BIRZEIT UNIVERSITY

Faculty of Engineering

Department of Electrical & Computer Engineering

ENCS5343 Computer Vision

Handwritten Arabic Letters Classification using CNN

Prepared By: Abdelrahman Hamza - 1200020

Instructor Dr. Aziz Qaroush

Section: 2

Date: Jan 29, 2024

Introduction

Classification of Arabic Handwriting is a complex and important topic, this project will focus on developing a CNN Computer Vision model that classifies an input image of a dis-connected arabic letter (1/28). The model will be trained on 13,340 input images in 4 different techniques.

1. CNN-14: This is a network based on the following research paper https://www.researchgate.net/publication/372192699_Arabic_Handwritten_Character_Recognition_Using_Convolutional_Neural_Networks This architecture was selected due its high accuracy as well as non-complex structure and low number of trainable parameters compared to other networks.

2. CNN-14 with Data Augmentation

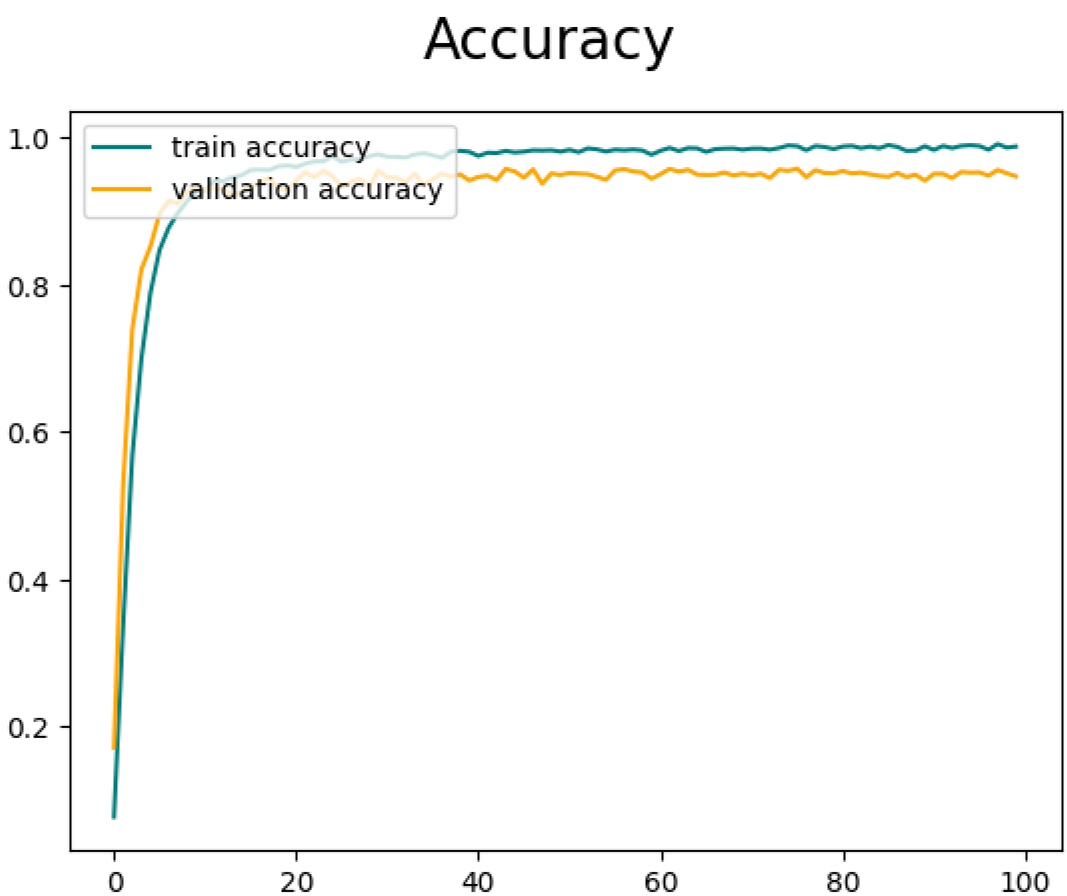
3. DenseNet121: I choose to try a predefined model and train it on our dataset. I chose DenseNet121 because it doesn't require a lot of operations while still managing a decent accuracy rate.

