

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	12
3.6. Testiranje sustava	12
4. Rezultati	13
5. Budući rad	14
6. Zaključak	15
Literatura	18
Popis slika	20

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

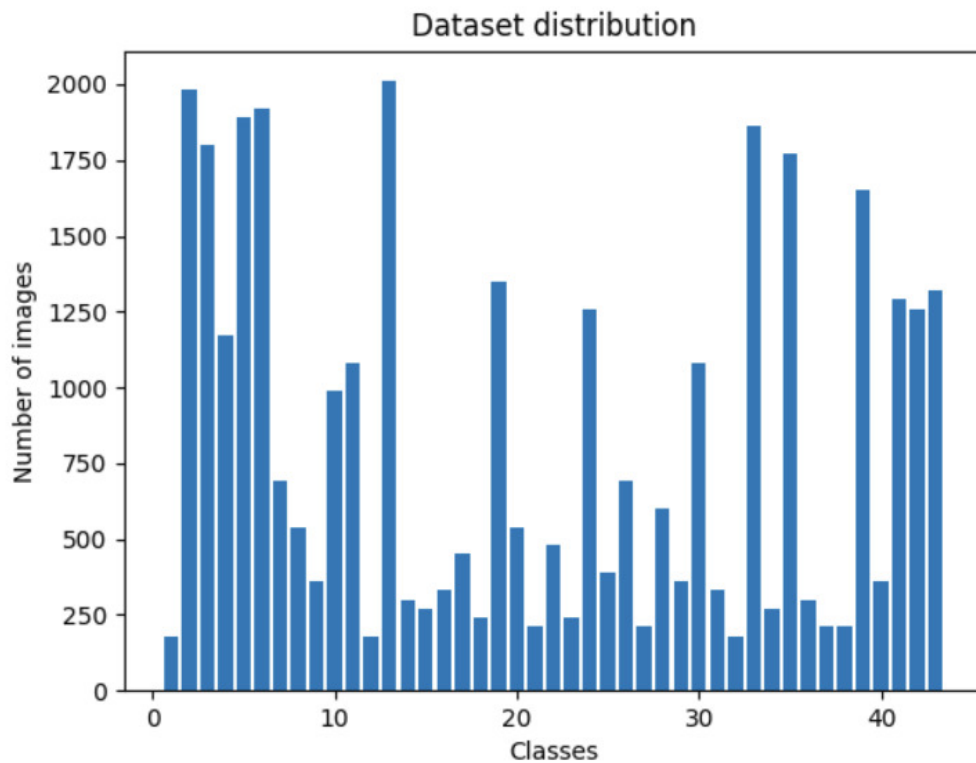
Klasifikacija prometnih znakova nije nov pojam u području računarske znanosti i specifičnije području računalnog vida. Shodno tome postoji velik broj znanstvenih radova koji se bave tom problematikom. Većina pristupa problemu klasifikacije se bazira na pristupu usmjerenom na prepoznavanje boja i oblika.(Zaklouta et al., 2011) Uz klasifikaciju prometnih znakova se često veže i pojam detekcije. Kad razmatramo klasifikatore koji su učestalo korišteni pri izradi sustava za detekciju prometnih znakova pronalazimo širok spektar metoda. U prvom redu se ističu neuronske mreže, metode potpornih vektora, u hrvatskoj literaturi poznata i kao metoda jezgrenih funkcija, konvolucijske neuronske mreže, nasumične šume, metoda najbližih susjeda... Iz ovog skupa klasifikatora se za zadaće klasifikacije ipak najčešće koriste konvolucijske neuronske mreže.(Mehta et al., 2019) Cireşan, Meier, Masci i Schmidhuber su pak izbjegli korištenje pretpostavljenih ekstraktora značajki i zamjenili ih postupcima nadziranog učenja. Kombinacijom raznih dubokih neuronskih mreža treniranih na različito preprocesiranim podacima ostvarili su MCDNN, odnosno višestupčanu neuronsku mrežu, tako ostvarivši sustav koji je otporan na varijacije u kontrastu i osvjetljenju. (Cireşan et al., 2012)

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. Isto tako detaljno ću opisati i arhitekturu same duboke neuronske mreže, argumentirati izbor aktivacijskih funkcija korištenih na izlazima iz slojeva neuronske mreže, odabir optimizatora te odabir metode za proračun pogreške predikcije modela.

