

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 000

Klasifikacija prometnih znakova

Matija Pavlović

Zagreb, svibanj 2023.

*Umjesto ove stranice umetnite izvornik Vašeg rada.
Da bi ste uklonili ovu stranicu obrišite naredbu \izvornik.*

SADRŽAJ

1. Uvod	1
2. Pregled postojeće literature	2
3. Metodologija rada	3
3.1. Prikupljanje podataka za treniranje	4
3.2. Pretprocesiranje	5
3.3. Augmentacija skupa podataka	7
3.4. Treniranje modela	8
3.5. Testna aplikacija	10
4. Rezultati	11
5. Budući rad	12
6. Zaključak	13
Popis slika	15

1. Uvod

Razvoj tehnologije u automobilske industriji u stopu prate i sve veći zahtjevi tržišta za novim sigurnosnim značajkama te značajkama koje doprinose udobnosti korištenja vozila. Novi modeli vozila tako postaju opremljeni značajnim brojem senzora na vanjskoj strani vozila i značajnim brojem ekrana i signalnih lampica u unutrašnjosti vozila. Kada sjednemo za upravljač novijih vozila sve češće možemo primijetiti da nas vozilo upozorava na prometne znakove, primjerice ograničenja brzine, zabrane pretjecanja, znakove obaveznog zaustavljanja itd. Razmotrimo li i činjenicu da ubrzano raste i broj vozila s određenim stupnjem autonomije pri vožnji postaje jasno da su sustavi koji u stvarnom vremenu detektiraju i klasificiraju prometne znakove postali izrazito važni u razvoju novih modela vozila. Cilj ovog završnog rada je demonstracija rada jednog takvog sustava uz detaljni opis primjene, problema s kojima se sustav može suočavati u stvarnim okolnostima, te opis implementacije sustava. U sklopu rada ću razviti model strojnog učenja temeljen na dubokoj konvolucijskoj mreži, obraditi skup podataka za treniranje i testiranje modela, te programski kod koji će koristiti kameru prijenosnog računala kako bi klasificirao prometne znakove.

2. Pregled postojeće literature

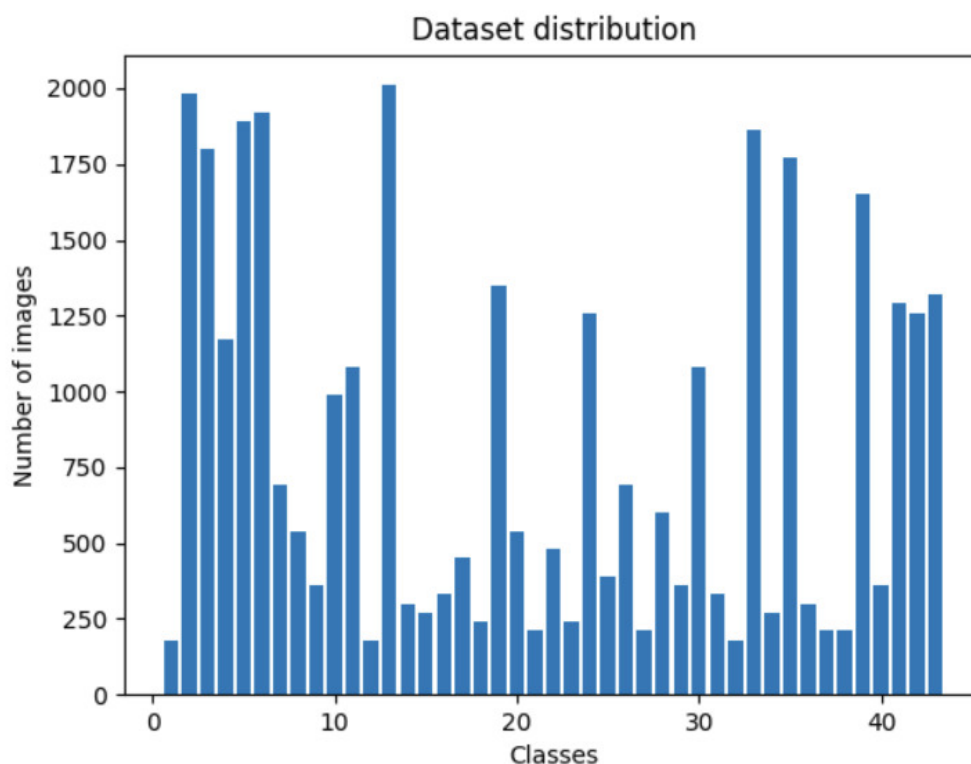
Klasifikacija prometnih znakova nije nov pojam u području računarske znanosti i specifičnije području računalnog vida. Shodno tome postoji velik broj znanstvenih radova koji se bave tom problematikom. Većina pristupa problemu klasifikacije se bazira na pristupu usmjerenom na prepoznavanje boja i oblika.(?)