4. Transfer Learning: I found an online model that was built and trained to Classify English Digits (0-9). I fine-tuned the entire model since my dataset is large enough to do so.

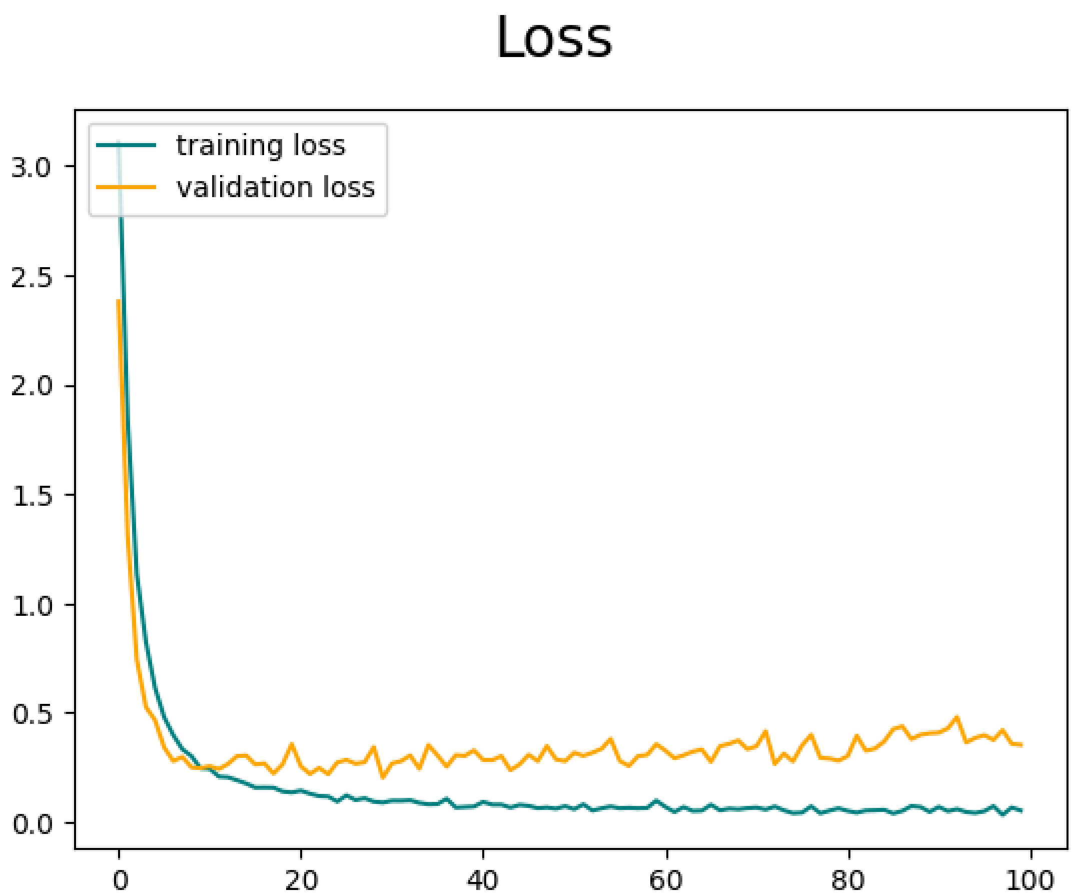
The dataset provided is split into 2 parts, training & testing. The model will be trained only on the training data and will be evaluated based on the testing data. Performance metrics like Accuracy & Loss will be plotted vs Epoch for each model.

The code is run on google-colab.

