Computer Vision Exam

Image Classification

## **boot or sandal?**

**Ognjen Radaković 2019230431**

A picture containing clothing, black, shoes, feet

Description automatically generated

# Brief introduction with problem description

This project is about a binary image classification model that is using Convolutional Neural Network (CNN) in Tensorflor (Keras), where our model is used to classify/recognize the images of either boot or sandal. First, we loaded and imported our dataset using google drive.

The model architecture is defined using the Sequential model and adding several Conv2D, MaxPooling2D, Flatten, and Dense layers.

Then the model is compiled with the Adam optimizer, loss function 'binary\_crossentropy', and accuracy metric.

Then of course the model is being trained on the training dataset for 10 epochs and validated on test dataset, after which the predictions on the test dataset were made.

The performance of the model is evaluated using the evaluate method and is reported using the ‘classification\_report’ method. Afterwards we were using VGG19 model.

That VGG19 model is pre-trained on a large dataset and can be used to extract features from new image data to train a new classifier on top.

Also, the pre-trained VGG19 model is fine-tuned by unfreezing some of the layers and training the model for additional epochs.

The accuracy and loss for both the training and validation sets are plotted over the course of the fine-tuning process. The final test accuracy of the model is also reported.

The dataset that we have been using has 804 training pictures and 96 testing pictures. There could be a high possibility of wrong guessing whether things shown on those images can be recognized as boots or sandals, with the fact that there are some similar and to the human eye confusing models. In this project, we are going to try to improve/make a model which will have the best guessing and of course, to minimize the errors.