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.



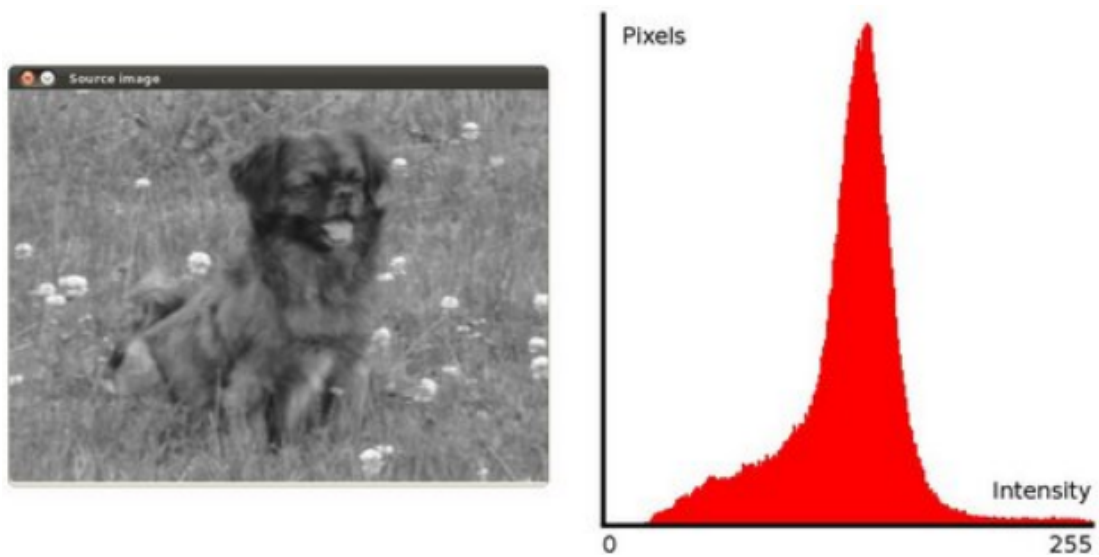
Slika 3.1: Raspodijela dataseta po razredima.

3.2. Pretprocesiranje

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

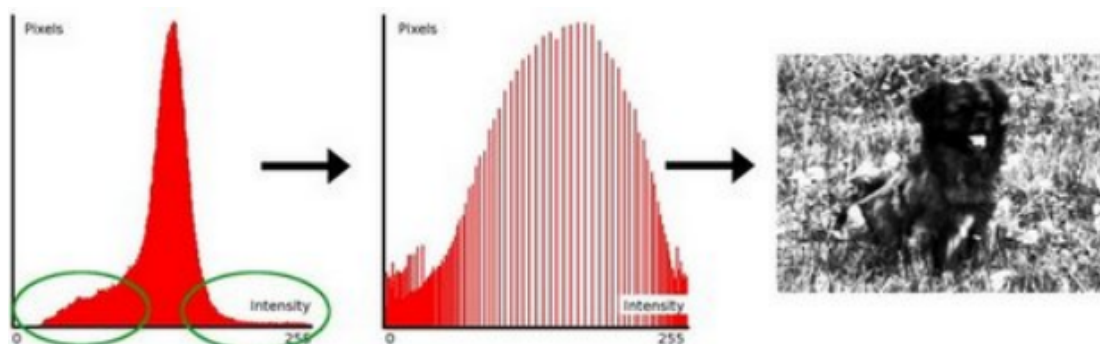
```
1 import cv2
2
3
4 def preprocess(image) :
5     image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
6     image = cv2.equalizeHist(image)
7     image = image / 255
8     return image
9     return data_gen
```