CNN-14


Accuracy
train accuracy
validation accuracy
1.0
0.8
0.6
0.4
0.2
0
20
40
60
80
100

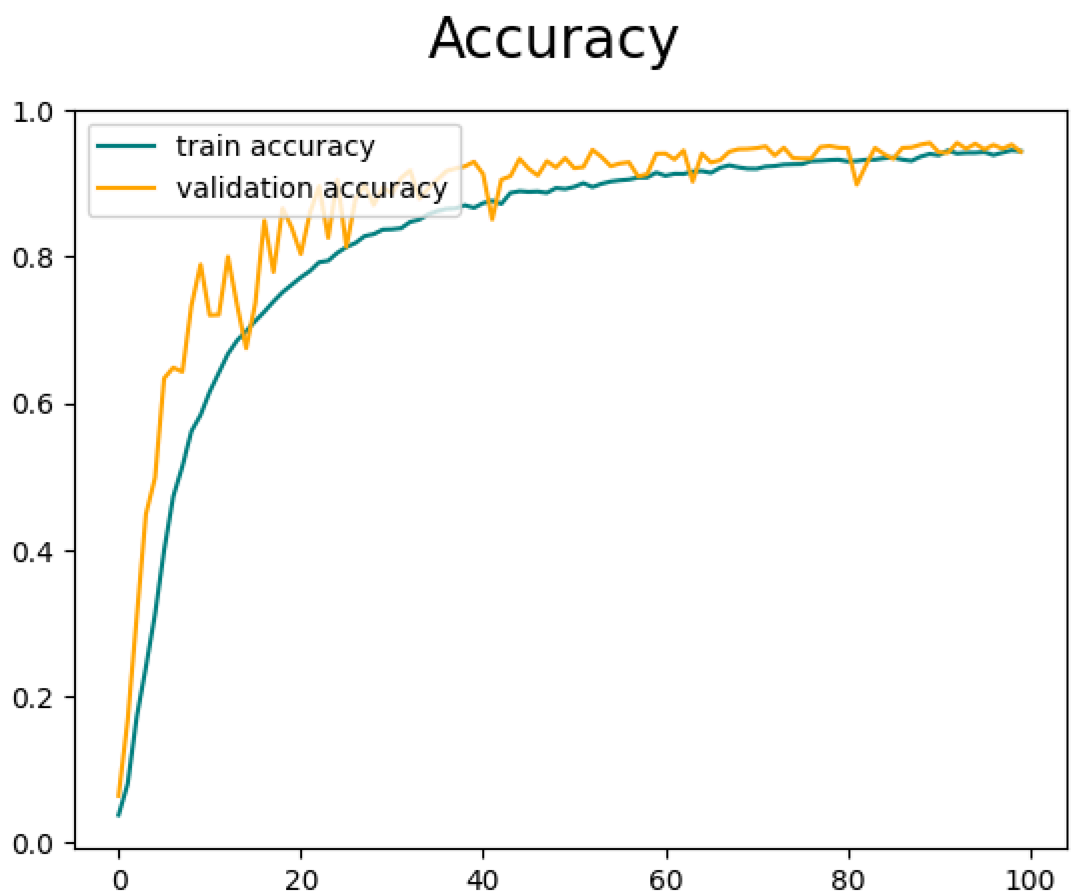
Loss Plot



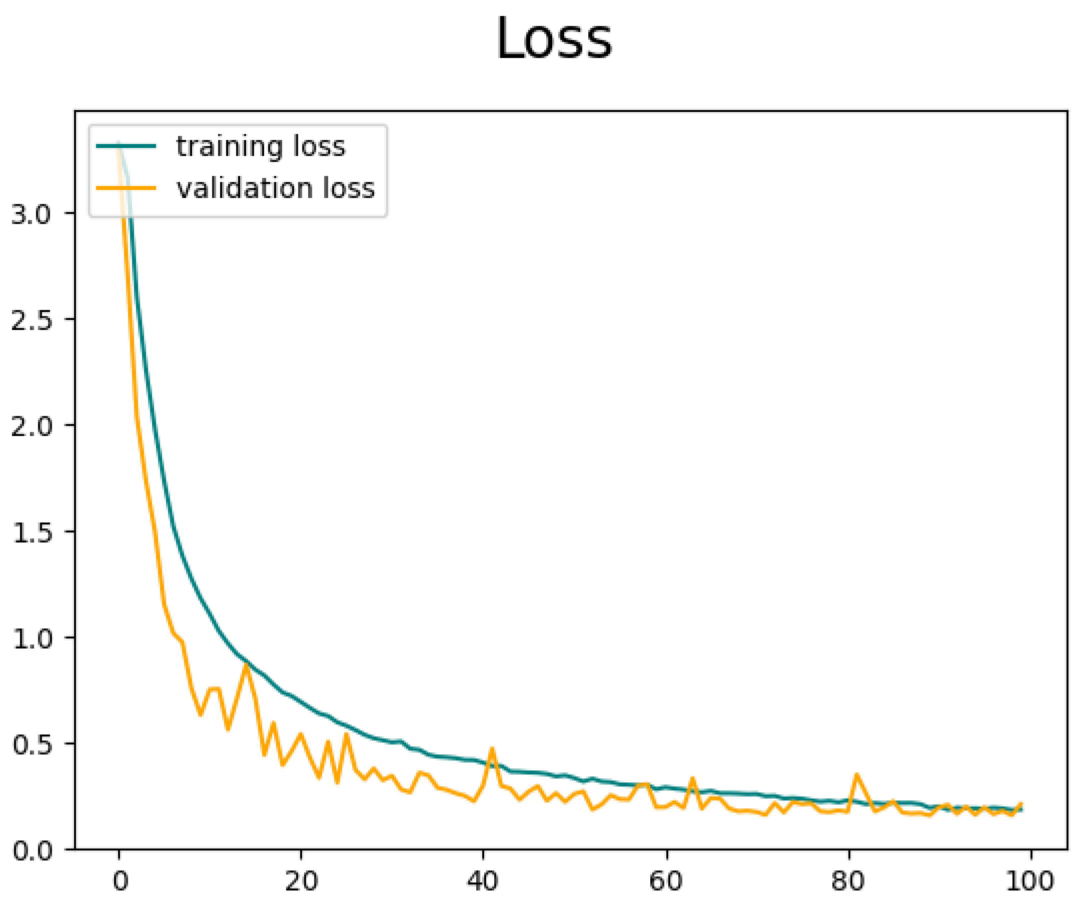
We can see that CNN-14 has a great performance. Reaching an accuracy of over 94% on the testing data. The network doesn't have many trainable parameters.
The CNN-14 took very little time to train due to its simple structure.

Data Augmentation

Accuracy Plot



Loss