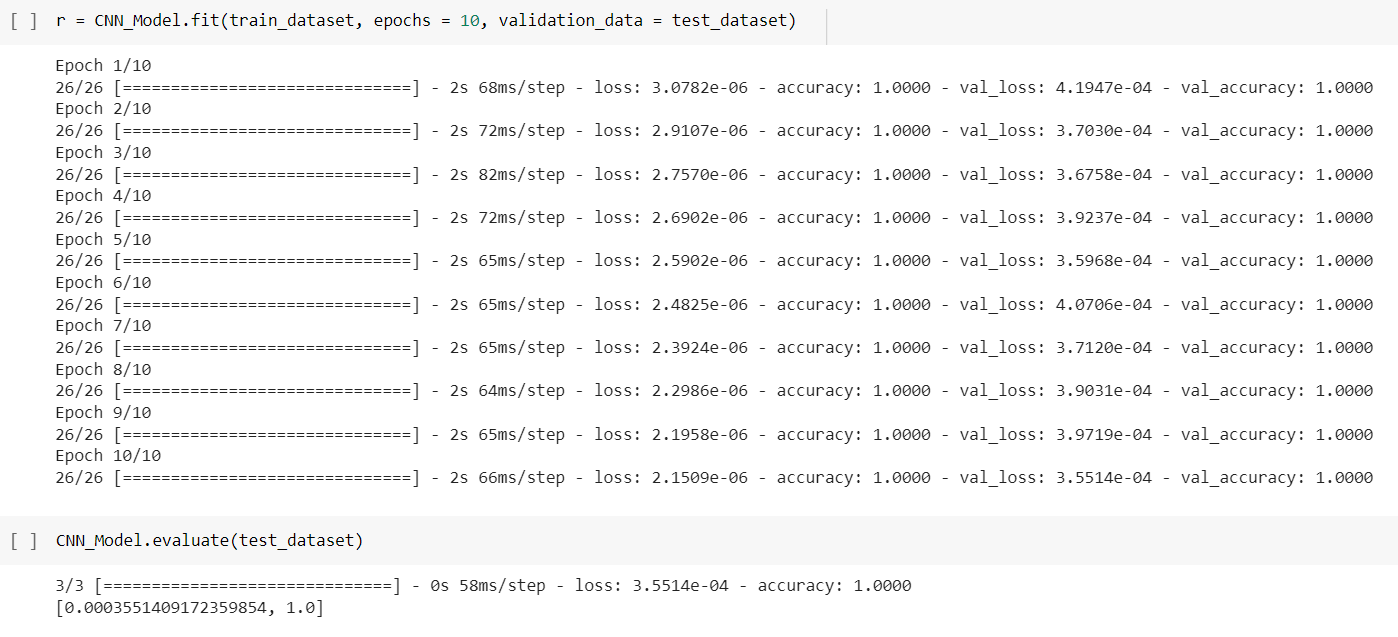
Diagram, engineering drawing

Description automatically generated

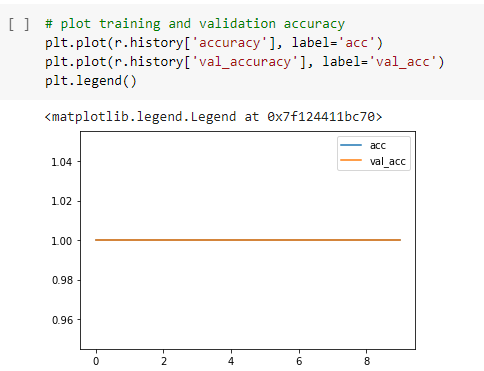
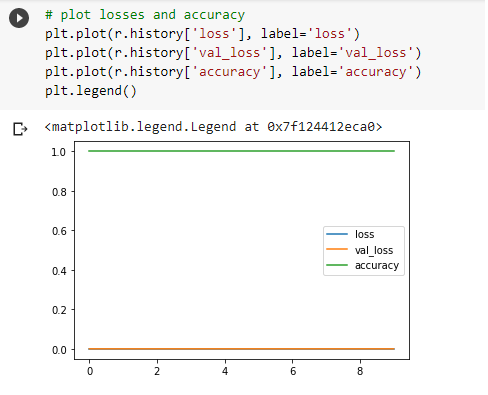
# Solution

In here the problem solution will be shown, by going through the more important parts of code and with some of the code snippets added.

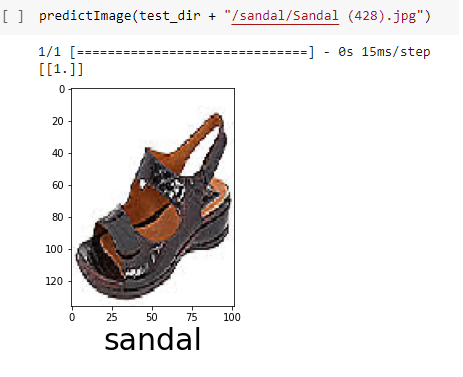
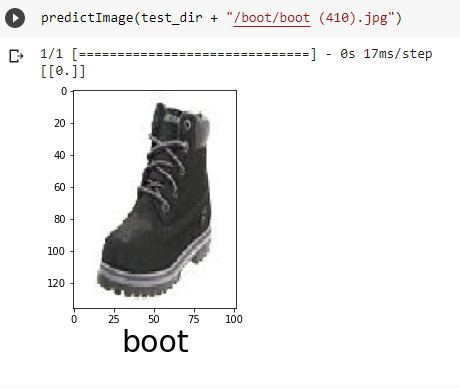
On the beginning here in the code below – the output shows the progress of the training process of each epoch. It reports the loss, accuracy, validation loss, and validation accuracy for each epoch. The model accuracy is 1.0, which indicates that it is making perfect predictions on the training data. The validation accuracy is also close to 1.0, indicating that the model is not overfitting to the training data. The decrease in the loss and validation loss over the 10 epochs suggests that the model is learning and improving with each iteration.



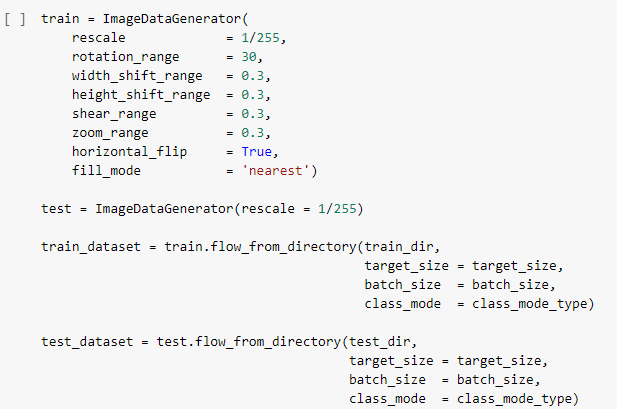
All of it is also shown here on the graphs below.



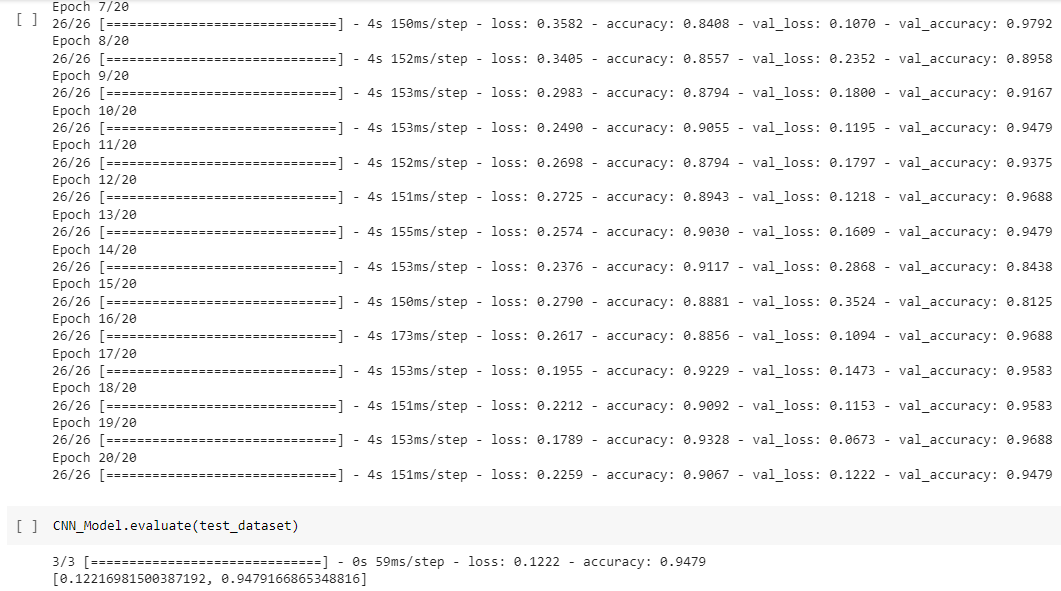
Down here the predictions are being shown. The images from the test direction have been used to prove that the model works perfect.