3. Metodologija rada

U poglavlju metodologija rada opisati ću metode pri izradi projekta od razine prikupljanja i prilagodbe podataka za treniranje modela strojnog učenja, kreiranje samog modela, treniranje modela te naposljetku i izradu testne aplikacije kojom se demonstrira rad sustava.

3.1. Prikupljanje podataka za treniranje

Skup podataka za treniranje odnosno *Dataset* korišten u ovom radu je je preuzet iz elektronske arhive istraživačkih radova Sveučilišta u Kopenhagenu (Electronic Research Data Archive). *Dataset* je dio *German Traffic Sign Recognition Benchmark-a* (GTSRB), a kreirali su ga Johannes Stallkamp, Marc Schlipsing, Jan Salmen, Christian Igel. Navedeni skup podataka se sastoji od 34799 slika, raspodijeljenih u 43 razreda koji predstavljaju 43 različita prometna znaka.



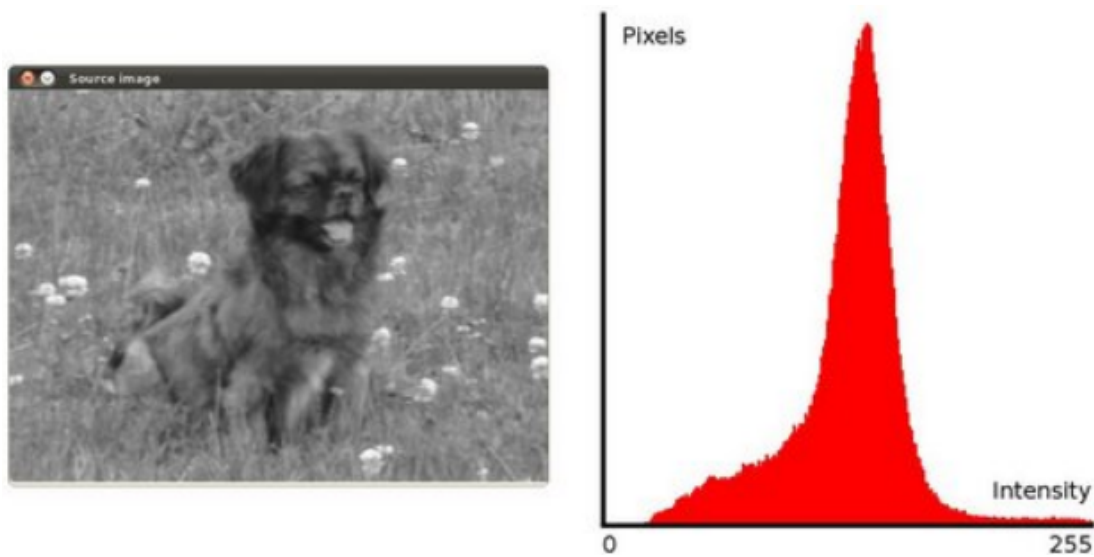
Slika 3.1: Raspodijela dataseta po razredima.

3.2. Pretprocesiranje

Pretprocesiranje ulaznog skupa podataka obavlja se pomoću sljedećeg bloka programskog koda:

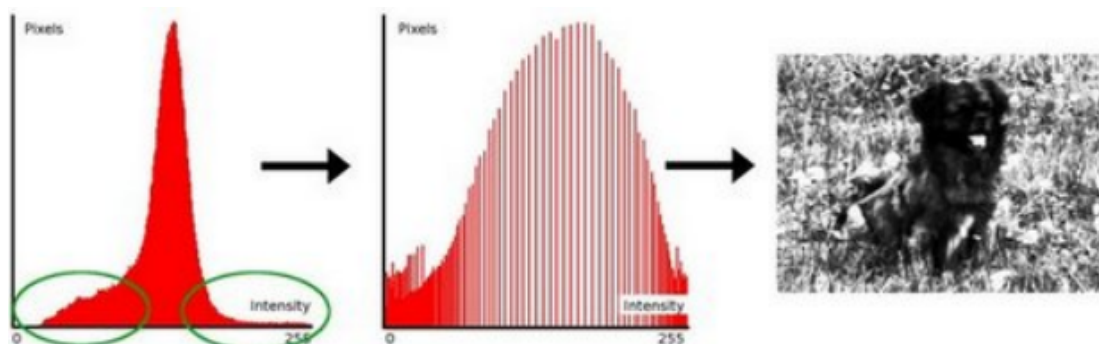
```
1 import cv2
2
3
4 def preprocess(image) :
5     image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
6     image = cv2.equalizeHist(image)
7     image = image / 255
8     return image
9     return data_gen
```

