu (usvojeno gornja trećina) slike jer će tu uglavnom biti nebo, i takođe nema potrebe pretraživati nekoliko redova u dnu slike jer ni tu ne može da se nađe automobil. Sledeća slika pokazuje regiju na kojoj se vrši pretraga i lokacije na kojima se nalaze pikseli koji se za koje se koristi model:



*Slika 28. Regija slike u kojoj se vrši pretraga.*

Iako ta regija izgleda premalo, za svaki piksel koji se vrši pretraga gleda se ceo prozor, tako da regija u kojoj je moguće detektovati automobil je zapravo:



Slika 29. Regija u kojoj je moguće detektovati automobil..

Za svaki prozor, računaju se obeležja (192 broja histograma boja) i ona se ubacuju u obučenu neuralnu mrežu. Neuralna mreža je napravljena tako da na izlazu daje vrednost između 0 i 1 koja predstavlja sigurnost da je trenutni prozor automobil. Kada prozor prolazi kroz piksele vrednosti izlaza neuralne mreže se pamte u jednu matricu. Sledeća slika pokazuje matricu koja je dobijena za prethodni primer:

0.15...	0.14...	0.14...	0.15...	0.13...	0.12...	0.13...	0.20...	0.21...	0.16...	0.16...	0.17...	0.20...	0.27...	0.28...	0.28...	0.29...	0.23...	0.19...
0.80...	0.774	0.79...	0.78...	0.70...	0.57...	0.40...	0.39...	0.38...	0.24...	0.17...	0.13...	0.12...	0.15...	0.19...	0.21...	0.31...	0.39...	0.35...
0.77...	0.76...	0.76...	0.77...	0.76...	0.63...	0.48...	0.38...	0.37...	0.26...	0.21...	0.14...	0.14...	0.16...	0.21...	0.16...	0.16...	0.20...	0.29...
0.67...	0.57...	0.47...	0.41...	0.32...	0.17...	0.11...	0.09...	0.12...	0.12...	0.14...	0.12...	0.17...	0.25...	0.30...	0.28...	0.23...	0.26...	0.31...
0.10...	0.07...	0.05...	0.03...	0.03...	0.02...	0.02...	0.03...	0.05...	0.09...	0.17...	0.25...	0.38...	0.55...	0.64...	0.73...	0.69...	0.60...	0.47...
0.01...	0.01...	0.01...	0.01...	0.01...	0.02...	0.03...	0.05...	0.10...	0.16...	0.29...	0.45...	0.59...	0.79...	0.90...	0.94...	0.93...	0.86...	0.67...

Slika 30. Matrica vrednosti dobijenih posle pretrage.

Tokom pretrage normalno je da model daje visoke vrednosti za nekoliko susednih piksela ako se u oba prozora delimično ili potpuno nalazi automobil. Treba postaviti prag odlučivanja, iznad koga će se smatrati da je slika automobil, problem je što ovaj prag dosta zavisi od slike na kojoj se program pokreće (osvetljenja, vremenskih uslova...) tako da se on može smatrati hiperparametrom ovog sistema. Ovaj rad se ne bavi automatskim



podešavanjem ovog parametra tako da sistem radi podjednako za svaki test slučaj ali to može biti nastavak ovog rada. Za ovaj primer zgodno je izabrati prag 0.9, ako se izabere recimo prag 0.8 dobijamo neželjen rezultat.

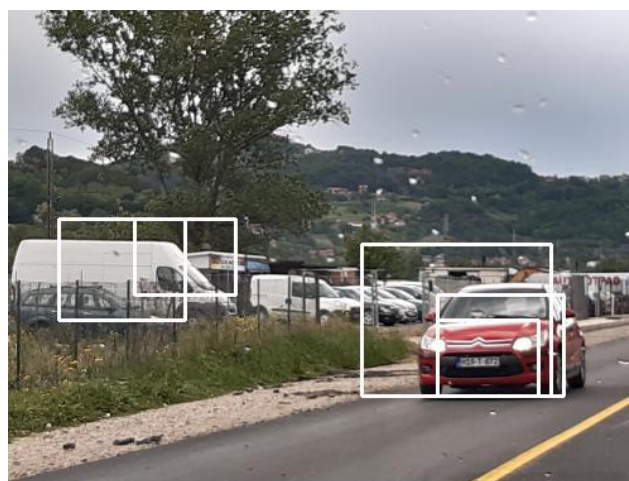


*Slika 31. Rezultati pretrage slike i korišćenja praga 0.8 i 0.9.*

Problem dobijanja više susednih prozora koji zadovoljavaju prag za jedan automobil je rešen kasnije u delu sa više prozora različitih veličina.

