

Neuronska mreža za prepoznavanje automobilskih oznaka

Kuveždić, Domagoj

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:723724>

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

Download date / Datum preuzimanja: **2024-12-23**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

DOMAGOJ KUVEŽDIĆ

NEURONSKA MREŽA ZA PREPOZNAVANJE AUTOMOBILSKIH OZNAKA

Diplomski rad

Pula, rujan, 2024.

Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

DOMAGOJ KUVEŽDIĆ

NEURONSKA MREŽA ZA PREPOZNAVANJE AUTOMOBILSKIH OZNAKA

Diplomski rad

JMBAG: 0303082792, redoviti student

Studijski smjer: Informatika

Kolegij: Teorija informacija

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacijske znanosti

Znanstvena grana: Informacijski sustavi i informatologija

Mentor: izv. prof. dr. sc. Nicoletta Saulig

Komentor: izv. prof. dr. sc. Željka Tomasović

Pula, rujan, 2024.



IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani Domagoj Kuveždić, kandidat za magistra informatike ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student
Domagoj Kuveždić

U Puli, rujan 2024. godine



IZJAVA
o korištenju autorskog djela

Ja, Domagoj Kuveždić dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom „Neuronska mreža za prepoznavanje automobilskih oznaka“ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama. Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, rujan 2024. godine

Potpis

Sadržaj

1. Uvod	1
1.1 Motivacija	1
1.2 Ciljevi rada	1
1.3 Organizacija rada.....	1
2. Pregled literature	3
2.1 Automobilski sustavi za prepoznavanje oznaka automobila	3
2.2 Neuronske mreže u prepoznavanju oznaka automobila	3
2.3 Dosadašnja istraživanja i rezultati.....	3
3. Metodologija	5
3.1 Skup podataka	5
3.2 Prikupljanje i priprema podataka.....	7
3.3 Arhitektura neuronske mreže	8
3.4 Postupak učenja i evaluacija	15
3.5 Kriteriji evaluacije performansi	15
4. Implementacija.....	16
4.1 Odabir programskog okruženja i biblioteka	16
4.2 Postupak učenja i hiperparametri.....	16
4.3 Evaluacija performansi.....	17
5. Rezultati i analiza.....	18
5.1 Analiza performansi neuronske mreže	18
5.2 Poboljšanja neuronske mreže za prepoznavanje oznaka automobila.....	20
5.2.1 Augmentacija slike	20
5.2.2 Poboljšani model	22
5.2.3 Stopa učenja	24
5.2.4 Rano zaustavljanje.....	25
5.3 Rezultati i analiza unaprijeđenog modela.....	25
5.3.1 Točnost i gubitak	27
5.3.2 Podtreniranost	27
5.3.3 Nastavak treniranja	28
6. Usporedba s <i>transfer learning</i> metodama za prepoznavanja oznaka automobila	32
6.1 ResNet152V2.....	32
6.2 VGG16.....	36
6.3 Xception	38
6.4 Zaključak usporedbe <i>transfer learning</i> metoda	43
7. Utjecaj parametara na performanse	45

7.1 Utjecaj stope učenja na točnost	45
8. Zaključak	47
9. Literatura	48
10. Prilozi.....	51
10.1 Popis slika.....	51
10.2 Popis tablica.....	51
10.3 Popis grafova	51

1. Uvod

1.1 Motivacija

U svijetu automobila, prepoznavanje oznaka automobila, kao što su proizvođač i model automobila, igra ključnu ulogu u mnogim područjima, uključujući marketinške analize, prometno planiranje te sigurnosne i nadzorne sustave. S napretkom tehnologije, pojavila se potreba za automatskim sustavima koji mogu precizno identificirati proizvođača i model vozila na temelju slika.

Ovaj rad motiviran je potrebom za razvojem učinkovitog sustava za prepoznavanje automobilskih oznaka pomoću neuronskih mreža. Pojam "oznaka" ovdje obuhvaća ključne karakteristike automobila, kao što su proizvođač i model, koji omogućuju točnu identifikaciju. Korištenjem dubokog učenja, neuronske mreže imaju potencijal automatski naučiti relevantne značajke iz slika, omogućujući precizno prepoznavanje tih oznaka bez potrebe za ručnim definiranjem značajki.

1.2 Ciljevi rada

Osnovni ciljevi ovog rada su: istraživanje primjene neuronskih mreža u prepoznavanju oznaka automobila na temelju slika, ispitivanje utjecaja različitih arhitektura neuronskih mreža i pristupa u učenju na performanse sustava za prepoznavanje. Krajnji cilj je razvijanje učinkovitog sustava za automatsko prepoznavanje oznaka automobila i evaluacija njegove krajnje točnosti. Kroz ostvarenje navedenih ciljeva, ovaj rad nastoji unaprijediti tehnologiju prepoznavanja oznaka automobila te pružiti nove uvide u primjenu dubokog učenja u domeni računalnog vida, s naglaskom na stvarne izazove prepoznavanja automobila u složenim okruženjima.

1.3 Organizacija rada

Ovaj diplomski rad organiziran je na sljedeći način: poglavlje „*Pregled literature*“ pruža pregled postojećih pristupa prepoznavanju oznaka automobila, s fokusom na primjenu konvolucijskih neuronskih mreža. Poglavlje „*Metodologija*“ opisuje

metodologiju korištenu u istraživanju, uključujući prikupljanje podataka, pripremu podataka i arhitekturu neuronske mreže. Poglavlje „*Implementacija*“ detaljno prikazuje implementaciju sustava za prepoznavanje oznaka automobila, s posebnim naglaskom na izbor alata i postupak učenja. Poglavlje „*Rezultati i analiza*“ prezentira rezultate eksperimenata i analizira performansi sustava. Poglavlje „*Usporedba s transfer learning metodama za prepoznavanja oznaka automobila*“ uspoređuje različite *transfer learning* modele te njihove performanse na skupu podataka.

2. Pregled literature

2.1 Automobilski sustavi za prepoznavanje oznaka automobila

Automobilski sustavi za prepoznavanje proizvođača automobila igraju ključnu ulogu u različitim aplikacijama, uključujući sigurnosne sustave, pametna prometna rješenja i marketinške analize. Takvi sustavi često zahtijevaju preciznu segmentaciju i izvlačenje značajki, što ih može učiniti osjetljivima na varijacije u osvjetljenju, pozadini i položaju automobila.

2.2 Neuronske mreže u prepoznavanju oznaka automobila

U posljednjem desetljeću, neuronske mreže, posebno konvolucijske neuronske mreže, su značajno unaprijedile područje računalnog vida. Konvolucijske neuronske mreže poput VGG, ResNet i Inception, pokazali su izvrsne rezultate u zadacima prepoznavanja objekata, uključujući i prepoznavanje automobilskih oznaka.¹ Korištenjem složenih arhitektura s više konvolucijskih slojeva i velikih skupova podataka, konvolucijske neuronske mreže mogu automatski učiti hijerarhijske značajke iz slika, od jednostavnih oblika do složenih objekata. Ova sposobnost čini ih obećavajućim rješenjem za prepoznavanje oznaka automobila, s potencijalom da pruže visoku točnost čak i u složenim scenarijima. Međutim, treniranje dubokih neuronskih mreža zahtijeva veliku računalnu snagu i velike skupove visokokvalitetnih podataka. Također, izazov predstavlja mogućnost prilagodbe modela na različite svakodnevne situacije u prometu, kao što su loši vremenski uvjeti i velike sličnosti između dva modela automobila.

2.3 Dosadašnja istraživanja i rezultati

Dosadašnja istraživanja u području prepoznavanja oznaka automobila obuhvaćaju širok spektar pristupa. Studije su često usmjerene na različite aspekte

¹ Donny Avianto, Agus Harjoko, Afiahayati, *CNN-Based Classification for Highly Similar Vehicle Model Using Multi-Task Learning*, Journal of Imaging, DOI: 10.3390/jimaging8110293, 2022.

problema, uključujući arhitekturu mreže, kvalitetu skupa podataka i evaluacijske metrike. Istraživanje poput „A Large-Scale Car Dataset for Fine-Grained Categorization and Verification“² koristi veliki skup podataka od 150,000 slika automobila raznih vrsta, snimljenih iz različitih kutova. Automobili su klasificirani prema godini proizvodnje i različitim dijelovima vozila, kao što su prednja i zadnja svjetla, te svjetla za maglu. U ovom radu autori potiču istraživačku zajednicu na daljnje istraživanje klasifikacije automobila, predstavljajući CompCars dataset, koji nudi jedinstvene karakteristike poput većeg broja podkategorija, hijerarhijske strukture i velikog broja slika. Najveća točnost postignuta je kombinacijom modela konvolucijske neuronske mreže i tehnikom izvlačenja značajki.

„CNN-Based Classification for Highly Similar Vehicle Model Using Multi-Task Learning“³ istražuje problem klasifikacije automobila korištenjem osnovne arhitekture konvolucijske neuronske mreže kao ekstraktora značajki. U radu autora Donny Avianto, Agus Harjoko i Afiahayati, nova arhitektura omogućava istovremenu klasifikaciju dva zadatka – prepoznavanje proizvođača i modela vozila. Najbolji rezultati postignuti su s VGG-16 MT modelom, koji je ostvario impresivnu točnost od 98.73% u klasifikaciji proizvođača automobila, te točnost od 97.67% u klasifikaciji modela automobila. Autori su također zaključili kako su klase automobila Agya–Ayla bile najzahtjevnije za klasifikaciju zbog velike sličnosti između ovih dviju klasa.

Istraživanja kao što su „CNN-Based Classification for Highly Similar Vehicle Model Using Multi-Task Learning“⁴, te „Klasifikacija automobila koristeći transferirano učenje na ResNet arhitekturi neuronske mreže“⁵ su pokazala da neuronske mreže, posebno konvolucijske neuronske mreže, postižu impresivne rezultate u prepoznavanju oznaka automobila.

² Linjie Yang, Ping Luo, Chen Change Loy, Xiaoou Tang, *A Large-Scale Car Dataset for Fine-Grained Categorization and Verification*, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

³ Donny Avianto, Agus Harjoko, Afiahayati, *CNN-Based Classification for Highly Similar Vehicle Model Using Multi-Task Learning*, Journal of Imaging, DOI: 10.3390/jimaging8110293, 2022.

⁴ Donny Avianto, Agus Harjoko, Afiahayati, *CNN-Based Classification for Highly Similar Vehicle Model Using Multi-Task Learning*, Journal of Imaging, DOI: 10.3390/jimaging8110293, 2022.

⁵ Stjepan Ložnjak, Tin Kramberger, Ivan Cesar, Renata Kramberger, *Klasifikacija automobila koristeći transferirano učenje na resnet arhitekturi neuronske mreže*, 2020.