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.(Celebi et al., 2021) 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

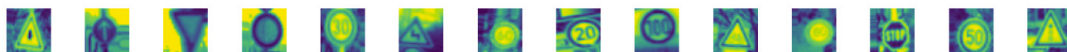
3.3. Augmentacija skupa podataka

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 ulazni sloj neuronske mreže koji prima *tensor* oblika (32, 32, 1), odnosno broječanu reprezentaciju slike veličine 32*32 piksela s jednim kanalom (*grayscale*). Zatim slijedi Conv2D sloj od 60 filtera veličine 5x5 piksela. 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. (Albawi et al., 2017) Korištenjem nekolicine pojednostavljenja konvolucijski slojevi smanjuju broj stvorenih veza i za nekoliko redova veličine.(Turner, 2014) Sloj koristi ReLU aktivacijsku funkciju,

odnosno zglobovcu definiranu formulom:

$$ReLU(x) = \max(0, x)$$

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

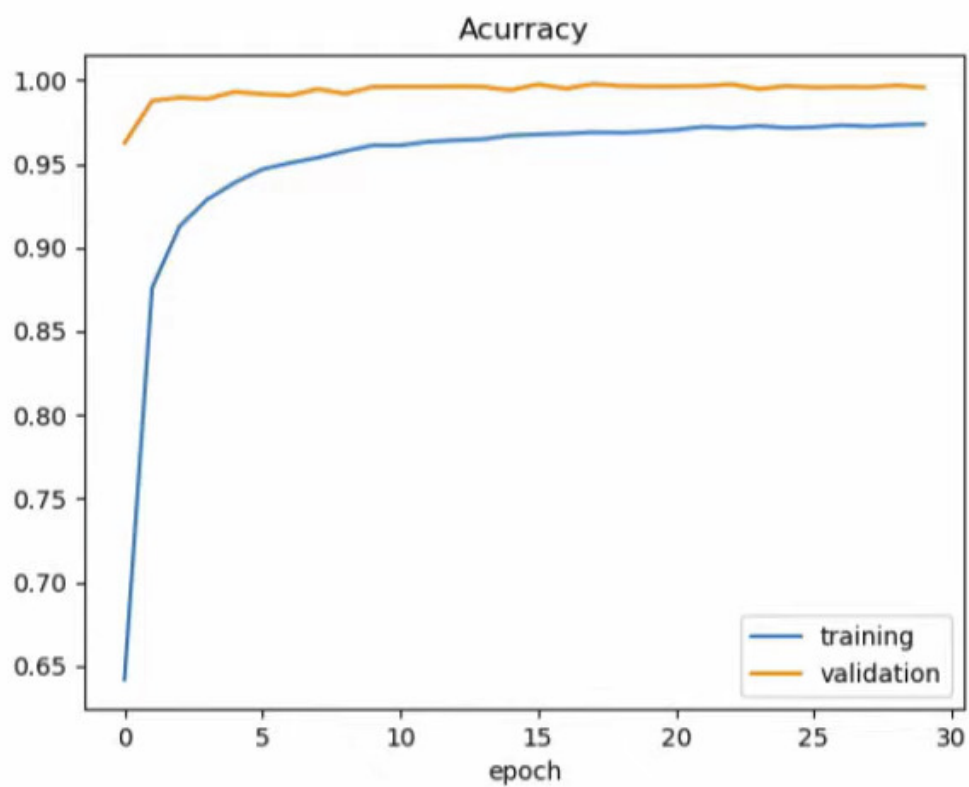
Glavna prednost ove aktivacije je što za ulaz 0, uvijek vraća 0, što nije slučaj kod nekih zasićenih funkcija poput *tanh* i *sigmoid* funkcija. Ovakvo ponašanje može naštetiti treniranju modela. Treći, peti i šesti slojevi modela imaju identičnu zadaću kao i prvi te se stoga ne razmatraju detaljnije. Četvrti sloj je MaxPooling2D sloj, točnije agregacijski sloj. Zadaća agregacijskih slojeva je smanjivanje reprezentacije ulaza. Agregacijski sloj korišten u ovom modelu koristi prozor veličine 2*2 piksela te od te četiri vrijednosti uzima isključivo najveću. Ovim postupkom se pospješuje generalizacija modela i suzbija se mogućnost stvaranja prenaučenog modela. Isto tako čini model robusnijim na translacije i pomake. Dodatni benefit se očituje i u performansama modela, zbog smanjene dimenzionalnosti računalni sustav vrši manje matričnih proračuna što pak pospješuje brzinu učenja modela. U kontekstu smanjenja prenaučenosti i povećanja generalizacije modela koriste se i Dropout slojevi čija je zadaća nasumično isključivanje zadanog postotka ulaznih neurona kako bi se smanjio njihov individualni utjecaj. Flatten sloj služi za pretvaranje višedimenzionalnih karakterističnih mapa u jednodimenzionalni vektor kako bi se omogućilo povezivanje sa slojevima potpuno povezane mreže. Konkretno, ako su ulazne karakteristične mape višedimenzionalne, poput 2D mapa dobivenih na izlazima konvolucijskih slojeva, Flatten sloj će izravnati te mape u jedan vektor, čime se gube prostorne informacije, ali se omogućuje dovođenje izlaza iz prethodnog sloja na ulaz potpuno povezanih slojeva koji očekuju jednodimenzionalne ulaze. Flatten sloj, dakle, osigurava da izlaz prethodnih slojeva bude u obliku pogodnom za obradu u potpuno povezanim slojevima, gdje svaki element vektora predstavlja značajku koju će mreža koristiti kako bi razriješila zadaće klasifikacije, ili u drugim primjenama regresije. Dense slojevi, odnosno potpuno povezani slojevi povezuju sve ulazne jedinice (neurone) s izlaznim jedinicama u prethodnom sloju. Dense slojevi su često korišteni kao završni slojevi u neuronskim mrežama za klasifikaciju. Izlazi Dense sloja obično predstavljaju vjerojatnosti različitih klasa u problemu klasifikacije, a aktivacijska funkcija koja se koristi na posljednjem sloju je obično softmax funkcija koja pretvara izlaze u vjerojatnosti. Aktivacijska funkcija korištena u posljednjem sloju je upravo gore spomenuti softmax. Softmax funkcija je definirana formulom:

$$\sigma(y_i) = \left(\frac{e^{y_i}}{\sum_j e^{y_j}} \right) \quad j = 1, \dots, n \quad (3.1)$$

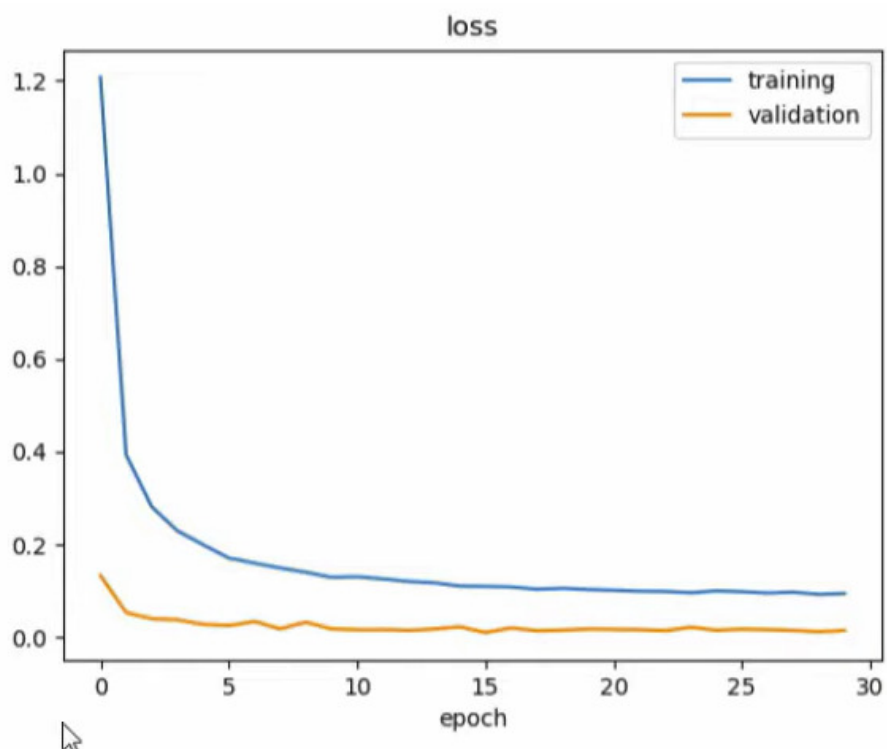
Motivacija za odabir ove aktivacijske funkcije je njena učinkovitost kad se kao metoda izračuna pogreške, odnosno funkcija gubitka koristi *categorical crossentropy*. (Dunne i Campbell, 1997) Dunne i Campbell su u svom radu "O uparivanju softmax aktivacijske funkcije i cross-entropy funkcije penalizacije", potaknuli hipotezom da postoji prirodno uparivanje te dvije funkcije pokušali empirijski potvrditi tu pretpostavku te su ostvarili izrazito pozitivne rezultate.