S obzirom da za prepoznavanje zadanog skupa prometnih znakova nije potrebna informacija o boji, svaka ulazna slika se pretvara u *grayscale*. Također kako bi *dataset* bio što iskoristiviji, slikama se povećava kontrast tehnikom histogramskog izjednačavanja. Metoda izjednačavanja histograma (HE) je široko korištena tehnika pretprocesiranja slike i svrstava se u kategoriju korekcija svjetline. Primjeri primjene HE su poboljšanje medicinskih slika, znanstvenih fotografija, povijesnih slika i sonarskih slika. Histogramsko izjednačavanje se također koristi i za poboljšanje kontrasta slike u digitalnim slikama. Navedeni učinci se postižu učinkovitim distribuiranjem najčešćih vrijednosti intenziteta. To omogućuje da povećamo kontrast na lokalnim dijelovima slike s lošim kontrastom.(?) Tehnikom histogramskog izjednačavanja se efektivno proširuje raspon vrijednosti intenziteta s ciljem povećanja kontrasta. Ova tehnika se često koristi i na slikama koje su imale loše osvjetljenje. Naposljetku vrijednosti svakog piksela slike se normaliziraju na vrijednost iz skupa [0, 1]. Navedenim postupcima pretprocesiranja se pospješuje brzina i učinkovitost treniranja modela.



Slika 3.2: Primjer slike prije histogramskog izjednačavanja i odgovarajući histogram.

Source: https://docs.opencv.org/3.4/d4/d1b/tutorial_histogram_equalization.html



Slika 3.3: Primjer slike nakon histogramskog izjednačavanja i odgovarajući histogram.

Source: https://docs.opencv.org/3.4/d4/d1b/tutorial_histogram_equalization.html

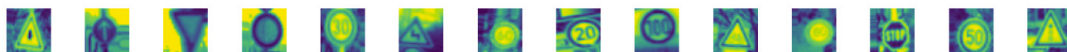
3.3. Augmentacija skupa podataka

Kako bi iskoristivost skupa podataka za treniranje bila maksimizirana korištena je augmentacija nad ulaznim skupom. Augmentacija se provodi u sljedećem bloku programskog koda:

```
1 from keras.preprocessing.image import ImageDataGenerator
2
3
4 def augment():
5     data_gen = ImageDataGenerator(width_shift_range=0.1,
6                                   height_shift_range=0.1,
7                                   zoom_range=0.2,
8                                   shear_range=0.1,
9                                   rotation_range=10)
10    return data_gen
```

Gore prikazana funkcija koristi `ImageDataGenerator` funkciju iz `keras.preprocessing.image` modula. Uz navedene argumente ova metoda proširuje *dataset* tako što svaku sliku

- Nasumično pomiče horizontalno uz maksimalni faktor od 10% širine slike
- Nasumično pomiče vertikalno uz maksimalni faktor od 10% visine slike
- Uvećava sliku u rasponu od 0% do 20%
- Posmiče slike uz maksimalni kut posmaka od 10°
- Rotira sliku uz maksimalni kut rotacije od 10°



Slika 3.4: Prikaz augmentiranih slika.

3.4. Treniranje modela

Model strojnog učenja korišten za izradu ovog rada zasniva se na Keras API-u. Sastoji se od jedanaest slojeva i reprezentiran je sljedećim isječkom Python koda:

```
1 from keras.models import Sequential
2 from keras.layers import Dense
3 from keras.layers import Dropout, Flatten
4 from keras.layers.convolutional import Conv2D, MaxPooling2D
5
6
7 def learning_model(classes_len):
8
9     model = Sequential()
10    model.add(Conv2D(60, (5, 5), input_shape=(32, 32, 1),
11    activation='relu'))
12    model.add(Conv2D(60, (5, 5), activation='relu'))
13    model.add(MaxPooling2D(pool_size=(2, 2)))
14
15    model.add(Conv2D(30, (2, 2), activation='relu'))
16    model.add(Conv2D(30, (2, 2), activation='relu'))
17    model.add(MaxPooling2D(pool_size=(2, 2)))
18    model.add(Dropout(0.5))
19
20    model.add(Flatten())
21    model.add(Dense(500, activation='relu'))
22    model.add(Dropout(0.5))
23    model.add(Dense(classes_len, activation='softmax'))
24
25    model.compile(optimizer='adam', loss='categorical_crossentropy',
26    metrics=['accuracy'])
27    return model
```