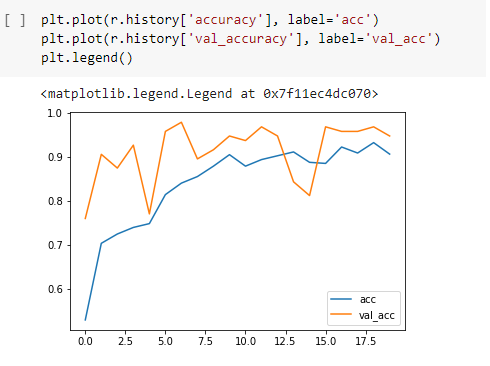
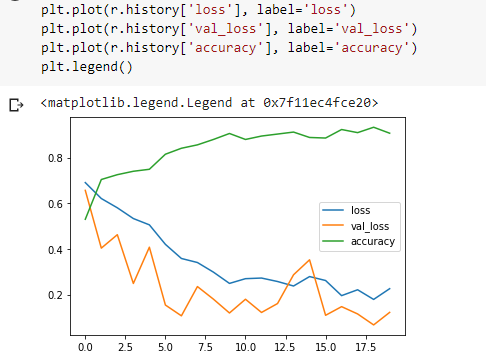
As shown below, here is the part of code for Data augmentation, which is a technique to artificially increase the size of a training dataset by creating modified version of existing images. This can help prevent overfitting, which is when a model performs (too) well on the training data but poorly on new, unseen data.



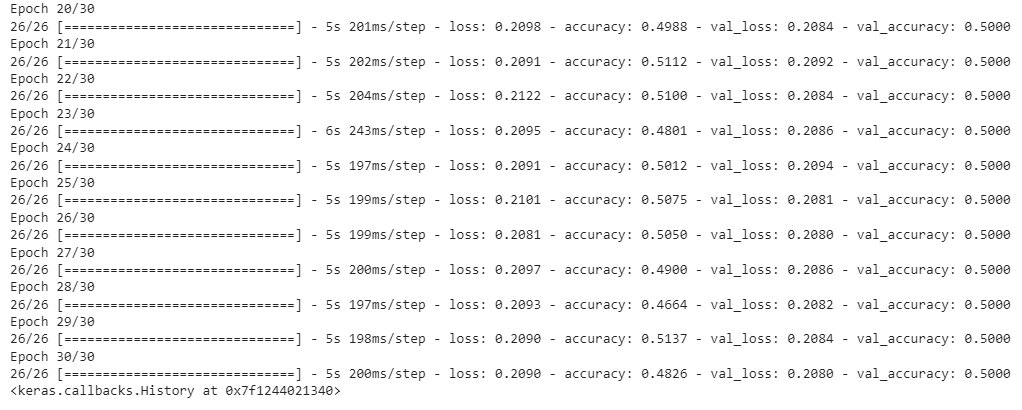
Here are the new results after implementing data augmentation:



And graphs as well:



Later in the code VGG19 model is being implemented, which is a type of convolutional neural network (CNN) that is used for image classification tasks. The network consists of multiple convolutional layers, each followed by a ReLU activation function, and max pooling layers. The output of the final pooling layer is then flattened and passed through multiple dense layers with ReLU activation functions and dropout regularization. The model is then compiled using the Adam optimizer with a learning rate of 0.003, a categorical cross-entropy loss function, and accuracy as a metric. Down here are the results from the last 11 epochs (30 in total):



These were some of the most important code snippets and they’ve been shown to clarify what project is about.

# Summary

In this project a few image classification methods have been implemented. The network has been trained, loss function checked, including the training and validation results. Looking at the results and comparing them, it is clear that the results have been the best in the beginning and that our first model gave actually the perfect predictions with accuracy value of 1.0, error value of 0.00035 and validation accuracy value of 1.0. After implementing the Data augmentation, the results got just a bit worse but good enough, and after implementing VGG 19 model all the resulting values got just much worse, as it has been seen in the code snippets above. Even after implementing fine tuning there was not a lot of change in those results.