We can see that by augmenting the data for our model, the final accuracy has increased as the accuracy has peaked at over 95.5%. And the Loss is much much less than the loss in Part 1.

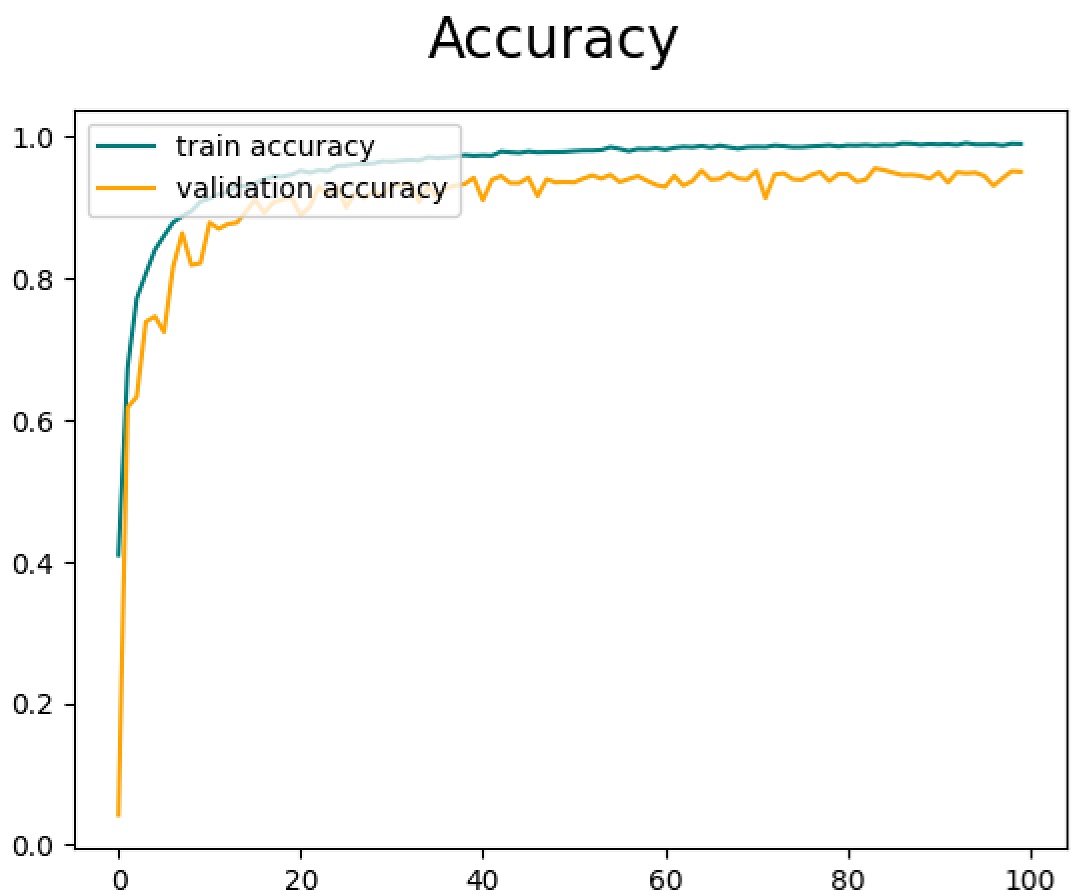
The model took longer to converge using Data Augmentation probably due to the large amount of data as well as lower learning rate (1/10th of part 1) which was selected to carefully adjust weights as there could be more meaningful features in the augmented images.

