### SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU ELEKTROTEHNIČKI FAKULTET OSIJEK

### Domin Radić

CAR IMAGES DATASET

IZVJEŠTAJ KONSTRUKCIJSKIH VJEŽBI IZ KOLEGIJA STROJNO UČENJE

### Osijek, 2024 godina.

## SADRŽAJ

1. [UVOD I OPIS PROBLEMA 1](#_bookmark0)
2. [ODABIR ARHITEKTURE MREŽE 2](#_bookmark1)
3. [PROCES TRENIRANJA MREŽE 3](#_bookmark2)
4. [EVULACIJA MREŽE 4](#_bookmark3)

# UVOD I OPIS PROBLEMA

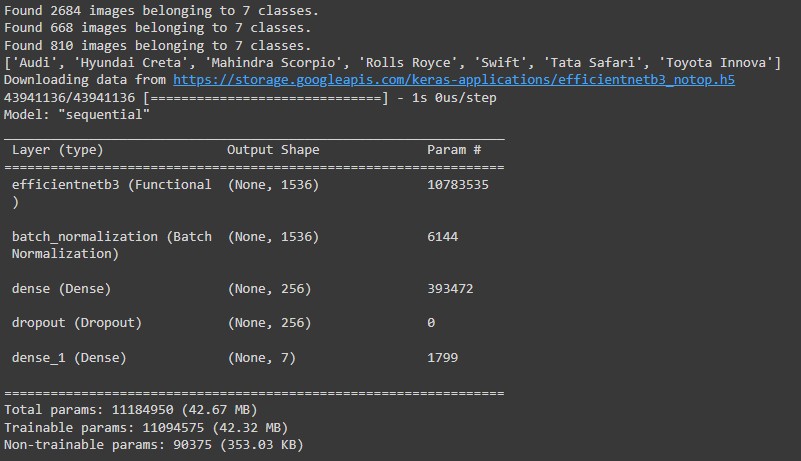
Potrebno je napraviti i istrenirati neuronsku mrežu za dani skup podataka "Car Images Dataset" kako bi se vršila klasifikacija različitih modela automobila. Skup podataka sastoji se od raznih vrsta automobila, organiziran je u 2 mape (trening, test) i sadrži podmape za svaku kategoriju automobila. Postoji 4165 slika (JPG) i 7 klasa automobila. Također, potrebno je pripremiti skup podataka, odvojiti ga na skupove za treniranje, validaciju i testiranje. Napraviti samostalno novi model ili odraditi prijenosno učenje na postojećem modelu. Ugraditi TensorBoard radi lakšeg praćenja i dokumentiranja rezultata, koristiti tehnike optimizacije treniranja kako bi se postigli što bolji rezultati (augmentacija podatkovnog skupa, primjena optimizatora, regularizacija, dropout, primjena metoda koje koriste pozivne funkcije kao što su EarlyStopping i Learning Rate Scheduling). Testirati model na testnom skupu i opisati rezultate.

# ODABIR ARHITEKTURE MREŽE

Tokom izgradnje neuronske mreže korišteno je više modela, od početne mreže koja je korištena na laboratorijskim vježbama, do postojećih modela kao što su VGG16, EfficientNetB0 i EfficientNetB3. Za ovaj projekt korišten je prijenosni model EfficientNetB3 kao bazni model, kojem su dodani dodatni slojevi za prilagodbu specifičnom zadatku klasifikacije automobila. EfficientNetB3 je odabran zbog svoje učinkovitosti, visoke točnosti u različitim zadacima klasifikacije slika i jer je od svih navedenih korištenih modela dao najveću preciznost na validacijskom skupu.

Arhitektura mreže:

* 1. **Bazni model -** EfficientNetB3 bez završnog sloja.
  2. **BatchNormalization -** Za normalizaciju aktivacija slojeva i ubrzanje procesa treniranja mreže.
  3. **Dense sloj -** 256 neurona s L2 i L1 regularizacijama, te ReLU aktivacijskom funkcijom.
  4. **Dropout -** S stopom ispuštanja od 45% za sprečavanje overfittinga.
  5. **Izlazni sloj -** Klasifikacijski sloj s brojem neurona jednakim broju klasa u skupu podataka i softmax aktivacijskom funkcijom.