3. Metodologija

3.1 Skup podataka

Za potrebe ovog rada korišten je skup podataka sa slikama automobila različitih proizvođača i modela, koji sadrži ukupno 3736 slika. Od toga, 3005 slika je u trening skupu, a preostala 731 slika je u testnom skupu. Skup podataka obuhvaća raznolike scenarije snimanja, uključujući različite uvjete osvjetljenja, izgleda okoline i položaja automobila u odnosu na kameru. Cilj je osigurati da neuronska mreža dobro generalizira i da je sposobna prepoznati proizvođača i model automobila u različitim situacijama. Skup podataka izvorno sastojao se od 48 klasa, no zbog izrazitog malog broja slika u 5 klasa sa najmanjim brojem slika, iste su uklonjene kako ne bi značajno utjecale na rezultate treniranja. Tako konačni skup podataka sadrži 43 klase, uključujući popularne proizvođače automobila kao što su Toyota, Suzuki, Kia, Honda, etc. Sljedeći primjeri prikazuju nekoliko slika iz trening i testnog skupa (*Slika 1, Slika 2*).



Slika 1 - Primjer automobila Suzuki Swift iz trening skupa podataka⁶

⁶ Ali Mohsin, *Vehicle images dataset for make and model recognition.*, Mendeley Data, V1, DOI: 10.17632/hj3vvx5946.1, 2022.



Slika 2 - Primjer automobila Honda Civic iz testnog skupa podataka⁷

Jedan od dodatnih izazova u klasifikaciji predstavljaju fotografije na kojima je automobil djelomično vidljiv, bilo zbog prepreke ili zato što je izvan kadra. Sljedeći primjeri ilustriraju ovu problematiku (*Slika 3*).



Slika 3 - Djelomično prekriveni automobili⁸

Kako bi se model učinio otpornijim na varijacije u osvjetljenju, primijenjene su različite tehnike augmentacije podataka. U priloženim primjerima vidljivo je kako različite razine osvjetljenja, poput direktnog sunčevog svjetla i noćnih uvjeta drastično utječu na kvalitetu slika automobila. Jaka svjetlost, odsjaj ili nedostatak svjetla otežavaju prepoznavanje ključnih karakteristika automobila, što predstavlja značajan izazov za algoritme klasifikacije koji se oslanjaju na točne vizualne informacije. Sljedeći primjer prikazuje slike u različitim osvjetljenjima (*Slika 4*).

⁷ Ali Mohsin, *Vehicle images dataset for make and model recognition*, Mendeley Data, V1, DOI: 10.17632/hj3vvx5946.1, 2022.

⁸ Izvor: Ali Mohsin, *Vehicle images dataset for make and model recognition*, Mendeley Data, V1, DOI: 10.17632/hj3vvx5946.1, 2022.



Slika 4 - Automobili u različitim osvjetljenjima⁹

3.2 Prikupljanje i priprema podataka

Skup podataka slika vozila korišten u ovom radu prikupljen je s platforme „Mendeley Data“ autora Mohsina Alija¹⁰. Prije korištenja modela, slike su prošle proces augmentacije kako bi se povećala njihova raznolikost i kvaliteta samoga modela. Augmentacija je uključivala operacije poput skaliranja, rotacije, pomicanja širine i visine, smicanja i uvećavanja.

Kod koji je korišten pri augmentaciji slika:

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=30,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True
)
```

⁹ Izvor: Ali Mohsin, *Vehicle images dataset for make and model recognition*, Mendeley Data, V1, DOI: 10.17632/hj3vvx5946.1, 2022.

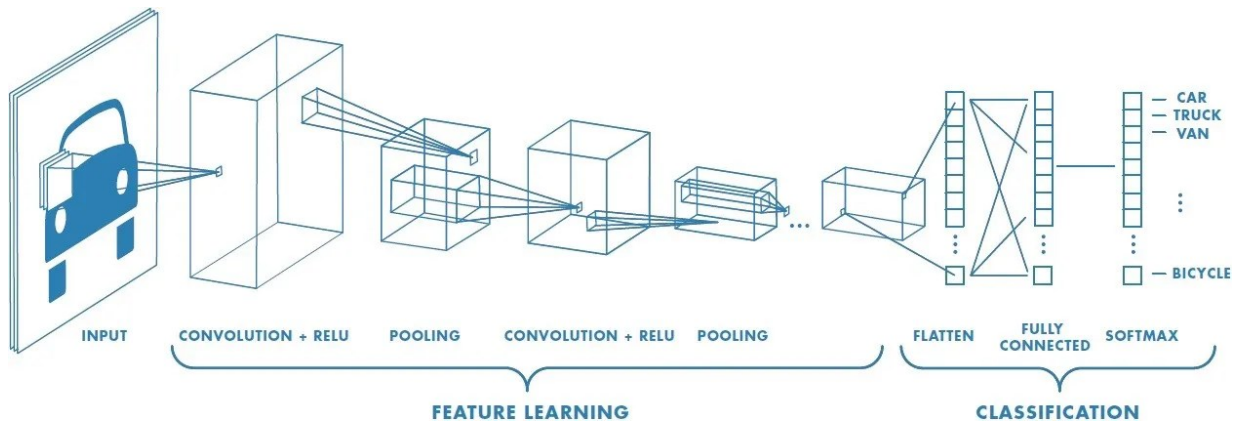
¹⁰ Ali Mohsin, *Vehicle images dataset for make and model recognition*, Mendeley Data, V1, DOI: 10.17632/hj3vvx5946.1, 2022.

Uvođenjem horizontalnog pomaka slike postiže se umjetno proširivanje postojećeg skupa podataka. Time se dobivaju primjeri slika automobila koji su zrcalno prikazani, što omogućuje modelu da nauči opće značajke i bolje prepozna je automobile bez obzira na njihovu orijentaciju. Skup podataka po broju slika ostaje nepromijenjen, ali model „vidi“ više različitih varijanti istih slika kroz proces treninga prolascima kroz epohe.

3.3 Arhitektura neuronske mreže

Konvolucijske neuronske mreže su vrsta dubokih neuronskih mreža specijalizirana za obradu podataka s rešetkastom strukturom, poput slika. Inovativna arhitektura konvolucijskih neuronskih mreža, sastavljena od konvolucijskih, *pooling* i potpuno povezanih slojeva, omogućuje im automatsko učenje i izvlačenje relevantnih značajki iz slika.¹¹ Zahvaljujući ovoj sposobnosti, konvolucijske neuronske mreže su postale nezaobilazni alat u području računalnog vida, zbog pronalaženja široke primjene u zadacima kao što su klasifikacija slika, detekcija objekata i segmentacija slika. Na slici u nastavku prikazan je primjer arhitekture konvolucijske neuronske mreže (*Slika 5*).

¹¹ Andrew Ng, *Convolutional Neural Networks*, UFLDL Tutorial, Stanford University, <http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/>, 2013.



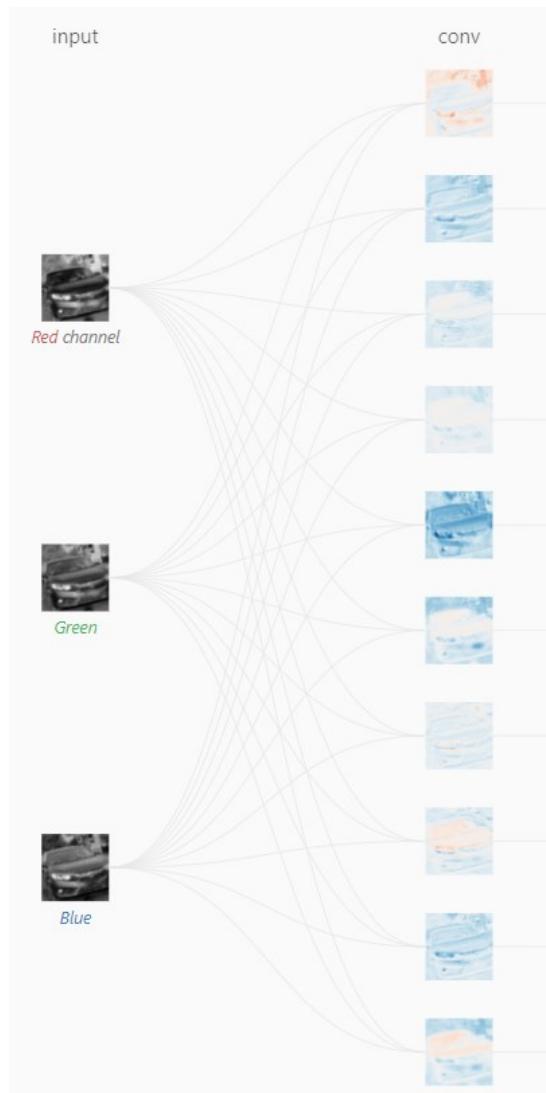
Slika 5 - Arhitektura konvolucijske neuronske mreže. Sastoji se od sljedećih slojeva: ulazni sloj, konvolucijski slojevi, pooling slojeva, potpuno povezani sloj, te izlazni sloj¹²

Ključne komponente konvolucijskih neuronskih mreža uključuju:

- **Konvolucijske slojeve**, koji primjenjuju operacije konvolucije na ulazne podatke koristeći filtere (jezgre) koji se pomiču preko slike dok svaki filter detektira specifične značajke poput rubova, tekstura ili oblika. Konvolucije omogućuju prostornu invarijabilnost, što znači da mreža može prepoznati značajke bez obzira na njihov položaj na slici.¹³ Sljedeća slika prikazuje primjer prolaska filtera kroz sliku automobila (*Slika 6*).

¹² Sumit Saha, *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*, Towards Data Science, <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

¹³ Nanning Zheng, George Loizou, Xiaoyi Jiang, Xuguang Lan, Xuelong Li, *Computer vision and pattern recognition*, International Journal of Computer Mathematics, DOI: 10.1080/00207160701303912, 2007., str. 355.-357.



Slika 6 - proces konvolucije u neuronskim mrežama, gdje se ulazna slika razlaže na crveni, zeleni i plavi kanal, a zatim prolazi kroz konvolucijske filtere koji generiraju mape značajki.¹⁴

- **Aktivacijske funkcije** (npr. *ReLU - Rectified Linear Unit*) koji se primjenjuju nakon svakog konvolucijskog sloja u neuronskoj mreži. Uvođenjem nelinearnosti, aktivacijske funkcije omogućuju modelu da uči složenije odnose između podataka, čime se povećava njegova sposobnost rješavanja kompleksnih problema.¹⁵ Na sljedećoj fotografiji nalaze se često korištene aktivacijske funkcije (Slika 7).

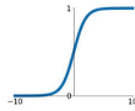
¹⁴ Jay Wang, Robert Turko, Omar Shaikh, Haekyu Park, Nilaksh Das, Fred Hohman, Minsuk Kahng, Polo Chau, *CNN Explainer: Learning Convolutional Neural Networks with Interactive Visualization*, <https://poloclub.github.io/cnn-explainer/>, 2020.

¹⁵ Nanning Zheng, George Loizou, Xiaoyi Jiang, Xuguang Lan, Xuelong Li, *Computer vision and pattern recognition*, International Journal of Computer Mathematics, DOI: 10.1080/00207160701303912, 2007.

Activation Functions

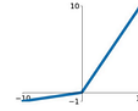
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



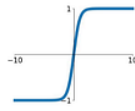
Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$

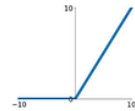


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

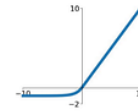
ReLU

$$\max(0, x)$$



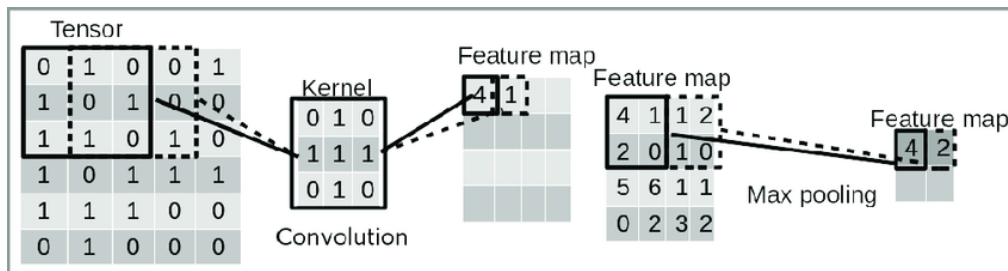
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Slika 7 - Različite aktivacijske funkcije i njihovi grafikoni¹⁶

- **Pooling slojeve**, poput „max pooling“ sloja, koji imaju ključnu ulogu u smanjenju dimenzionalnosti podataka. Oni to postižu sažimanjem informacija iz susjednih piksela, pri čemu se izdvajaju najvažnije značajke. Ovaj proces ne samo da smanjuje računalnu složenost modela, već ga čini otpornijim na male pomake u ulaznim podacima.¹⁷



Slika 8 - Primjer operacije konvolucije i max pooling-a¹⁸

- **Potpuno povezani slojevi** koji klasificiraju podatke nakon što konvolucijski i *pooling* slojevi izvuku relevantne značajke iz njih. Izlazi iz *pooling* slojeva spljošte se u jedan dugi vektor koji se zatim prosljeđuje kroz jedan ili više potpuno povezanih slojeva. Ovi

¹⁶ Shruti Jadon, *Introduction to Different Activation Functions for Deep Learning*, Medium, <https://medium.com/@shrutijadon/survey-on-activation-functions-for-deep-learning-9689331ba092>, 2018.

¹⁷ Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, Genetic Programming and Evolvable Machines, DOI: 10.1007/s10710-017-9314-z, 2018., str. 339., 340.

¹⁸ Izvor: Jan Schafer, Philipp Schmitt, Mark W. Hlawitschka, and Hans-Jorg Bart, *Measuring Particle Size Distributions in Multiphase Flows Using a Convolutional Neural Network*, Chemie Ingenieur Technik, DOI: 10.1002/cite.201900099, 2019.

slojevi, temeljeni na naučenim težinama koje se množe sa ulaznim podacima te izvršavaju konačnu klasifikaciju, odnosno određuju kojoj klasi pripada ulazni podatak.

Osim osnovnih komponenti, konvolucijske neuronske mreže često koriste dodatne tehnike za poboljšanje performansi poput *dropout-a* koja nasumično isključuju neurone¹⁹, te *batch normalization* koja normalizira ulaze svakog sloja, pomaže u sprječavanju pretrniranosti i poboljšava generalizacijske sposobnosti modela.²⁰

Za ažuriranje težina modela koriste se optimizacijski algoritmi poput stohastičkog gradijentnog spusta (SGD) i Adama. Stohastički gradijentni spust jednostavan je i učinkovit, dok Adam adaptivno podešava stopu učenja za svaki parametar, što može ubrzati konvergenciju, što je trenutak kada mreža "nauči" zadatak na najbolji mogući način.

Funkcija gubitka, kao što je *categorical crossentropy* za probleme klasifikacije s dvije i više klasa, mjere koliko su predviđanja modela različita od stvarnih vrijednosti.²¹

Predloženi model koristi konvolucijske slojeve, ReLU aktivaciju i *max pooling* za ekstrakciju i smanjenje prostorne dimenzije slike sažimanjem informacija iz susjednih piksela.