The training time was also longer due to larger training data.

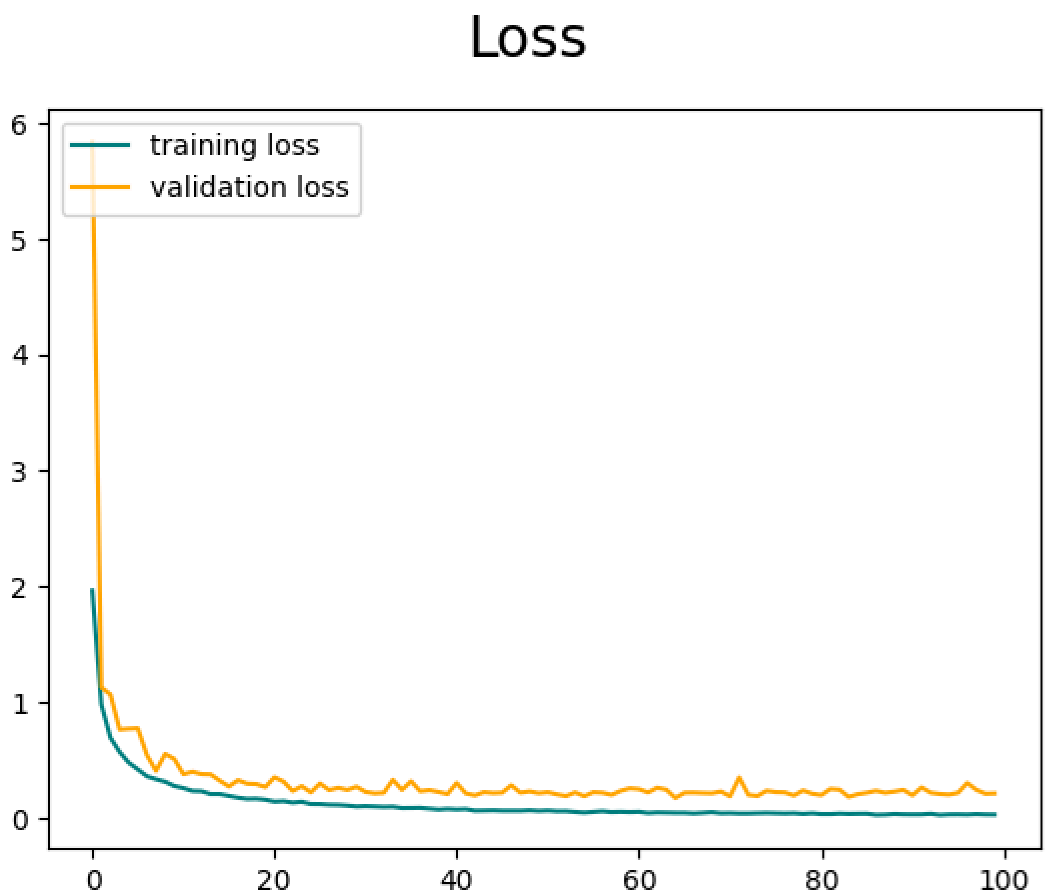
Can also notice that this model eventually performed better on testing data compared to training data (Accuracy & Loss).