Model koristi Adaptive Moment Estimation optimizator, skraćeno Adam. Ovaj optimizator je zapravo proširenje metode stohastičkog gradijentnog spusta. Adam kombinira prednosti dviju drugih tehnika optimizacije: AdaGrad, koja prilagođava stopu učenja za svaki parametar na temelju njihovih prošlih gradijenata, i RMSprop, koji koristi eksponencijalno smanjujuće promjenjive prosjeke kvadrata povijesnih gradijenata. Osnovna ideja iza Adama je održavanje prilagodljive stope učenja za svaki parametar estimirajući prvi moment (srednju vrijednost) i drugi moment (necentriranu varijancu) gradijenata. Ta prilagodba pomaže procesu optimizacije da brže i stabilnije konvergira. (Kingma i Ba, 2014)

Treniranje modela se provodilo kroz 30 epoha, no praćenjem podataka o preciznosti modela postaje jasno da se model mogao trenirati i značajno kraće. U priloženim grafikonima se jasno vidi da već nakon 10 do 15 epoha model postiže preciznost veću od 95%.



Slika 3.5: Prikaz vrijednosti izračunate preciznosti kroz epohe.



Slika 3.6: Prikaz vrijednosti funkcije gubitka kroz epohe.

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 previdio 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 prepozna 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.

3.6. Testiranje sustava

4. Rezultati

Početni zahtjevi koji su stavljeni pred sustav su se ponajviše bazirali na brzinu i točnost klasifikacije. Cilj je bio ostvariti točnu klasifikaciju u čim kraćem vremenu. Rezultati klasifikacije postignuti uporabom kreiranog modela su izrazito povezani s veličinom podskupa skupa podataka za učenje koji je sadržavao primjere klasificirane klase. Tako primjerice za znakove koji su u *dataset-u* bili reprezentirani s više od 1000 primjera model primjere u velikoj većini slučajeva klasificira ispravno i sa sigurnošću većom ili jednakom od 99%. Primjere znakova koji su bili znatno manje zastupljeni u skupu podataka za učenje model i dalje, uglavnom, ispravno klasificira no sa znatno manjom sigurnošću, tek nešto iznad zadanog praga od 90% sigurnosti. Vrijedi napomenuti da je navedeni prag upravo uveden kako bi se čim više smanjio broj lažno pozitivnih klasifikacija koje je model radio. Prilikom testiranja modela u uvjetima u kojima je klasifikacijski prag sigurnosti bio niži od 75% model je često klasificirao ljudska lica, vozila i ostale primjere koji nisu pripadali klasi prometnih znakova. Brzina s kojom model donosi odluke je također dovoljna za klasifikaciju u stvarnom vremenu, odnosno red veličine nekoliko desetaka milisekundi. Naučeni model koji se koristi u klasifikaciji je poprilično agnostičan prema poremećajima osvjetljenja i kontrasta koji su uneseni tijekom ispitivanja sustava te uspješno klasificira čak i u takvim uvjetima.