```
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(100, 100,
3)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
```

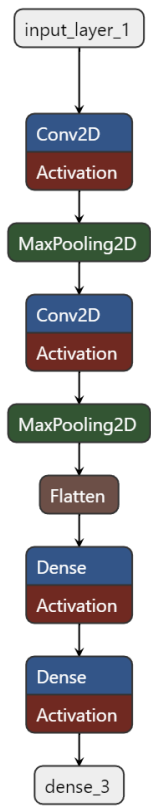
¹⁹ Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, Genetic Programming and Evolvable Machines, DOI: 10.1007/s10710-017-9314-z, 2018., str. 258.

²⁰ Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, Genetic Programming and Evolvable Machines, DOI: 10.1007/s10710-017-9314-z, 2018., str. 317.-321.

²¹ Ivan Marasović, Goran Majič, Ivan Škalić, Željka Tomasović, *Indoor Localization of Industrial IoT Devices and Applications Based on Recurrent Neural Networks*, Journal of Communications Software and Systems, 2024, str. 142.

```
Dense(128, activation='relu'),
Dense(num_classes, activation='softmax')
])
```

Sljedeća slika prikazuje strukturu početnog modela (Slika 9).



Slika 9 - Vizualni 2D prikaz početnog modela

U nastavku će biti razložen svaki pojedini sloj modela iz programskog koda, počevši od ulaznog sloja i analizirajući kako se podaci obrađuju kroz mrežu.

```
Conv2D(32, (3, 3), activation='relu', input_shape=(100, 100, 3))
```

Ulazna slika ima dimenzije 100x100 piksela s tri kanala, crvena, zelena, plava, preko koje konvolucijski sloj prolazi sa 32 filtera veličine 3x3, prepoznajući manje značajke ulazne slike. Aktivacijske funkcije unose nelinearnosti u neuronske mreže, omogućavajući im da stvaraju složenije odnose između podataka. ReLU (*Rectified Linear Unit*) funkcija postala je standardni izbor u mnogim vrstama neuronskih mreža, posebno u konvolucijskim neuronskim mrežama koje se koriste za obradu slika. Njena jednostavna definicija glasi da je izlaz jednak ulazu ako je ulaz pozitivan, inače je nula.

Matematički, ReLU se može izraziti kao: $f(x)=\max(0,x)$, gdje je x ulazna vrijednost za funkciju f , dok $\max(0, x)$ predstavlja funkciju koja bira veću vrijednost između 0 i x . Ovo je čini je računski učinkovitim, a istovremeno omogućuje mreži da uči složenije reprezentacije podataka. Za razliku od sigmoidnih ili tangens hiperboličnih funkcija, ReLU ne pati od problema zasićenja na pozitivnoj strani, što ubrzava poboljšanja rezultata tijekom treniranja.²²

`MaxPooling2D((2, 2))`

Nakon aktivacijske funkcije, *max pooling* sloj s dimenzijama 2×2 smanjuje dimenzije izlaza iz prethodnog sloja uzimajući maksimalnu vrijednost iz svakog filtera veličine 2×2 pomičući ih kroz sliku.

`Conv2D(64, (3, 3), activation='relu')`

Drugi konvolucijski sloj s 64 filtera veličine 3×3 prepoznaje složenije značajke slike, kao što su uzorci i veći oblici. Zatim se ponovno koristi ReLU aktivacija kako bi se dodala nelinearnost.

`MaxPooling2D((2, 2))`

Drugi *max pooling* sloj s prozorom veličine 2×2 ponovno smanjuje dimenzije izlaza iz prethodnog sloja, sažimajući informacije, što na kraju smanjuje računsku složenost.

`Flatten()`

Sloj za izravnavanje pretvara višedimenzionalni izlaz iz konvolucijskih slojeva u jednodimenzionalni vektor, koji se dalje šalje u potpuno povezane slojeve. Na taj način, podaci sa slike transformiraju se u oblik prikladan za klasifikaciju.

`Dense(128, activation='relu')`

Ovaj potpuno povezani sloj kombinira informacije iz prethodnog sloja i koristi ReLU aktivaciju za dodavanje nelinearnosti. Sloj određuje koje su karakteristike najvažnije za klasifikaciju.

²² Nanning Zheng, George Loizou, Xiaoyi Jiang, Xuguang Lan, Xuelong Li, *Computer vision and pattern recognition*, International Journal of Computer Mathematics, DOI: 10.1080/00207160701303912, 2007., str. 358.

Dense(num_classes, activation='softmax')

Završni potpuno povezani sloj s brojem jedinica jednakim broju klasa koristi softmax aktivaciju kako bi normalizirao izlaze tako da njihov zbroj bude 1. Ovo olakšava interpretaciju izlaza kao vjerojatnosti za svaku klasu. U konačnici, ovaj sloj generira vjerojatnost da slika pripada jednoj od dostupnih klasa.²³

3.4 Postupak učenja i evaluacija

Postupak učenja proveden je na pripremljenom skupu podataka korištenjem tehnike gradijentnog spusta s prilagodljivom stopom učenja, ovisno o potrebama. Nakon završetka procesa učenja, provedena je evaluacija performansi neuronske mreže na odvojenom skupu testnih podataka.

3.5 Kriteriji evaluacije performansi

Performanse neuronske mreže ocijenjene su na temelju nekoliko kriterija, uključujući točnost prepoznavanja oznaka automobila, otpornost na varijacije u uvjetima snimanja i gubitak. Drugim riječima, gubitak nam govori koliko je naš model pogriješio u svojim predviđanjima. Cilj treniranja neuronske mreže je minimalizirati ovu funkciju gubitka i pritom postići što je veću moguću točnost. Kriteriji evaluacije korišteni su kako bi se procijenila efikasnost i pouzdanost razvijenog sustava.

²³ Nanning Zheng, George Loizou , Xiaoyi Jiang , Xuguang Lan, Xuelong Li, *Computer vision and pattern recognition*, International Journal of Computer Mathematics, DOI: 10.1080/00207160701303912, 2007., str. 355.-364.

4. Implementacija

4.1 Odabir programskog okruženja i biblioteka

Pri implementaciji neuronske mreže za prepoznavanje oznaka automobila odabran je Python²⁴ kao primarni programski jezik. Odabrana je biblioteka TensorFlow²⁵, koja pruža bogat skup alata i modula za jednostavnu implementaciju dubokih neuronskih mreža, što olakšava razvoj i optimizaciju modela. Dodatno, korištena je biblioteka Keras²⁶ koja pruža visokorazinski API za izgradnju, treniranje i evaluaciju neuronskih mreža. Za korištenje navedenih aplikacija bilo je dovoljno preuzeti i instalirati Python sa službene stranice, dok su Tensorflow i Keras preuzeti i instalirani pomoću pip²⁷ sljedećim naredbama:

```
pip install tensorflow
pip install keras
```

4.2 Postupak učenja i hiperparametri

U procesu učenja primijenjen je optimizator Adam s dinamički prilagođavanom stopom učenja. Stopa učenja je detaljno opisana u sekciji 5.2.3. Tipične vrijednosti stope učenja kretat će se u rasponu od $1e-4$ što je jednako 0.0001 te do $1e-3$ što iznosi 0.001.

Hiperparametre određuje sam istraživač prije početka učenja, za razliku od parametara koje mreža sama uči tijekom treniranja. Hiperparametri mogu biti:

- **U konvolucijskim slojevima:** broj filtera, veličina filtera i pomak filtera
- **U slojevima sažimanja:** veličina filtera i pomak filtera
- **Prilikom augmentacije podataka:** rotacija, pomicanje, skaliranje i osvjetljenje
- **Stopa učenja** je također neizostavni hiperparametar koji može drastično utjecati na performanse modela

²⁴ Python je programski jezik koji se koristi za razvoj širokog spektra aplikacija, od web stranica do umjetne inteligencije, <https://www.python.org/>

²⁵ TensorFlow je Googleov alat za izgradnju i treniranje modela dubokog učenja. Izvor: <https://www.tensorflow.org/>

²⁶ François Chollet, Keras, 2015., Izvor: <https://keras.io/>

²⁷ Pip je alat za instalaciju paketa za Python. Pomoću Pip-a možemo instalirati pakete iz Python Package Indeksa i drugih indeksa. Izvor: <https://pypi.org/project/pip/>

Oni igraju ključnu ulogu u performansama modela i odabiru se eksperimentalno.

4.3 Evaluacija performansi

Nakon faze treniranja, izvršena je evaluacija performansi razvijenog modela na neovisnom skupu testnih podataka. Performanse su kvantificirane pomoću metrika točnosti i gubitka.

Za vizualizaciju rezultata korištena je Python biblioteka Matplotlib²⁸. Generirani su linijski grafovi koji prikazuju promjene točnosti i gubitka tijekom epoha treniranja, kako na skupu za treniranje tako i na testnom skupu. Kako bi se olakšala interpretacija rezultata, krivulje trening skupa prikazane su plavom bojom, dok su krivulje testnog skupa prikazane narančastom bojom. Ovi grafovi omogućavaju vizualnu analizu procesa učenja i identifikaciju potencijalnih problema kao što su pretreniranost i podtreniranost koji su objašnjeni detaljnije u sekciji 5.3.2.

²⁸ Matplotlib je Python biblioteka za stvaranje visokokvalitetnih grafičkih prikaza podataka. Izvor: <https://matplotlib.org/>

5. Rezultati i analiza

5.1 Analiza performansi neuronske mreže

Proveden je trening navedenog modela sa prethodno definiranim hiperparametrima, te stopom učenja koja iznosi 0.001 kako je prikazano unutar sljedećeg koda:

```
from tensorflow.keras.optimizers import Adam
model.compile(optimizer=Adam(learning_rate=1e-3),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Nakon treniranja modela kroz 100 epoha, prikazani su rezultati posljednjih 5 epoha:

...

Epoch 96/100

94/94 ————— **79s** 791ms/step - accuracy: 0.6105 - loss:
1.3283 - val_accuracy: 0.5089 - val_loss: 2.1939

Epoch 97/100

94/94 ————— **70s** 689ms/step - accuracy: 0.6179 - loss:
1.3370 - val_accuracy: 0.4843 - val_loss: 2.6456

Epoch 98/100

94/94 ————— **79s** 781ms/step - accuracy: 0.6260 - loss:
1.2730 - val_accuracy: 0.5280 - val_loss: 2.0953

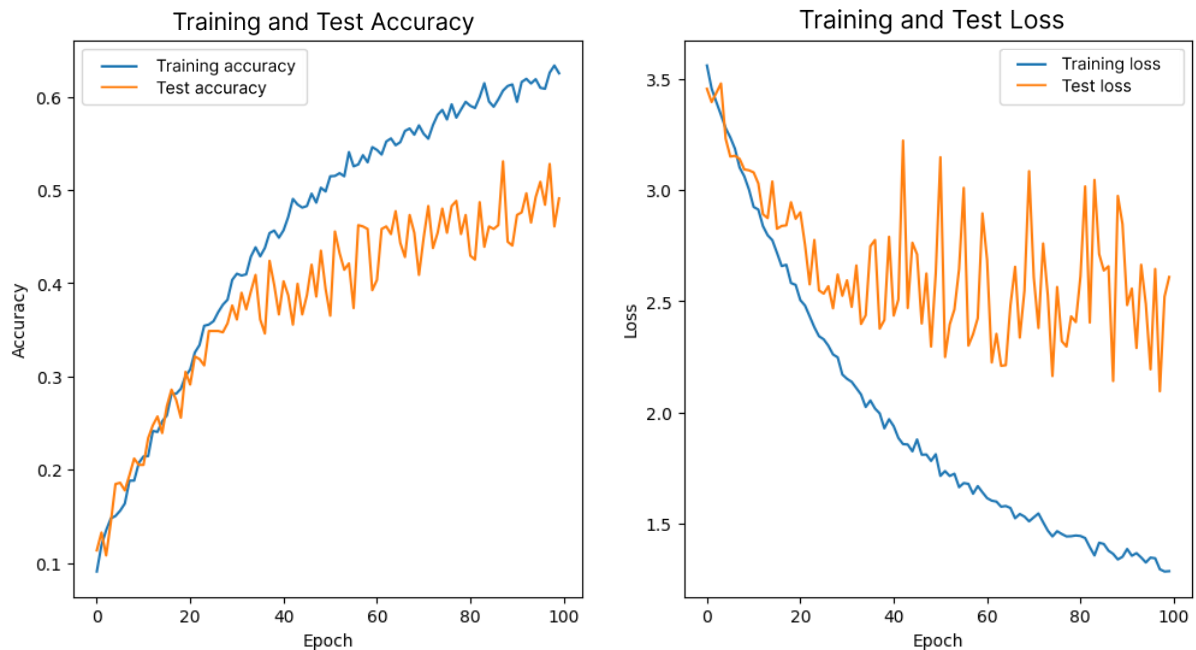
Epoch 99/100

94/94 ————— **70s** 690ms/step - accuracy: 0.6581 - loss:
1.2017 - val_accuracy: 0.4610 - val_loss: 2.5208

Epoch 100/100

94/94 ————— **78s** 777ms/step - accuracy: 0.6322 - loss:
1.2746 - val_accuracy: 0.4911 - val_loss: 2.6094

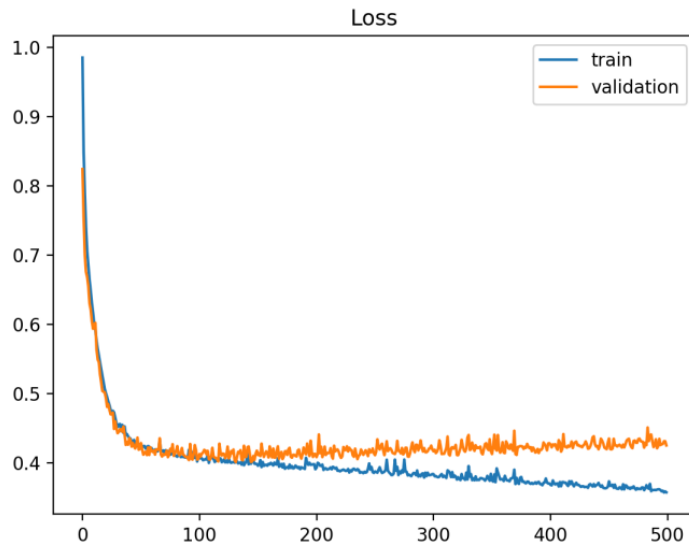
Iz dobivenih rezultata vidimo kako je model dosegao točnost od 63% na skupu za treniranje, te 49% na testnom skupu. Sljedeći graf prikazuje dobivene vrijednosti točnosti i gubitaka prilikom treninga i testa (*Graf 1*).



Graf 1 - Prikaz točnosti i gubitka nakon 100 epoha

Postupno povećavanje razlike točnosti i gubitka između skupa za treniranje i testnog skupa ukazuje na pojavu pretreniranosti, čestog problema u dubokom učenju. Modeli koji su pretrenirani dobro se generaliziraju na podacima na kojima su trenirani, ali slabo na novim, neviđenim podacima.

Tijekom treniranja, gubitak na skupu za treniranje postupno se smanjuje, što je znak da model uspješno uči iz podataka. Istodobno, gubitak na testnom skupu u početku opada, ali ubrzo dostiže minimum, te razlika između točnosti i gubitka na trening i testnom skupu postepeno raste, što ukazuje na pojavu pretreniranosti. Sljedeći graf prikazuje primjer pretreniranog modela (*Graf 2*).



Graf 2 - Primjer krivulja učenja za trening i validaciju koje pokazuju pretreniran model²⁹

5.2 Poboljšanja neuronske mreže za prepoznavanje oznaka automobila

S ciljem poboljšanja performansi modela, provest će se niz eksperimenata. Ti eksperimenti će uključivati proširenje tehnika augmentacije slika, dodavanje novih slojeva u postojeći model, prilagodbu stope učenja te implementaciju strategije ranog zaustavljanja.