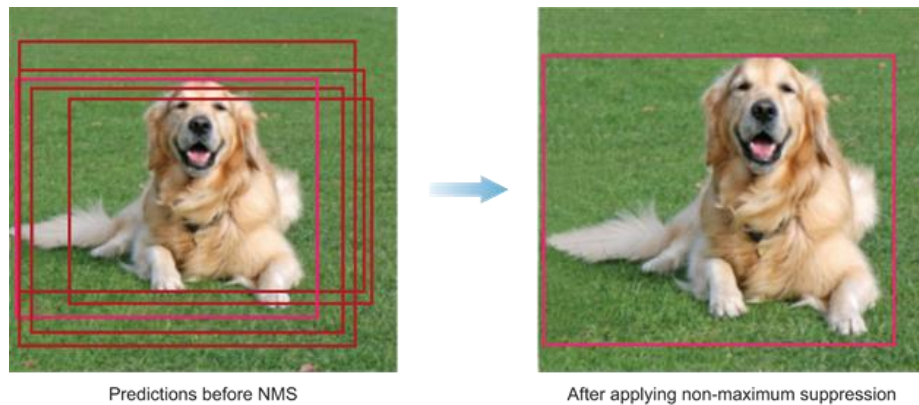
Pitanje je kako izabrati veličine prozora za pretragu tako da je moguće detektovati i udaljene automobile i one koji su blizu. Na osnovu test slika koje su korišćene izabrano je sledeće četiri veličine: (200, 150), (150, 120), (100, 80), (80, 60). Ove veličine prozora su primenjene na reskaliranu sliku kojoj je širina postavljena na 500. Veličine prozora za pretragu su takođe parametar na kome može da se napravi unapređenje sistema, jer su ove veličine izabrane čisto eksperimentalno i posmatranjem.

Sada se za svaku veličinu prozora poziva program koji vrši pretragu za jednu određenu veličinu i svi ti rezultati se spajaju u jedno, sledeća slika demonstrira taj međurezultat (prag je 0.8 samo zbog lakše demonstracije):



*Slika 32. Rezultat pokretanja programa za više veličina prozora.*

Sada problem postaje to što za jedan automobil dobijamo više prozora čiji izlaz in neuralne mreže zadovoljava zadati prag. Ovaj problem se mogao izbjeći tako što smo prvi put kada smo dobili rezultat veći od praga prestali da pretražujemo tu regiju, ali ovde je korišćen drugačiji pristup. Korišćen je NMS (eng. *Non-maximum Compression*) algoritam za izdvajanje jednog prozora iz grupe prozora koja pokazuje na isti objekat.



Slika 33. Demonstracija NMS algoritma, preuzeto iz [9].

NMS algoritam funkcioniše tako što prolazi kroz sve prozore koji su dali rezultat veći od praga, i za svaki par prozora ako se preklapaju računa koeficijent preklapanja para prozora. Taj koeficijent se računa kao odnos regije koja je preklapljenjena i cele regije jednog od prozora i za one prozore za koje je koeficijent veći od nekog praga se računa da nisu potrebni pa se izbacuju.

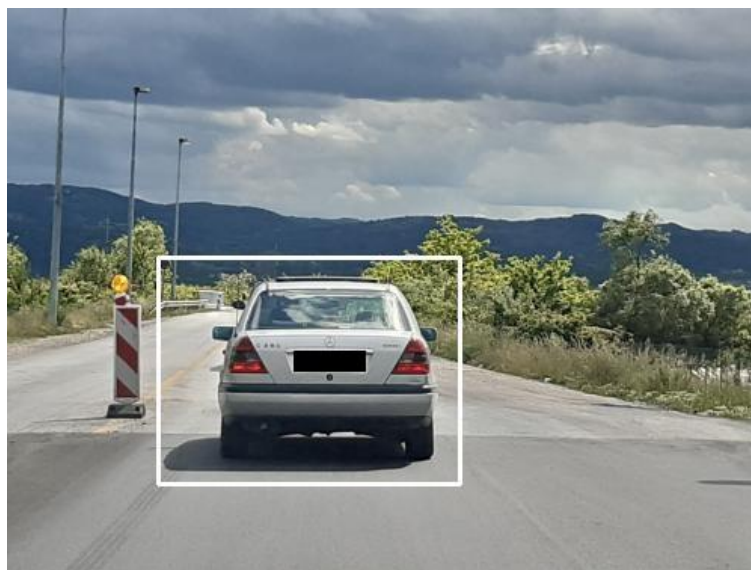


Slika 34. Rezultat primene NMS.