DenseNet 121

Accuracy



Loss

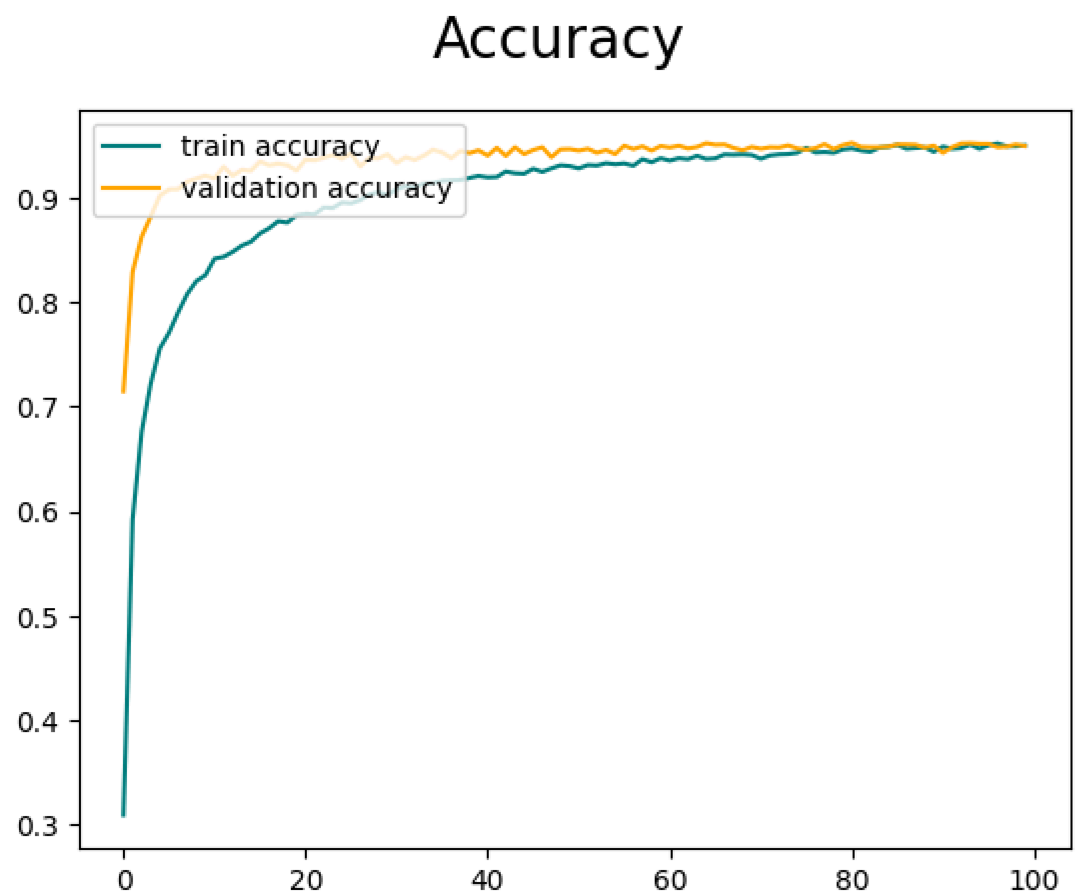


The selection of DenseNet121 was based on the fact that it doesn’t have many operations required and smaller number of trainable parameters relative to other common CNN structures whilst also maintaining a high accuracy.