5.2.1 Augmentacija slike

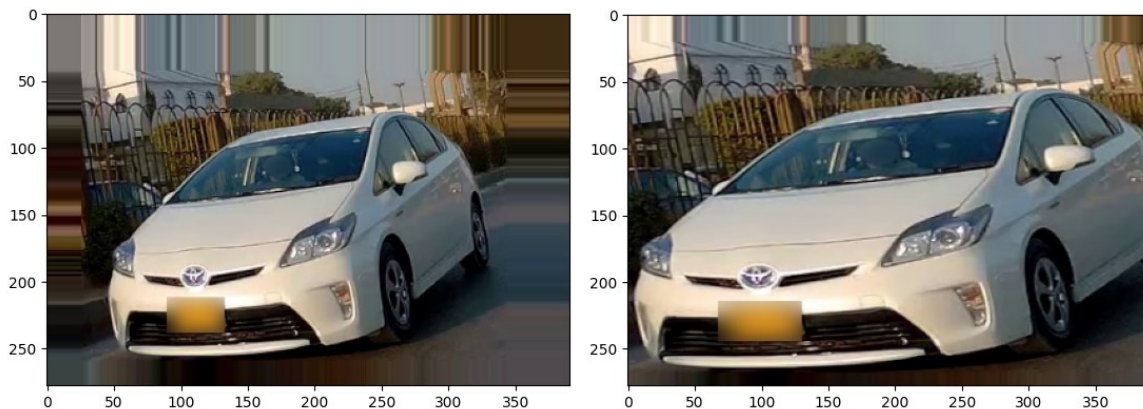
Kako bi se ublažila pretreniranost modela, povećati će se raznolikost podataka primjenom augmentacije. Konkretno, podaci će biti obogaćeni varijacijama u svjetlini i uvećavanju slike.

```
brightness_range=[0.8, 1.2],
zoom_range=[0.8, 1.2]
```

Primjenom raspona svjetline tijekom augmentacije slika, model se izlaže varijacijama u osvjetljenju koje su vrlo prisutne u stvarnim scenarijima. Na taj način, model uči prepoznavati objekte pod različitim uvjetima osvjetljenja, poput onih koji nastaju zbog

²⁹ Jason Brownlee, *Better Deep Learning*, Machine Learning Mastery, 2018., str. 29.

različitog doba dana, dok uvećavanje slike izlaže model različitoj udaljenosti automobila od pozicije kamere. Izlaganjem modela ovakvim proširivanjem skupa podataka poboljšava se učinkovitost modela u stvarnom svijetu. Sljedeći primjeri demonstriraju utjecaj raspona svjetlosti i uvećavanja tijekom augmentacije (Slika 10, Slika 11).



Slika 10 - Primjer slike sa različitim rasponom uvećavanja³⁰



Slika 11 - Primjer slike sa različitim rasponom svjetline³¹

³⁰ Ali Mohsin, *Vehicle images dataset for make and model recognition.*, Mendeley Data, V1, DOI: 10.17632/hj3vvx5946.1, 2022.

³¹ Ali Mohsin, *Vehicle images dataset for make and model recognition.*, Mendeley Data, V1, DOI: 10.17632/hj3vvx5946.1, 2022.

5.2.2 Pobješani model

Kako bi se poboljšala sposobnost modela da uči složenije obrasce, predlaže se povećanje njegove kompleksnosti dodavanjem većeg broja filtera u određenim slojevima i uvođenjem dodatnog sloja sažimanja. Iako povećanje broja parametara može pridonijeti boljoj generalizaciji, postoji i povećan rizik od pretreniranosti.

Kako bi se ublažio rizik od pretreniranosti, primijenit će se sljedeće tehnike regularizacije:

- **Batch normalization** je tehnika koja stabilizira proces treniranja normalizacijom vrijednosti aktivacija tako da imaju približno istu srednju vrijednost i varijancu koja nam govori koliko su vrijednosti aktivacija "razbacane" oko srednje vrijednosti.³²
- **Dropout**, koji nasumično isključuje određeni postotak neurona tijekom treninga, čime se smanjuje međuovisnost neurona i sprječava pretreniranost te značajno poboljšava generalizacijsku sposobnost modela. Jedna od najvećih prednosti *dropout-a* jest njegova široka primjena, budući da se može uspješno primijeniti na gotovo sve modele.³³ Kako Krizhevsky tvrdi, *dropout* otprilike udvostručuje broj iteracija potrebnih za konvergenciju.³⁴ To je blago negativni učinak *dropout* sloja, što zahtjeva dodatne resurse i više vremena za učenje, ali zato može dovesti do modela koji je robusniji i bolje generalizira, što je često važnije od brzine treniranja.

Nakon dodavanja slojeva u model, konačni model sadrži sljedeće slojeve:

```
model = Sequential()  
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(100,  
100, 3)))
```

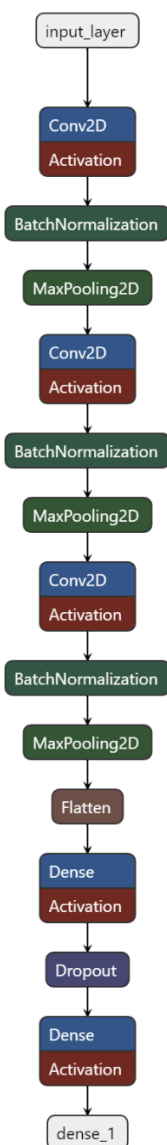
³² Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, Genetic Programming and Evolvable Machines, DOI: 10.1007/s10710-017-9314-z, 2018., str. 317.-320.

³³ Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, Genetic Programming and Evolvable Machines, DOI: 10.1007/s10710-017-9314-z, 2018., str. 258.-320.

³⁴ Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, DOI: 10.1145/3065386, Communications of the ACM, 2017., str. 6.

```
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))
```

Sljedeća slika prikazuje cjelovitu strukturu unaprijeđenog modela (Slika 12).



Slika 12 - Vizualni 2D prikaz unaprijeđenog modela

5.2.3 Stopa učenja

Stopa učenja kontrolira brzinu kojom model uči. Drugim riječima, kontrolira veličinu koraka kojim se ažuriraju težine modela nakon svake iteracije, primjerice na kraju svake skupine podataka za treniranje.³⁵ Smanjenjem stope učenja, model konvergira prema boljem rješenju izводеći manje iteracije. Na taj način izbjegava se prekoračenje optimalnih težina i pronalazi se precizniji minimum.³⁶ Sljedećim

³⁵ Jason Brownlee, *Better Deep Learning*, Machine Learning Mastery, 2018.

³⁶ Andrew Ng, *Machine Learning Yearning*, 2018.

programskim kodom promijenit će se stopa učenja, koja iznosi 0.0001 za razliku od prethodnog modela gdje je ona iznosila 0.001.

```
from tensorflow.keras.optimizers import Adam

model.compile(optimizer=Adam(learning_rate=1e-4),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

5.2.4 Rano zaustavljanje

Uvodi se rano zaustavljanje. Kako Goodfellow navodi, „rano zaustavljanje je oblik regularizacije koji se koristi kako bi se izbjeglo prenaučavanje prilikom treniranja modela iterativnim metodama, poput gradijentnog spusta.“³⁷

```
from tensorflow.keras.callbacks import EarlyStopping

early_stop = EarlyStopping(monitor='val_loss', patience=10)
```

Rano zaustavljanje funkcionira praćenjem performansi modela na zasebnom validacijskom skupu tijekom treninga. Ukoliko se točnost modela na ovom skupu prestane poboljšavati unutar unaprijed definiranog broja epoha (strpljenje), proces treniranja se prekida. Ova metoda omogućava modelu bolju generalizaciju na nove podatke te smanjuje rizik od pretreniranosti, čime se učinkovito koriste računalni resursi.

5.3 Rezultati i analiza unaprijeđenog modela

Nakon treniranja modela kroz 100 epoha, prikazani su rezultati posljednjih 5 epoha:

³⁷ Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, Genetic Programming and Evolvable Machines, DOI: 10.1007/s10710-017-9314-z, 2018., str. 246.-252.

...

Epoch 96/100

94/94 ————— **49s** 492ms/step - accuracy: 0.3673 - loss: 2.3224 - val_accuracy: 0.4131 - val_loss: 2.3292

Epoch 97/100

94/94 ————— **48s** 486ms/step - accuracy: 0.3731 - loss: 2.2422 - val_accuracy: 0.4309 - val_loss: 2.1169

Epoch 98/100

94/94 ————— **49s** 488ms/step - accuracy: 0.3805 - loss: 2.1897 - val_accuracy: 0.4323 - val_loss: 2.0975

Epoch 99/100

94/94 ————— **49s** 496ms/step - accuracy: 0.3786 - loss: 2.2396 - val_accuracy: 0.4802 - val_loss: 2.0009

Epoch 100/100

94/94 ————— **49s** 490ms/step - accuracy: 0.4060 - loss: 2.1950 - val_accuracy: 0.4596 - val_loss: 2.0485

Sljedeći graf prikazuje dobivene vrijednosti točnosti i gubitaka prilikom treninga i testa (*Graf 3*).



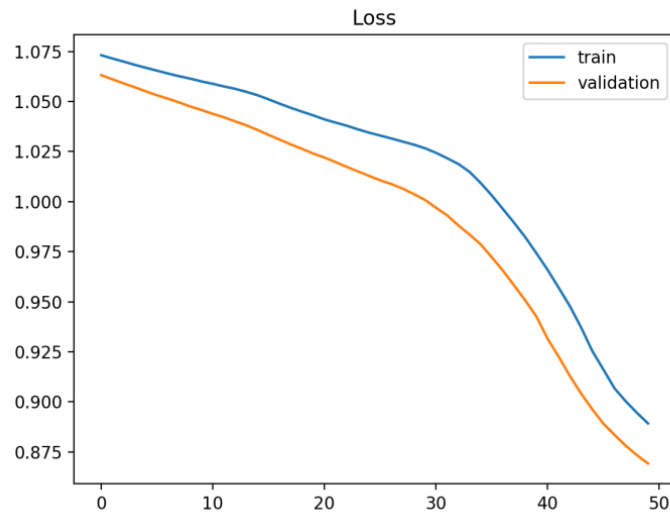
5.3.1 Točnost i gubitak

Iako unaprijeđeni model postiže nižu točnost od 41% na skupu za treniranje u usporedbi s početnim modelom (60%), na testnom skupu pokazuje nešto bolju točnost od 46%. Ovaj rezultat je očekivan s obzirom na to da je unaprijeđeni model kompleksniji i manje sklon prekomjernom prilagođavanju trening podacima. Poboljšani model ima veći početni gubitak, ali postepeno dostiže stabilniji i niži gubitak na testu (2.04 naspram 2.60 kod početnog modela). Uvođenjem *BatchNormalization* i *Dropout* slojeva u unaprijeđeni model, pokazuje bolju sposobnost generalizacije, što potvrđuje da su primijenjene tehnike povećavanja kompleksnosti modela bile uspješne.

5.3.2 Podtreniranost

Dok novi model bolje generalizira, primjećuje se visoka stopa gubitka. To ukazuje na podtreniranost. „Podtreniranost nastaje kada model nije u mogućnosti

dobiti dovoljno nisku vrijednost greške na trening skupu.“³⁸ Sljedeći graf prikazuje primjer podtreniranog modela (*Graf 4*).



*Graf 4 - Primjer krivulje učenja koja pokazuje nedovoljno obučen (podtreniran) model koji zahtijeva dodatno treniranje.*³⁹

5.3.3 Nastavak treniranja

Model bi mogao postići bolje rezultate duljim treningom, stoga će se trening nastaviti s većim brojem epoha. Nakon treniranja modela kroz 400 epoha, prikazani su rezultati posljednjih 5 epoha:

Epoch 396/400

94/94 ————— **83s** 831ms/step - accuracy: 0.8169 - loss: 0.5987 - val_accuracy: 0.7784 - val_loss: 1.1111

Epoch 397/400

94/94 ————— **93s** 925ms/step - accuracy: 0.7870 - loss: 0.6656 - val_accuracy: 0.7811 - val_loss: 1.0895

Epoch 398/400

94/94 ————— **93s** 934ms/step - accuracy: 0.8004 - loss: 0.6449 - val_accuracy: 0.7798 - val_loss: 1.0670

³⁸ Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, Genetic Programming and Evolvable Machines, DOI: 10.1007/s10710-017-9314-z, 2018., str. 111.

³⁹ Jason Brownlee, *Better Deep Learning*, Machine Learning Mastery, 2018., str. 28

Epoch 399/400

94/94 ————— **89s** 892ms/step - accuracy: 0.8192 - loss: 0.6081 - val_accuracy: 0.7798 - val_loss: 1.0952

Epoch 400/400

94/94 ————— **87s** 860ms/step - accuracy: 0.8032 - loss: 0.6482 - val_accuracy: 0.7811 - val_loss: 1.1222

Sljedeći graf prikazuje dobivene vrijednosti točnosti i gubitaka prilikom treninga i testa (*Graf 5*).



Graf 5 - Prikaz točnosti i gubitka poboljšanog modela nakon 400 epoha

Matrica konfuzije je tablični prikaz koji uspoređuje stvarne i predviđene vrijednosti modela. Idealno, brojevi na glavnoj dijagonali matrice trebali bi biti što veći, jer oni predstavljaju točno klasificirane primjere. Svi ostali brojevi u matrici predstavljaju pogreške u predviđanju, odnosno primjere koje je model pogrešno svrstao u neku drugu klasu.s

se previše prilagodi trening skupu te proširivanjem skupa podataka omogućće modelu da nauči dublje značajke. Iako su postignuti visoki rezultati, može se uočiti povećanje razlike između gubitka na treningu i testu u drugom dijelu treninga, što ukazuje na potencijalnu pretreniranost.

6. Usporedba s *transfer learning* metodama za prepoznavanja oznaka automobila

Transfer learning je tehnika strojnog učenja u kojoj se znanje stečeno kroz jedan zadatak ili skup podataka koristi za poboljšanje izvedbe modela na drugom povezanom zadatku i/ili različitom skupu podataka.⁴⁰

Drugim riječima, *transfer learning* koristi ono što je naučeno u jednoj postavci za poboljšanje generalizacije u drugom okruženju.⁴¹

6.1 ResNet152V2

U usporedbi s drugim modelima, ResNet152v2⁴² predstavlja duboku konvolucijsku neuronsku mrežu sa 152 sloja, koja se oslanja na rezidualne blokove kako bi ublažio problem nestajanja gradijenta. Rezidualni blokovi poboljšavaju treniranje dubokih mreža tako što omogućavaju mreži da preskoči neke slojeve, čime se smanjuje problem nestajućeg gradijenta i omogućuje lakše učenje složenih reprezentacija podataka.⁴³ Primjena *batch normalization-a* prije aktivacije pridonosi stabilnosti procesa treniranja te poboljšava ukupne performanse modela.⁴⁴

Model ResNet152V2 završava s potpuno povezanim slojem koji je zadužen za konačnu klasifikaciju. Ovaj sloj kombinira visokorazinske značajke izvučene iz prethodnih slojeva i pretvara ih u prostor vjerojatnosti, što omogućava modelu da donosi odluke o pripadnosti unaprijed definiranim klasama.⁴⁵ Upravo zbog te ključne uloge, zadnji sloj ResNet152V2 modela će biti korišten i prilagođavan tijekom faze treniranja.