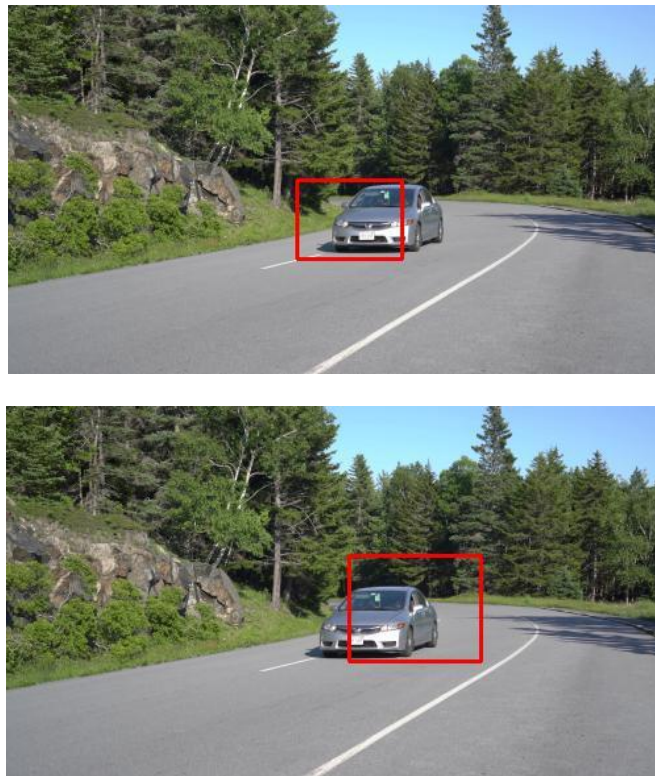
### 3 REZULTATI

Primenom svih koraka objašnjenih u prethodnom poglavlju dobijamo program koji radi zadovoljavajuće dobro na primerima koji su slični onima na kojima je modela treniran ali može da ima poteškoće u situacija koje su mu nove. Sledi nekoliko slika rezultata primene i kratki komentari:



*Slika 35.*

Na ovoj slici se vidi da dosta zavisi od izbora veličina prozora, rezultat je vrlo zadovoljavajuć ali boljim izborom dimenzija može se dobiti preciznija detekcija. Dobra stvar je što što je prozor veći to manje piksela treba da proverava, tako da vremenska cena dodavanja još većih prozora nije velika. Za razliku od toga mali prozori pozivaju klasifikaciju modelom mnogo više puta tako da ne treba ubaciti previše male prozore ako želimo vremensku efikasnost.



*Slika 36.*

Ovaj rezultat je dobijen kada je sistem pokrenut na celom video snimku automobila koji vozi šumskim putem, ovo su dva susedna prozora. Rezultat je ponovo zadovoljavajuć ali se vidi nepreciznost koja je dobijena zbog vremenske uštede uvođenjem preskoka i menjanjem dimenzija slike.



*Slika 37.*

Još jedan dobar rezultat, ovo je slika koja je izvučena iz videa iz kog su isečene podslike za obučavanje modela (drugi deo videa se koristi za obučavanje). Sistem je pravilno detektovao jedan automobil ali nije uspeo da detektuje sitnije automobile u daljini. Ovo bi moglo da se popravi dodavanjem manjih prozora za pretragu, ali što je prozor manji to mu je i rezolucija manja i to može da utiče na tačnost dobijenih rezultata.



*Slika 38.*

U nastavku istog snimka dolazimo do ove situacije, sistem pravilno detektuje automobil ali detektuje i pešaka koji se nalazi na trotoaru. Ovo može biti rezultat lošeg praga odlučivanja, nedostatka dobrih podataka za obučavanje za klasu nije-automobil, ili može da se desi da neki automobil jednostavno ima sličan histogram boja kao ova podslika u kom slučaju teško da je moguće rešiti problem ovim pristupom.



*Slika 39.*

Na ovoj slici vidi se da je sistem pravilno detektovao automobil koji mu dolazi u susret ali nije detektovao parkirani automobil sa strane. Ovaj rezultat je očekivan jer je model treniran samo na slikama na kojima se automobili vise od napred, od pozadi ili eventualno pod nekim uglom ali ne sa strane. Rešenje bi moglo da bude ili obučavanje nekog modela koji bi prepoznao automobil iz bilo koje perspektive (teško jer je prevelika varijacija takvih slučajeva) ili obučavanje više modela koji bi se koristili istovremeno. Prozor bi prolazio kroz sliku i za svaki piksel bi pokretao više modela i na osnovu njihovih rezultata donosio zaključke.