*Slika 2.1 Parametri najbolje mreže*

# PROCES TRENIRANJA MREŽE

Proces treniranja mreže započinje pripremom skupa podataka, gdje se slike automobila augmentiraju kako bi se povećala raznolikost i spriječio overfitting. Zatim se definira arhitektura modela, koristeći EfficientNetB3 kao bazni model, kojem se dodaju dodatni slojevi za klasifikaciju. Model se zatim kompajlira s Adam optimizatorom, sa početnim learning rate-om od 0.001. Treniranje se provodi korištenjem trenirajućeg skupa podataka, uz validacijski skup za praćenje performansi. Za poboljšanje procesa treniranja, postavljene su callback funkcije: ModelCheckpoint za spremanje najboljeg modela, EarlyStopping za rano zaustavljanje treniranja ako se validacijska točnost ne poboljša, ReduceLROnPlateau za smanjenje stope učenja ako se validacijski gubitak ne smanji, te TensorBoard za praćenje treniranja.

Model je treniran na trenirajućem skupu podataka uz validaciju na validacijskom skupu tijekom 25 epoha. Nakon treniranja, model je evaluiran na testnom skupu, pri čemu su rezultati prikazani kroz grafove preciznosti i gubitka. Izrađena je i konfuzijska matrica za validacijski skup i testni skup te je prikazan izvještaj klasifikacije.

# EVULACIJA MREŽE

Na temelju rezultata evaluacije mreže, može se zaključiti da model pokazuje izvrsne performanse u klasifikaciji automobila na temelju slika. Tijekom obuke, model je postigao visoku točnost na trening i validacijskim podacima, što je prikazano grafikonima točnosti (accuracy) i gubitka (loss). Model je evaluiran na testnom skupu, gdje je postigao accuracy od 98.40% i loss od 0.7767.

Epoch 1: val\_accuracy improved from -inf to 0.89970, saving model to best\_model.h5

/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file format is considered legacy. We recommend using instead the native Keras format, e.g.

`model.save('my\_model.keras')`. saving\_api.save\_model(

168/168 [==============================] - 157s 576ms/step - loss: 7.9630 - accuracy: 0.6747 - val\_loss: 6.6145 -

val\_accuracy: 0.8997 - lr: 0.0010 Epoch 2/25

168/168 [==============================] - ETA: 0s - loss: 5.7046 - accuracy: 0.9050

Epoch 2: val\_accuracy improved from 0.89970 to 0.95210, saving model to best\_model.h5

168/168 [==============================] - 92s 546ms/step - loss: 5.7046 - accuracy: 0.9050 - val\_loss: 4.8887 -

val\_accuracy: 0.9521 - lr: 0.0010 Epoch 3/25

168/168 [==============================] - ETA: 0s - loss: 4.2758 - accuracy: 0.9411

Epoch 3: val\_accuracy improved from 0.95210 to 0.97006, saving model to best\_model.h5

168/168 [==============================] - 90s 535ms/step - loss: 4.2758 - accuracy: 0.9411 - val\_loss: 3.6676 -

val\_accuracy: 0.9701 - lr: 0.0010 Epoch 4/25

168/168 [==============================] - ETA: 0s - loss: 3.1915 - accuracy: 0.9680

Epoch 4: val\_accuracy did not improve from 0.97006

168/168 [==============================] - 88s 525ms/step - loss: 3.1915 - accuracy: 0.9680 - val\_loss: 2.6974 -

val\_accuracy: 0.9701 - lr: 0.0010 Epoch 5/25

168/168 [==============================] - ETA: 0s - loss: 2.3537 - accuracy: 0.9803

Epoch 5: val\_accuracy improved from 0.97006 to 0.97605, saving model to best\_model.h5

168/168 [==============================] - 89s 530ms/step - loss: 2.3537 - accuracy: 0.9803 - val\_loss: 1.9893 -

val\_accuracy: 0.9760 - lr: 0.0010 Epoch 6/25

168/168 [==============================] - ETA: 0s - loss: 1.7277 - accuracy: 0.9858

Epoch 6: val\_accuracy did not improve from 0.97605

168/168 [==============================] - 96s 571ms/step - loss: 1.7277 - accuracy: 0.9858 - val\_loss: 1.4547 -

val\_accuracy: 0.9746 - lr: 0.0010 Epoch 7/25

168/168 [==============================] - ETA: 0s - loss: 1.2609 - accuracy: 0.9870

Epoch 7: val\_accuracy did not improve from 0.97605

168/168 [==============================] - 88s 524ms/step - loss: 1.2609 - accuracy: 0.9870 - val\_loss: 1.0610 -

val\_accuracy: 0.9731 - lr: 0.0010 Epoch 8/25

168/168 [==============================] - ETA: 0s - loss: 0.9375 - accuracy: 0.9862