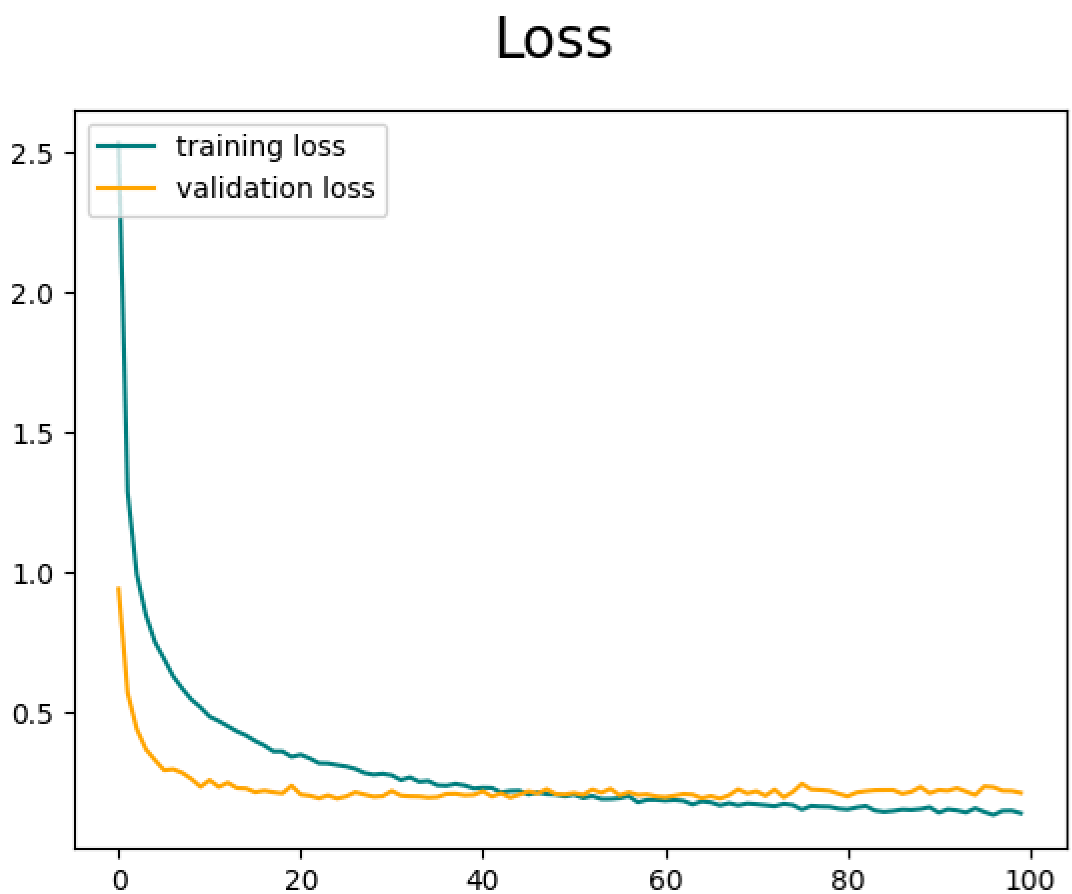
This model was trained on the augmented data as well, it reached an accuracy of over 92% on the testing data. Which is not great compared to previous models that are specifically built to work for this Topic.

Transfer Learning

Accuracy



Loss



This transfer learning model was built on top of a model that classifies English Numbers. I decided to fine-tune the entire model since:

1. My dataset is large enough, especially with data augmentation
2. The model is very simple and doesn't take much time to train
3. The model was not trained on many epochs, the initial model had 80-85% accuracy on its testing data

The model eventually converged to an accuracy similar to the Part 2 model, it achieved over 95% accuracy on its peak but it took quite a lot of epochs to reach that because I had started training it with a low learning rate.

Conclusion

From the project we concluded a few points:

1. CNN-14: CNN-14 was used for part 1 & 2 of the project and it proved its' effectiveness and high performance.

2. Data Augmentation: Data Augmentation was found to be a very effective technique to increase the performance of the model and decrease overfitting problems. Granted it takes longer to train due to larger training data size.

3. DenseNet 121: This model was doing alright given it's a general CNN structure that is not built for a specific task, although the testing accuracy was much lower than parts 1 & 2 and the testing accuracy was much lower than the training accuracy but overall it's an alright model for classification tasks.

4. Transfer Learning: Transfer learning was a great way of developing a model to classify our images, we did not have to worry about the model structure, parameter count or anything. We could just fine-tune parts we wish and it gets high performance metrics.