```
for layer in base_model.layers[-1:]:
```

⁴⁰ Emilio Soria Olivas, Jose David Martin Guerrero, Marcelino Martinez Sober, Jose Rafael Magdalena-Benedito, Antonio Jose Serrano Lopez, *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, DOI: 10.4018/978-1-60566-766-9, 2010.

⁴¹ Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, Genetic Programming and Evolvable Machines, DOI: 10.1007/s10710-017-9314-z, 2018.

⁴² <https://keras.io/api/applications/resnet/#resnet152v2-function>

⁴³ Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun, *Deep Residual Learning for Image Recognition*, IEEE Conference on Computer Vision and Pattern Recognition, 2016.

⁴⁴ Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun, *Deep Residual Learning for Image Recognition*, IEEE Conference on Computer Vision and Pattern Recognition, 2016., str. 4.

⁴⁵ Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun, *Deep Residual Learning for Image Recognition*, IEEE Conference on Computer Vision and Pattern Recognition, 2016., str. 3.

```
layer.trainable = True
```

Nakon treniranja modela kroz 50 epoha sa stopom učenja od 0.0001, prikazani su rezultati posljednjih 5 epoha:

...

Epoch 46/50

94/94 ————— **234s** 2s/step - accuracy: 0.9733 - loss:
0.1037 - val_accuracy: 0.8728 - val_loss: 0.6439

Epoch 47/50

94/94 ————— **234s** 2s/step - accuracy: 0.9678 - loss:
0.1061 - val_accuracy: 0.8523 - val_loss: 0.8211

Epoch 48/50

94/94 ————— **235s** 2s/step - accuracy: 0.9665 - loss:
0.1087 - val_accuracy: 0.8482 - val_loss: 0.7274

Epoch 49/50

94/94 ————— **235s** 2s/step - accuracy: 0.9726 - loss:
0.1096 - val_accuracy: 0.8536 - val_loss: 0.7068

Epoch 50/50

94/94 ————— **234s** 2s/step - accuracy: 0.9791 - loss:
0.0775 - val_accuracy: 0.8536 - val_loss: 0.7471

Sljedeći graf prikazuje dobivene vrijednosti točnosti i gubitaka prilikom treninga i testa (*Graf 6*).



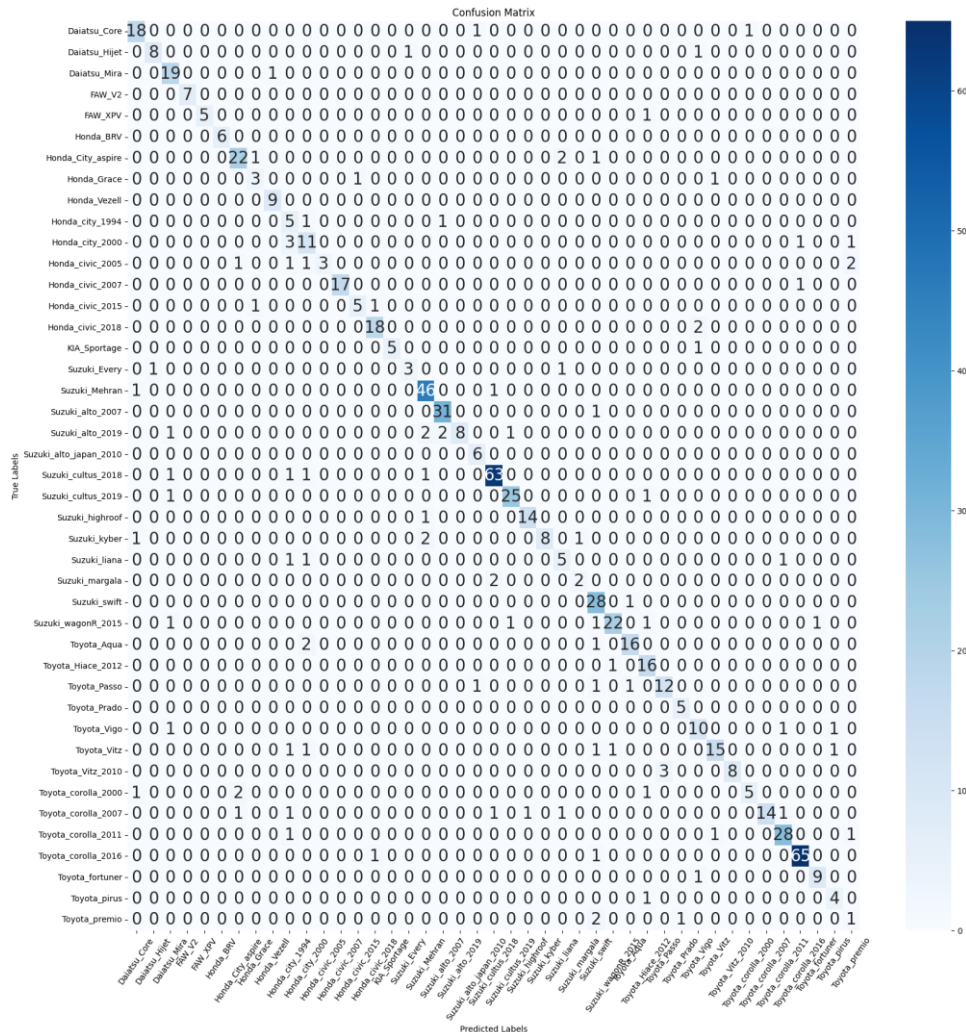
Graf 6 - Prikaz točnosti i gubitka ResNet152V2 modela nakon 50 epoha

Model ResNet152v2 u posljednoj epohi pokazao je točnost od 98% na skupu za treniranje i 85% na skupu za testiranje. Ovaj model postiže bolje rezultate od modela kreiranog od strane autora skupa podataka koji je primijenio ResNet152 model i postigao 69.24% na testom skupu. Razloge veće točnosti na našem modelu možemo pronaći u početnom uklanjanju 5 klasa sa najmanjim brojem slika prilikom pripreme podataka. Drugim riječima, pretpostavka o potencijalnom utjecaju klasa sa izrazito malim brojem slika na završne rezultate se pokazala točnom.⁴⁶

Zbog velike kompleksnosti, model je u početnim epohama brzo usvojio karakteristike automobila, što je rezultiralo visokom točnošću na skupu za treniranje. Međutim, model se počeo prekomjerno prilagođavati treniranim podacima, što se očituje u

⁴⁶ Mohsin Ali, *Vehicle images dataset for make and model recognition*, School of Computer Science, National University of Computer and Emerging Sciences in Karachi, 2022.

usporevanju brzine učenja na trening i testnom skupu i većoj razlici između točnosti na oba skupa. Matrica konfuzije navedenog modela prikazana je na slici (Slika 14).



Slika 14 - Matrica konfuzije ResNet152V2 modela nakon 50 epoha

Za usporedbu, Ložnjak⁴⁷ u svom radu koristi model ResNet152v2 za zadatak klasifikacije automobila. Postignuta točnost u Ložnjakovom istraživanju bila je 88% na testnom skupu, što je malo veća točnost od postignutih 85% u ovom istraživanju. Ova razlika od 3% može se smatrati relativno malom i može biti posljedica korištenja različitih skupova podataka, kao što je Stanford Cars Dataset⁴⁸ koji je Ložnjak koristio.

⁴⁷ Stjepan Ložnjak, Tin Kramberger, Ivan Cesar, Renata Kramberger, *Klasifikacija automobila koristeći transferirano učenje na resnet arhitekturi neuronske mreže*, 2020., str. 59.

⁴⁸ Jonathan Krause, Michael Stark, Jia Deng, Li Fei-Fei, <https://www.kaggle.com/datasets/jessicali9530/stanford-cars-dataset>, 2013.

6.2 VGG16

VGG16⁴⁹ je duboka konvolucijska neuronska mreža sa 16 slojeva koja se temelji na ponavljanju 3x3 konvolucijskih blokova.⁵⁰ Poznata je po svojoj jednostavnosti i učinkovitosti, te se često koristi u raznim zadacima računalnog vida, kao što su klasifikacija slika, detekcija objekata i semantička segmentacija.