Epoch 8: val\_accuracy improved from 0.97605 to 0.98353, saving model to best\_model.h5

168/168 [==============================] - 91s 540ms/step - loss: 0.9375 - accuracy: 0.9862 - val\_loss: 0.7833 -

val\_accuracy: 0.9835 - lr: 0.0010 Epoch 9/25

168/168 [==============================] - ETA: 0s - loss: 0.7201 - accuracy: 0.9870

Epoch 9: val\_accuracy did not improve from 0.98353

168/168 [==============================] - 88s 524ms/step - loss: 0.7201 - accuracy: 0.9870 - val\_loss: 0.6185 -

val\_accuracy: 0.9775 - lr: 0.0010 Epoch 10/25

168/168 [==============================] - ETA: 0s - loss: 0.5651 - accuracy: 0.9877

Epoch 10: val\_accuracy did not improve from 0.98353

168/168 [==============================] - 89s 527ms/step - loss: 0.5651 - accuracy: 0.9877 - val\_loss: 0.5121 -

val\_accuracy: 0.9760 - lr: 0.0010 Epoch 11/25

168/168 [==============================] - ETA: 0s - loss: 0.4507 - accuracy: 0.9929

Epoch 11: val\_accuracy improved from 0.98353 to 0.98802, saving model to best\_model.h5

168/168 [==============================] - 92s 546ms/step - loss: 0.4507 - accuracy: 0.9929 - val\_loss: 0.4022 -

val\_accuracy: 0.9880 - lr: 0.0010 Epoch 12/25

168/168 [==============================] - ETA: 0s - loss: 0.3963 - accuracy: 0.9892

Epoch 12: val\_accuracy did not improve from 0.98802

168/168 [==============================] - 95s 567ms/step - loss: 0.3963 - accuracy: 0.9892 - val\_loss: 0.3470 -

val\_accuracy: 0.9820 - lr: 0.0010 Epoch 13/25

168/168 [==============================] - ETA: 0s - loss: 0.3230 - accuracy: 0.9944

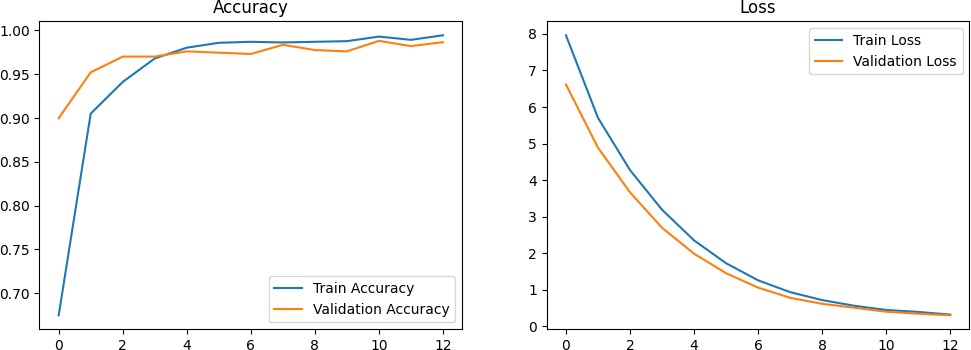
Epoch 13: val\_accuracy did not improve from 0.98802 Restoring model weights from the end of the best epoch: 8.

168/168 [==============================] - 100s 593ms/step - loss: 0.3230 - accuracy: 0.9944 - val\_loss: 0.3047 -

val\_accuracy: 0.9865 - lr: 0.0010 Epoch 13: early stopping

51/51 [==============================] - 6s 124ms/step - loss: 0.7767 - accuracy: 0.9840 Test Loss: 0.7766649127006531, Test Accuracy: 0.9839506149291992