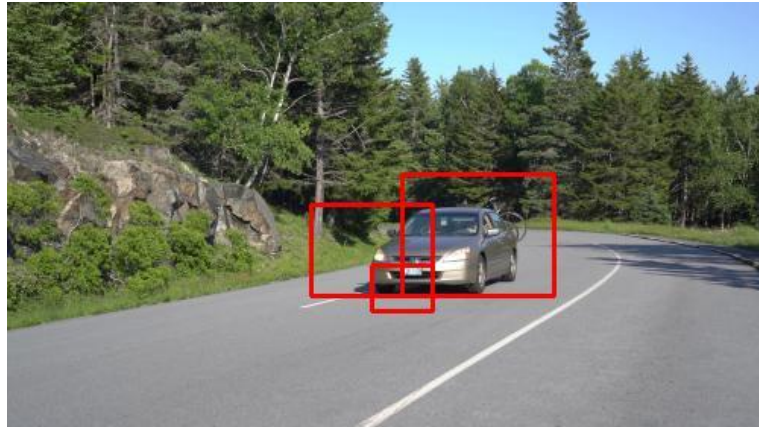


*Slika 40.*

Na ovoj slici se takođe vidi da neraznovrsnost skupa za obučavanje može da dovede do toga da sistem ne radi baš kako treba. Model automobila na slici nije bio u skupu za obučavanje i zbog toga sistem nije mogao da izvrši dobru detekciju.



Sledi još nekoliko rezultata kao demonstracija rada sistema:



*Slika 41.*



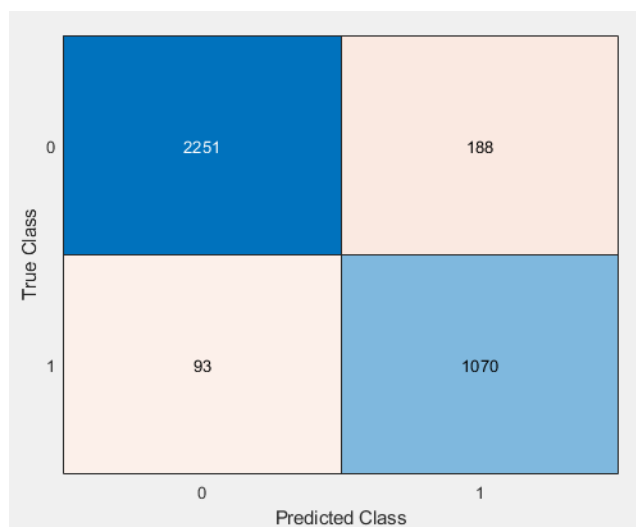
*Slika 42.*



*Slika 43.*

## 4 DISKUSIJA

Ovaj sistem se sastoji iz dve relativno odvojene celine, klasifikatora i algoritma pretrage pa će oni biti prodiskutovani odvojeno. Klasifikator (trenirana neuralna mreža) sam po sebi radi vrlo dobro, na test skupu koji je izdvojen iz ukupnog skupa podataka dobijena je tačnost od 92.19% i sledeća slika pokazuje njegovu konfuzionu matricu. Odnos broja slika u klasama u ukupnom, pa tako i test skupu, je oko 2:1 u korist klase nije-automobil (na matrici obeleženo sa 0) tako da se ispostavlja da klasifikator radi otprilike podjednako dobro za obe klase.



Slika 44. Konfuzionna matrica klasifikatora

Ono gde ovaj sistem počinje da ima problem je upotreba ovog klasifikatora u pretrazi slike. Kada prozor pretrage klizi kroz sliku on često nailazi na delimične slike automobila koje nije mogao da vidi u skupu na kome je obučavan tako da se često zbuni i izbacuje pogrešne rezultate. Uz to postoji još nekoliko razloga zbog kojih bi ovaj sistem mogao da bude manje precizan, ali ti nedostaci su istovremeno i mesto za popravku i dalji rad:

- Prvi je činjenica da je najveći deo trening podataka došao iz skupa *CompCars* koji sadrži slike koje nisu napravljene u saobraćaju već u izložbenim uslovima tako da je model dosta svojih resursa uložio u obučavanje za prepoznavanje takvih slika a to se malo razlikuje od scenarija u kojima je testiran.