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

Rad opisuje jednu moguću implementaciju sustava za klasifikaciju prometnih znakova u stvarnom vremenu. Razrađuju se primjene i zahtjevi sustava, opisuju se ranija ostvarenja u tom području i definira se pristup rješenju problema klasifikacije. Sustav klasificira prometne znakove uporabom jedne konvolucijske neuronske mreže koja se sastoji od 12 slojeva. Za potrebe treniranja neuronske mreže korišteni su razni pristupi pretprocesiranja fotografija iz skupa podataka za učenje kako bi se povećao broj ulaznih podataka koje model ima na raspolaganju za učenje ali i učinio model otpornijim na poremećaje u osvjetljenju i kontrastu do kojih može doći na ulazu. Istrenirani model se koristi u testnoj aplikaciji koja konstantno uzima slike s *web-kamere* prijenosnog računala, zatim te slike pokušava klasificirati te ukoliko s određenom sigurnošću uspije klasificirati fotografiju ispisuje rezultat na zaslon prijenosnog računala.

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

The paper describes a possible implementation of a real-time traffic sign classification system. The applications and requirements of the system are discussed, previous achievements in the field are described, and an approach to solving the classification problem is defined. Created system classifies traffic signs using a single convolutional neural network consisting of 12 layers. Various preprocessing approaches were used to augment the training dataset, increasing the number of input data available for the model to learn from and making the model more robust to disturbances in lighting and contrast that may occur at the input. The trained model is used in a test application that continuously captures images from a laptop's webcam, attempts to classify those images, and if successful with a certain confidence, displays the result on the laptop's screen.

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

LITERATURA

- Saad Albawi, Tareq Abed Mohammed, i Saad Al-Zawi. Understanding of a convolutional neural network. U *2017 International Conference on Engineering and Technology (ICET)*, stranice 1–6, 2017. doi: 10.1109/ICEngTechnol.2017.8308186.
- Tayfun Celebi, Ibraheem Shayea, Ayman A. El-Saleh, Sawsan Ali, i Mardeni Roslee. Histogram equalization for grayscale images and comparison with opencv library. U *2021 IEEE 15th Malaysia International Conference on Communication (MICC)*, stranice 92–97, 2021. doi: 10.1109/MICC53484.2021.9642082.
- Dan Cireşan, Ueli Meier, Jonathan Masci, i Jürgen Schmidhuber. Multi-column deep neural network for traffic sign classification. *Neural Networks*, 32:333–338, 2012. ISSN 0893-6080. doi: <https://doi.org/10.1016/j.neunet.2012.02.023>. URL <https://www.sciencedirect.com/science/article/pii/S0893608012000524>. Selected Papers from IJCNN 2011.
- Rob A Dunne i Norm A Campbell. On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function. U *Proc. 8th Aust. Conf. on the Neural Networks, Melbourne*, svezak 181, stranica 185. Citeseer, 1997.
- Diederik P Kingma i Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Smit Mehta, Chirag Paunwala, i Bhaumik Vaidya. Cnn based traffic sign classification using adam optimizer. U *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, stranice 1293–1298, 2019. doi: 10.1109/ICCS45141.2019.9065537.
- Richard E Turner. Lecture 14: Convolutional neural networks for computer vision, 2014.

Fatin Zaklouta, Bogdan Stanciulescu, i Omar Hamdoun. Traffic sign classification using k-d trees and random forests. U *The 2011 International Joint Conference on Neural Networks*, stranice 2151–2155, 2011. doi: 10.1109/IJCNN.2011.6033494.

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
3.5. Prikaz vrijednosti izračunate preciznosti kroz epohe.	11
3.6. Prikaz vrijednosti funkcije gubitka kroz epohe.	11