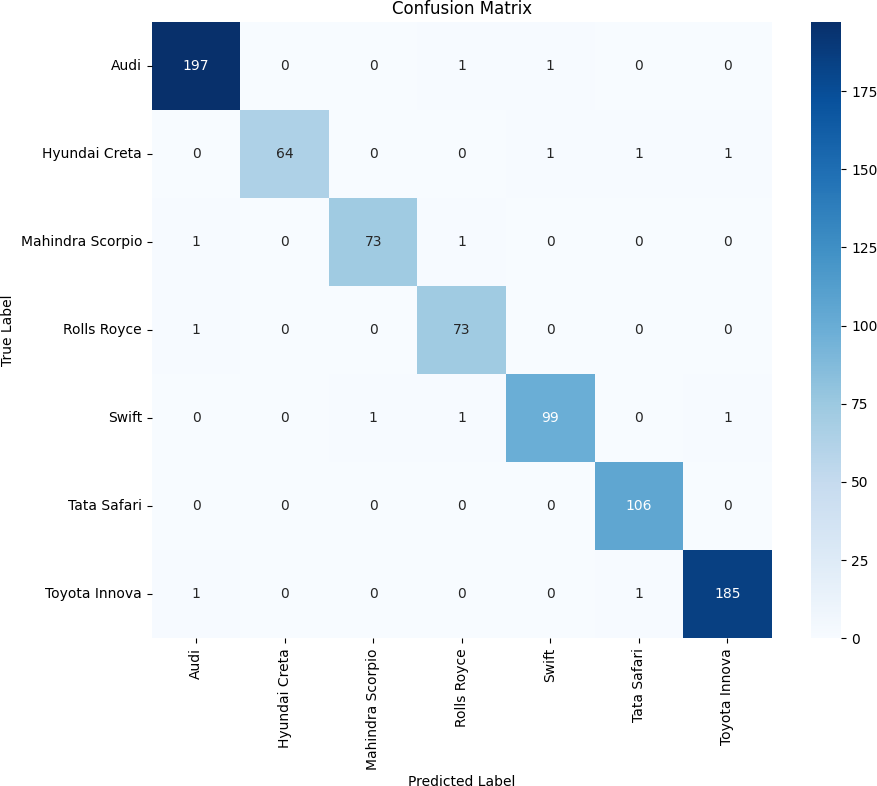
Model je zaustavljen nakon 13 epoha zbog EarlyStopping, gdje je zadržao najbolju točnost na validacijskom skupu postignutu do osme epohe. Sveukupno, model pokazuje visoku točnost i nisku stopu grešaka kroz cijeli proces obuke.



*Slika 4.1. Prikaz grafikona točnosti i gubitaka na validacijskom i tesntom skupu podataka*

Na grafikonima se vidi da točnost na trening i validacijskom skupu raste kroz epohe i postiže vrlo visoke vrijednosti. To znači da je model naučio prepoznati uzorke i dobro generalizira na validacijskim podacima. Gubitak na trening i validacijskom skupu pokazuje pad kroz epohe, što sugerira da model minimizira greške u predikcijama. Stabilizacija gubitka na validacijskom skupu nakon određene točke također sugerira da model ne nastavlja s overfittingom nakon što dosegne optimalnu točnost.

Grafovi pokazuju da model uči učinkovito i da se generalizira dobro na neviđene podatke, dok izbjegava značajan overfitting.

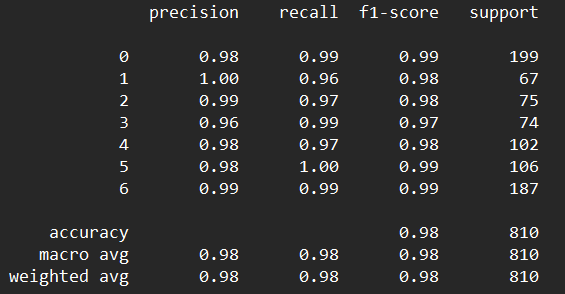


*Slika 4.2 Konfuzijska matrica validacijskog skupa podataka*



*Slika 4.3 Konfuzijska matrica testnog skupa podataka*

Konfuzijske matrice pokazuju da model ima visoku točnost i nisku stopu pogrešnih klasifikacija na oba skupa podataka (validacijski i testni). To znači da je model vrlo učinkovit u prepoznavanju i klasificiranju slika automobila za različite klase, a niska stopa pogrešnih klasifikacija upućuje na dobru generalizaciju modela.



Slika 4.4 Izvještaj o klasifikaciji za svaku klasu

Iz izvještaja o klasifikaciji može se vidjeti da model ima izuzetno visoke performanse u klasifikaciji sedam različitih klasa automobila. Točnost modela iznosi 98%, što znači da model pravilno klasificira 98% svih primjera u testnom skupu. Visoke vrijednosti preciznosti, odziva i F1-score za sve klase ukazuju na to da model rijetko čini pogreške u predviđanju te uspješno prepoznaje većinu stvarnih pozitivnih primjera.

Model je vrlo učinkovit i pouzdan za zadatak klasifikacije automobila, što je potvrđeno visokim vrijednostima evaluacijskih metrika i vizualizacijom rezultata.