Nakon treniranja modela kroz 50 epoha sa stopom učenja od 0.001, prikazani su rezultati posljednjih 5 epoha:

...

Epoch 46/50

94/94 ————— **45s** 467ms/step - accuracy: 0.4233 - loss: 2.0240 - val_accuracy: 0.4542 - val_loss: 1.9310

Epoch 47/50

94/94 ————— **46s** 473ms/step - accuracy: 0.4279 - loss: 2.0268 - val_accuracy: 0.4706 - val_loss: 1.8776

Epoch 48/50

94/94 ————— **45s** 469ms/step - accuracy: 0.4382 - loss: 1.9970 - val_accuracy: 0.4733 - val_loss: 1.8906

Epoch 49/50

94/94 ————— **46s** 476ms/step - accuracy: 0.4409 - loss: 1.9800 - val_accuracy: 0.4720 - val_loss: 1.8943

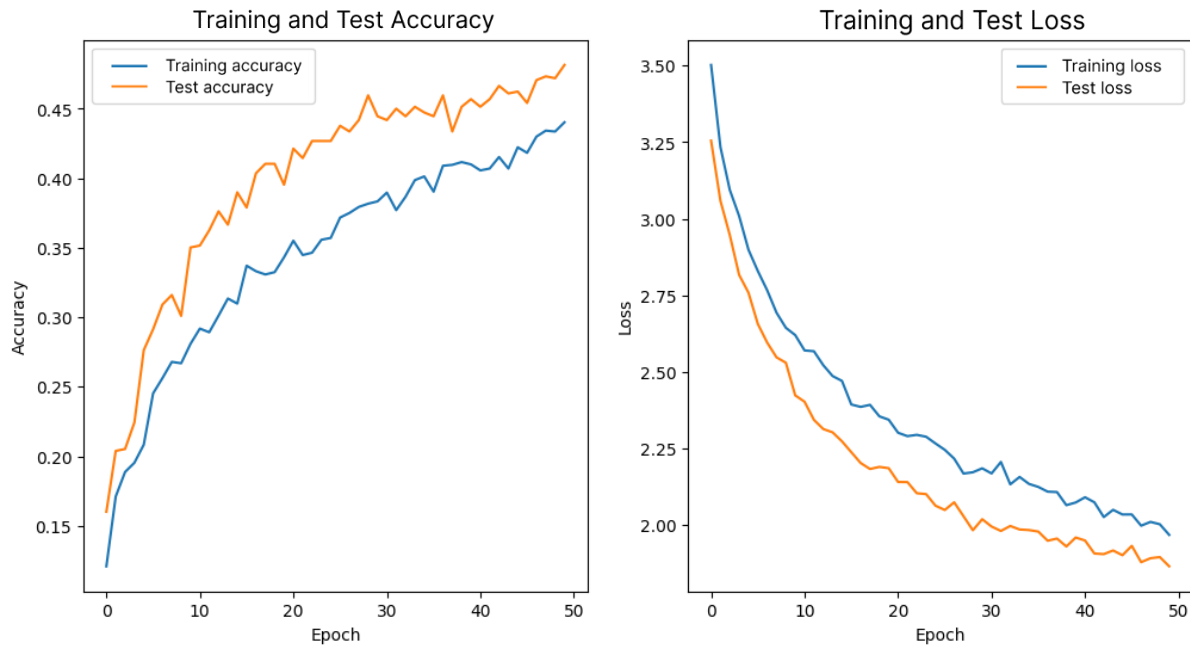
Epoch 50/50

⁴⁹ <https://keras.io/api/applications/vgg/#vgg16-function>

⁵⁰ Karen Simonyan, Andrew Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, International Conference on Learning Representations (ICLR), 2015.

94/94 ————— 47s 489ms/step - accuracy: 0.4486 - loss: 1.9303 - val_accuracy: 0.4815 - val_loss: 1.8641

Sljedeći graf prikazuje rezultate VGG16 modela nakon 50 epoha (Graf 7).



Graf 7 - Prikaz točnosti i gubitka VGG16 modela nakon 50 epoha

Ako usporedimo rezultate točnosti modela VGG16 od 45% na trening skupu i točnost od 48% na testnom skupu podataka sa rezultatima ResNet152V2 modela koji je postigao 98% na trening i 85% na testnom skupu vidimo kako se u ovom slučaju ResNet152V2 model pokazao boljim rješenjem. ResNet152V2, s većim brojem slojeva i složenijom arhitekturom, ima veći kapacitet za učenje i modeliranje složenih značajki u usporedbi s VGG16 modelom. Također, korištenje tehnika augmentacije podataka poput rotiranja i uvećavanja može povećati raznolikost podataka za treniranje i poboljšati sposobnost VGG16 modela. Matrica konfuzije navedenog modela prikazana je na slici (Slika 15).

odvojiti po dubini kako bi se smanjio trošak računanja i poboljšala učinkovitost, dok također uključuje zaostale veze za bolju stabilnost treninga.⁵²

Za potrebe ovog rada zamrznut ćemo sve slojeve Xception modela. Sljedeći kod prikazuje navedeno.

```
for layer in base_model.layers:  
    layer.trainable = False
```

Nakon 50 epoha sa stopom učenja od 0.003 postizemo sljedeće rezultate.

```
...  
Epoch 45/50  
94/94 ————— 31s 316ms/step - accuracy: 0.3787 - loss:  
2.1604 - val_accuracy: 0.4213 - val_loss: 2.0231  
Epoch 46/50  
94/94 ————— 31s 315ms/step - accuracy: 0.3791 - loss:  
2.1229 - val_accuracy: 0.4241 - val_loss: 2.0386  
Epoch 47/50  
94/94 ————— 32s 328ms/step - accuracy: 0.3660 - loss:  
2.1866 - val_accuracy: 0.4378 - val_loss: 2.0755  
Epoch 48/50  
94/94 ————— 32s 323ms/step - accuracy: 0.3706 - loss:  
2.1752 - val_accuracy: 0.4282 - val_loss: 2.0600  
Epoch 49/50  
94/94 ————— 31s 318ms/step - accuracy: 0.4070 - loss:  
2.0963 - val_accuracy: 0.4200 - val_loss: 2.0420  
Epoch 50/50  
94/94 ————— 31s 318ms/step - accuracy: 0.4064 - loss:  
2.1076 - val_accuracy: 0.4309 - val_loss: 2.0169
```

⁵² Francois Chollet, *Xception: Deep Learning with Depthwise Separable Convolutions*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017., str. 3.-4.

Sljedeći graf prikazuje rezultate Xception modela nakon 50 epoha (Graf 8).



Graf 8 - Prikaz točnosti i gubitka Xception modela nakon 50 epoha

Točnost na trening skupu porasla je s početnih 12% na 41% u posljednjoj epohi. Točnost na testnom skupu također se poboljšala, dostigavši 43% u posljednjoj epohi. Ovo ukazuje na to da model uči prepoznavati različite oznake automobila ali vidimo kako model usporava u drugom dijelu treninga.

"Stopa učenja ključni je hiperparametar koji kontrolira koliko se mijenjaju težine modela nakon svake iteracije gradijentnog spusta. Prevelika stopa učenja može dovesti do osciliranja oko optimalnog rješenja, dok premala stopa učenja može rezultirati sporom konvergencijom."⁵³

U kontekstu *transfer learning-a*, korištenje prethodno treniranog modela poput Xception-a često zahtijeva manju stopu učenja kako bi se model uspješnije prilagodio novom skupu podataka. U sljedećem primjeru, fokusirati ćemo se na fino ugađanje zadnjeg sloja i korištenje male stope učenja. Sljedećim kodom omogućavamo

⁵³ Andrew Ng, *Machine Learning Yearning*, 2018.

treniranje zadnjeg sloja Xception modela, te primjenjujemo stopu učenja koja je jednaka 0.0001.