- Druga stvar, takođe vezana za *CompCars* skup je to da je taj skup sačinjen od slika automobila novijih i skupljih modela pa je mreža imala problema sa generalizacijom na sve automobile.
- Treća stvar, možda i najbitnija, je to što sistem ne vrši nikakvu vrstu prepoznavanja oblika automobila već se sve radi isključivo korišćenjem histograma boja. Očekivano je da ovaj sistem nema performanse koje imaju sistemi koji koriste konvolucione neuralne mreže za detekciju, one su dosta bolji pristup baš iz razloga što u svoju obradu slike uključuju prostorni raspored piksela. Zanimljivo je zamisliti sliku automobila, korišćenjem bilo kakve transformacije slike takve da se ne menja vrednost piksela već samo njihov raspored apsolutno ne menjamo obeležja koja vidi neuralna mreža se ne menjaju tako da sistem ostaje siguran da je slika automobil iako ne bi trebalo da bude.
- Četvrto mesto za popravku je, kao i kod svakog sistema koji koristi tehnike mašinskog učenja, popravka količine, kvaliteta i raznovrsnosti podataka za obučavanje mreže.

## 5 ZAKLJUČAK

Odgovor na pitanje da li je moguće koristiti histograme boja za detekciju automobila je da je moguće ali možda ne samostalno već kao asistentski sistem nekom sigurnijem klasifikatoru, na primer ovaj sistem bi mogao da bude pokrenut sa nižim pragom odlučivanja (da bi što manje stvarnih automobila bilo odbačeno) a onda se može na njegove rezultate primeniti taj drugi klasifikator. Ukoliko je vremenski zahtevno da drugi klasifikator radi sa celom slikom (ako su recimo obeležja komplikovanija), ovim postupkom se možda može postići da klasifikacija bude brža od korišćenja samo drugog klasifikatora ali i preciznija od korišćenja samo ovog sistema.

Prednost ovakvog pristupa je definitivno jednostavnost izvlačenja obeležja iz slike i činjenica da slika proizvoljne dimenzije ima isti broj obeležja, dok bi mana definitivno bio nedostatak prepoznavanja morfoloških karakteristika automobila.

Dobar deo performansi je izgubljen i u pokušaju da se vreme izvršavanja svede na što manje jer je u ovakvim primenama to odlučujući faktor o primenjivosti sistema, dobijeno ubrzanje je reda veličine 5000 - 10000 puta što je zadovoljavajuće sa obzirom da je uvedeno samo nekoliko jednostavnih algoritama koji ne kvare previše performanse sistema.

Parametri koji se ne obučavaju a koji određuju rad sistema zovu se hiperparametri. Oni se mogu podeliti u dve grupe: oni koji se tiču modela i oni koji se tiču algoritma pretrage.

Hiperparametri modela su:

- Arhitektura mreže – broj slojeva i neurona u njima
- Kriterijumska funkcija
- Algoritam optimizacije
- Broj epoha obučavanja
- Broj podataka u seriji tokom obučavanja

Hiperparametri algoritma pretrage:

- Broj i dimenzije prozora za pretragu
- Preskok svakog prozora – može biti različit u zavisnosti od veličine
- Prag odlučivanja klasifikatora
- Prag odbacivanja prozora u *NMS* algoritmu

Prostora za unapređenje ima na nekoliko mesta: na podacima za obučavanje, na dodavanjima obeležja, na optimizaciji hiperparametara...



## 6 LITERATURA

- [1] Lin, C. T. (Chin Teng). Neural fuzzy systems : a neuro-fuzzy synergism to intelligent systems / Chin-Teng Lin, C. S. George Lee. B. Klaus and P. Horn, Robot Vision. Cambridge, MA: MIT Press, 1986.
- [2] Rafael C. Gonzales, Richard E. Woods, "Digital Image Processing"
- [3] Veljko Papić, Materijali sa predmeta Digitalna Obrada Slike (13E054DOS)
- [4] Pyimagesearch, "Sliding Windows for Object Detection with Python and OpenCV"
- [5] Željko Đurović, Materijali sa predmeta Prepoznavanje Oblika (13E054PO)
- [6] Linjie Yang, Ping Luo, Chen Change Loz, Xiaoou Tang, "A Large-Scale Car Dataset for Fine-Grained Categorization and Verification", Chinese Academy of Science, 2015.
- [7] J. Arróspide, L. Salgado, M. Nieto, "Video analysis based vehicle detection and tracking using an MCMC sampling framework", *EURASIP Journal on Advances in Signal Processing*, vol. 2012, Article ID 2012:2, Jan. 2012
- [8] Gabriel J. Brostow and Jamie Shotton and Julien Fauqueur and Roberto Cipolla, "Segmentation and Recognition Using Structure from Motion Point Clouds", 2008.
- [9] Mohamed Elgendy, "Deep Learning for Vision Systems", Jan. 2019
- [10] Jason Brownlee, "Deep Learning Performance", July 2017