Prvi sloj je Conv2D sloj od 60 filtera veličine 5x5 piksela sloj koji prima ulaz oblika 32 piksela visine, 32 piksela širine i 1 piksel dubine jer se radi o *grayscale* slici. Konvolucijski slojevi neuronske mreže povezuju uzorke na manjim lokalnim regijama sloja s uzorcima koji se pojavljuju na korespondirajućim regijama iz viših slojeva čime se eliminira potreba za stvaranjem potpune povezanosti među slojevima. (?) Korištenjem nekolicine pojednostavljenja konvolucijski slojevi smanjuju broj stvorenih veza i za

nekoliko redova veličine.(?) Sloj koristi ReLU aktivaciju definiranu formulom:

$$ReLU(x) = \max(0, x)$$
$$\frac{d}{dx}(x) = \{1 \text{ if } x > 0; 0 \text{ otherwise}\}$$

Glavna prednost ove aktivacije je što za ulaz 0, uvijek vraća 0, što nije slučaj kod nekih zasićenih funkcija poput *tanh* i *sigmoid* funkcija. Ovakvo ponašanje može naštetiti treniranju modela. Drugi, četvrti i peti slojevi modela imaju identičnu zadaću kao i prvi te se stoga ne razmatraju detaljnije. Treći sloj

Model koristi optimizator Adam...

3.5. Testna aplikacija

Kako bi se testirala učinkovitost modela, razvijena je jednostavna testna skripta u programskom jeziku Python. Skripta uključuje *web-kameru* računala, učitava istrenirani model te povezuje naziv svakog razreda s njegovim indeksom pomoću riječnika. Primjerice:

```
1 class_names = {  
2     0: 'Speed Limit 20 km/h',  
3     1: 'Speed Limit 30 km/h',  
4     ...  
5 }
```

Zatim u beskonačnoj petlji učitava sliku s *web-kamere*, pretprocesira tu sliku, vrši predikciju pomoću istreniranog modela. Ukoliko je vjerojatnost s kojom je model predividio da se na slici nalazi neki znak veća od unaprijed zadanog praga na zaslonu se ispisuje naziv znaka te vjerojatnost s kojom model klasificira sliku znaka u pretpostavljeni razred. Naprotiv, ukoliko je vjerojatnost niža od zadanog praga ispisuje se samo prikladna poruka da model ne prepoznaje znak na slici.

Uz jasna ograničenja koje ovaj model testne aplikacije posjeduje (nedostatak grafičkog sučelja, uporaba ugrađene kamere umjesto zasebne kamere ili senzora), aplikacija ipak može poslužiti kao kvalitetni pokazatelj učinkovitosti i brzine kojom model vrši predikcije i daje finalni rezultat.

4. Rezultati

5. Budući rad

Budući rad na ovom problemu bi uključivao implementaciju detekcije prometnih znakova na slici kako bi cjelokupni sustav bio što iskoristiviji u stvarnom svijetu. Nužno bi bilo uvesti i neku vrstu dedicanog senzora ili kamere koja bi imala dovoljne mogućnosti stabilizacije, rada u mraku i uvjetima smanjene vidljivosti (kiša, magla, snijeg...). Sljedeći korak bi bio treniranje modela na jačem *hardware-u* kroz više epoha kako bi se pospiješila klasifikacija znakova. Posljednji koraci budućeg razvoja bi uključivali verzioniranje sustava za razna tržišta, primjerice razvoj odvojenih verzija za tržište Sjedinjenih Američkih Država zbog razlike u standardu prometnih znakova u odnosu na europske standarde za koje je trenutna aplikacija prilagođena.

6. Zaključak

U zaključku...

Klasifikacija prometnih znakova

Sažetak

Sažetak na hrvatskom jeziku.

Ključne riječi: Klasifikacija, računalni vid, strojno učenje, duboko učenje, duboke neuronske mreže, konvolucijske neuronske mreže, prometni znakovi, promet, DNN, CNN, CV, ML

Traffic sign classification

Abstract

Abstract.

Keywords: Classification, Computer Vision, Machine Learning, Deep Learning, Deep Neural Networks, Convolutional Neural Networks, Traffic Signs, Traffic, DNN, CNN, CV, ML

POPIS SLIKA

3.1. Raspodijela dataseta po razredima.	4
3.2. Primjer slike prije histogramskog izjednačavanja i odgovarajući histogram.	6
3.3. Primjer slike nakon histogramskog izjednačavanja i odgovarajući histogram.	6
3.4. Prikaz augmentiranih slika.	7