```
for layer in base_model.layers[-1:]:  
    layer.trainable = True  
model.compile(optimizer=Adam(learning_rate=1e-4),  
loss='categorical_crossentropy', metrics=['accuracy'])
```

Nakon treniranja modela kroz 100 epoha, prikazani su rezultati posljednjih 5 epoha:

```
...  
Epoch 96/100  
188/188 ————— 158s 837ms/step - accuracy: 0.9876 -  
loss: 0.0371 - val_accuracy: 0.8960 - val_loss: 0.6383  
Epoch 97/100  
188/188 ————— 158s 838ms/step - accuracy: 0.9886 -  
loss: 0.0468 - val_accuracy: 0.8988 - val_loss: 0.5369  
Epoch 98/100  
188/188 ————— 158s 838ms/step - accuracy: 0.9888 -  
loss: 0.0375 - val_accuracy: 0.9207 - val_loss: 0.5217  
Epoch 99/100  
188/188 ————— 159s 844ms/step - accuracy: 0.9817 -  
loss: 0.0708 - val_accuracy: 0.9097 - val_loss: 0.5646  
Epoch 100/100  
188/188 ————— 158s 838ms/step - accuracy: 0.9859 -  
loss: 0.0471 - val_accuracy: 0.9124 - val_loss: 0.5906
```

Nakon treniranja dodatnim brojem epoha postićemo točnost od 99% na trening skupu i točnost od 91% na testnom skupu podataka što je znatno bolji rezultat od prethodnog, kada smo koristili stopu učenja jednaka 0.001. Sljedeći graf prikazuje rezultate Xception modela nakon 100 epoha i stopom učenja jednaka 0.0001 (*Graf 9*).



Graf 9 - Prikaz točnosti i gubitka Xception modela nakon optimiziranja

sporiju stopu učenja (0.0001), te koristeći zadnji sloj modela za treniranje. Ova metoda postigla je najveću točnost od 99% na trening skupu, te 91% na testnom skupu podataka. Vjerujemo da je razlog za to što je arhitektura Xception modela dobro prilagođena za zadatke prepoznavanja slika, a veliki i raznolik skup podataka pod nazivima ImageNet i JFT na kojem je model treniran omogućio je izvlačenje robusnih značajki, a zatim se model prilagodio našem problemu tako što je korišten zadnji sloj modela za treniranje.

7. Utjecaj parametara na performanse

Analiziran je utjecaj različitih hiperparametara na performanse modela kako bi se identificirali optimalni uvjeti za treniranje. Parametri kao što su veličina serije, stopa učenja i broj epoha značajno su utjecali na točnost i brzinu konvergencije modela. Rezultati pokazuju da je pažljivo podešavanje hiperparametara ključno za postizanje visokih performansi.

7.1 Utjecaj stope učenja na točnost

Kao što je već istaknuto u ranijim dijelovima rada, stopa učenja je ključan hiperparametar u treniranju neuronskih mreža. Odabir prikladne stope učenja izravno utječe na brzinu i stabilnost procesa učenja, a time i na konačnu točnost modela. U sljedećoj tablici prikazani su utjecaji različitih stopa učenja na završne rezultate treninga i testa, bez mijenjanja drugih hiperparametara. Za potrebe treninga korišten je početni model sa 7 slojeva, koji se sadrži od 2 konvolucijska i 2 *max pooling* sloja.

Stopa učenja	Broj epoha	Konačna točnost na trening skupu	Konačna točnost na testnom skupu
0.01	100	8%	9%
0.001	100	60%	50%
0.0001	100	30%	27%
0.00001	100	14%	18%

Tablica 1 - Usporedba različite stope učenja na istom skupu podataka

- **Prevelika stopa učenja (0.01):** Model konvergira prebrzo i može se zaglaviti u lokalnim minimumima, što rezultira niskom točnošću na skupu podataka.
- **Premala stopa učenja (0.00001):** Model konvergira sporo, što zahtijeva veći broj epoha za postizanje boljih rezultata. Međutim, čak i nakon 100 epoha, točnost je još uvijek relativno niska.

- **Optimalna stopa učenja (oko 0.001):** Model postiže najbolju točnost na trening skupu, ali dolazi do prenaučivosti, što se očituje u značajnom padu točnosti na testnom skupu.

Stopa učenja određuje koliko će se težine u modelu ažurirati u svakoj iteraciji treniranja. Prevelika stopa učenja može uzrokovati da model "preskoči" optimalne vrijednosti težina, dok premala stopa učenja može dovesti do sporog konvergiranja ili čak zaglavlivanja u lokalnim minimumima. U radu je dokazano kako je dobro koristiti različite stope učenja u različitim stadijima treninga.

8. Zaključak

Ovaj rad predstavlja sustav za prepoznavanje oznaka automobila temeljen na neuronskim mrežama. Korišteni su razni modeli sa raznovrsnom kombinacijom hiperparametara koji su se uspoređivali. Koristeći Xception model i trenirajući sa zadnjim, potpuno povezanim slojem Xception modela, sustav je uspješno prepoznao oznake automobila, postižući točnost od 99% na skupu za treniranje i 91% na skupu za testiranje. Buduća istraživanja trebala bi se fokusirati na daljnje poboljšavanje performansi modela korištenjem naprednih tehnika dubokog učenja, kao i na povećavanje otpornosti sustava na različite uvjete snimanja. Dodatno, istraživanje bi moglo uključivati primjenu sustava u stvarnom vremenu za potrebe sigurnosnih sustava i pametnih prometnih rješenja.

9. Literatura

1. Linjie Yang, Ping Luo, Chen Change Loy, Xiaoou Tang, *A Large-Scale Car Dataset for Fine-Grained Categorization and Verification*, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
2. Donny Avianto, Agus Harjoko, Afiahayati, *CNN-Based Classification for Highly Similar Vehicle Model Using Multi-Task Learning*, Journal of Imaging, DOI: 10.3390/jimaging8110293, 2022.
3. Nanning Zheng , George Loizou , Xiaoyi Jiang , Xuguang Lan, Xuelong Li, *Computer vision and pattern recognition*, International Journal of Computer Mathematics, DOI: 10.1080/00207160701303912, 2007.
4. Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, Genetic Programming and Evolvable Machines, DOI: 10.1007/s10710-017-9314-z, 2018.
5. Jan Schafer, Philipp Schmitt, Mark W. Hlawitschka, and Hans-Jorg Bart, *Measuring Particle Size Distributions in Multiphase Flows Using a Convolutional Neural Network*, Chemie Ingenieur Technik, DOI: 10.1002/cite.201900099, 2019.
6. Jason Brownlee, *Better Deep Learning*, Machine Learning Mastery, 2018.
7. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, DOI: 10.1145/3065386, Communications of the ACM, 2017.
8. Andrew Ng, *Machine Learning Yearning*, https://nessie.ilab.sztaki.hu/~kornai/2020/AdvancedMachineLearning/Ng_MachineLearningYearning.pdf, 2018.

9. Emilio Soria Olivas, Jose David Martin Guerrero, Marcelino Martinez Sober, Jose Rafael Magdalena Benedito, Antonio Jose Serrano Lopez, *Handbook of Research on Machine Learning Applications and Trends*, DOI: 10.4018/978-1-60566-766-9, 2010.
10. Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun, *Deep Residual Learning for Image Recognition*, IEEE Conference on Computer Vision and Pattern Recognition, 2016.
11. Stjepan Ložnjak, Tin Kramberger, Ivan Cesar, Renata Kramberger, *Klasifikacija automobila koristeći transferirano učenje na resnet arhitekturi neuronske mreže*, 2020.
12. Karen Simonyan, Andrew Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, International Conference on Learning Representations (ICLR), 2015.
13. Francois Chollet, *Xception: Deep Learning with Depthwise Separable Convolutions*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
14. Ivan Marasović, Goran Majič, Ivan Škalić, Željka Tomasović, *Indoor Localization of Industrial IoT Devices and Applications Based on Recurrent Neural Networks*, Journal of Communications Software and Systems, 2024.
15. Mohsin Ali, *Vehicle images dataset for make and model recognition*, School of Computer Science, National University of Computer and Emerging Sciences in Karachi, 2022
16. Andrew Ng, *Convolutional Neural Networks*, UFLDL Tutorial, Stanford University, <http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/>, 2013.

17. Sumit Saha, *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*, Towards Data Science, <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
18. Jay Wang, Robert Turko, Omar Shaikh, Haekyu Park, Nilaksh Das, Fred Hohman, Minsuk Kahng, Polo Chau, *CNN Explainer: Learning Convolutional Neural Networks with Interactive Visualization*, <https://poloclub.github.io/cnn-explainer/>, 2020.
19. Shruti Jadon, *Introduction to Different Activation Functions for Deep Learning*, Medium, <https://medium.com/@shrutijadon/survey-on-activation-functions-for-deep-learning-9689331ba092>, 2018.
20. Python Software Foundation, Python, <https://www.python.org/>, 1991.
21. Google, Tensorflow, <https://www.tensorflow.org/>, 2015.
22. François Chollet, Keras, <https://keras.io/>, 2015.
23. Ian Bicking, Pip, <https://pypi.org/project/pip/>, 2014.
24. John Hunter, Matplotlib, <https://matplotlib.org/>, 2003.
25. Jonathan Krause, Michael Stark, Jia Deng, Li Fei-Fei, Stanford Cars Dataset, <https://www.kaggle.com/datasets/jessicali9530/stanford-cars-dataset>, 2013.

10. Prilozi

10.1 Popis slika

Slika 1 - Primjer automobila Suzuki Swift iz trening skupa podataka	5
Slika 2 - Primjer automobila Honda Civic iz testnog skupa podataka.....	6
Slika 3 - Djelomično prekriveni automobili.....	6
Slika 4 - Automobili u različitim osvjetljenjima	7
Slika 5 - Arhitektura konvolucijske neuronske mreže. Sastoji se od sljedećih slojeva: ulazni sloj, konvolucijski slojevi, pooling slojeva, potpuno povezani sloj, te izlazni sloj	9
Slika 6 - proces konvolucije u neuronskim mrežama, gdje se ulazna slika razlaže na crveni, zeleni i plavi kanal, a zatim prolazi kroz konvolucijske filtere koji generiraju mape značajki.	10
Slika 7 - Različite aktivacijske funkcije i njihovi grafikoni	11
Slika 8 - Primjer operacije konvolucije i max pooling-a.....	11
Slika 9 - Vizualni 2D prikaz početnog modela	13
Slika 10 - Primjer slika sa različitim rasponom uvećavanja	21
Slika 11 - Primjer slike sa različitim rasponom svjetline	21
Slika 12 - Vizualni 2D prikaz unaprijeđenog modela	24
Slika 13 - Matrica konfuzije poboljšanog modela nakon 400 epoha.....	30
Slika 14 - Matrica konfuzije ResNet152V2 modela nakon 50 epoha.....	35
Slika 15 - Matrica konfuzije VGG16 modela nakon 50 epoha	38
Slika 16 - Matrica konfuzije Xception modela nakon optimiziranja	43

10.2 Popis tablica

Tablica 1 - Usporedba različite stope učenja na istom skupu podataka	45
--	----

10.3 Popis grafova

Graf 1 - Prikaz točnosti i gubitka nakon 100 epoha.....	19
Graf 2 - Primjer krivulja učenja za trening i validaciju koje pokazuju pretreniran model	20
Graf 3 - Prikaz točnosti i gubitka poboljšanog modela nakon 100 epoha	27

Graf 4 - Primjer krivulje učenja koja pokazuje nedovoljno obučen (podtreniran) model koji zahtijeva dodatno treniranje.	28
Graf 5 - Prikaz točnosti i gubitka poboljšanog modela nakon 400 epoha	29
Graf 6 - Prikaz točnosti i gubitka ResNet152V2 modela nakon 50 epoha	34
Graf 7 - Prikaz točnosti i gubitka VGG16 modela nakon 50 epoha	37
Graf 8 - Prikaz točnosti i gubitka Xception modela nakon 50 epoha	40
Graf 9 - Prikaz točnosti i gubitka Xception modela nakon optimiziranja	42