**Kannada MNIST Project Process**

**Project Overview**

The project data consists of numbers from Kannada alphabet as 28x28 pictures (from 1 to 10), and the numbers it represents. The data I used are the 60,000 training samples and the 10,240 evaluation samples.

**Data Analyzation and Choosing the Model Type**

First, I analysed the given data. The format of the files is csv files where first column is the label of the sample and the next 28x28=784 columns were the grayscale values of the pixels, from 0 to 255; each row being a different sample. From this, I figured how I would ready the data for the model training.

By using the mathplotlib, I viewed some of the samples and from the focus of the challenge and from the general idea of the samples, I decided to use CNN model to get a better accuracy, since the only limitation on the model would be my hardware for this project.

**Data Preparation**

The format of the data given to me made the data preparation rather easy.



Figure 1: Preparation of Raw Data

All I needed to do was to reshape the samples from a single vector to 28x28 pictures, also one more dimension to determine the colour, which is 1 dimension. The single vector format was given in a way that a single reshape function was enough to recreate the pictures. With the raw data ready, I started to build the model.

**Model Design and Training**

For model design, I usually start with either an already established complex model or a basic model depending on the data. The hardware accessible to me is limited, so I need to see approximately how much time it takes to train one epoch for the training data set. In this case, I was confident that my computer could train the model in a relatively short time frame. Therefore, I started off with a modified version of my CNN model that I used at one of my previous projects.

Then I start tinkering with the model, not fully realizing what does what, and end up with a similar model, which produced similar results. Making the model more complex did not help at all. The validation accuracy was around %76 to %81. I focused more on validation accuracy as accuracy from the training data could sometimes be misleading.

**Data Augmentation**

In my previous projects, I refrained from adding more training samples artificially, as the objective of the previous projects were to understand model building and not getting a better accuracy only. However, in this project, I wanted to get the best accuracy I can achieve, therefore, I decided to augment the training data.

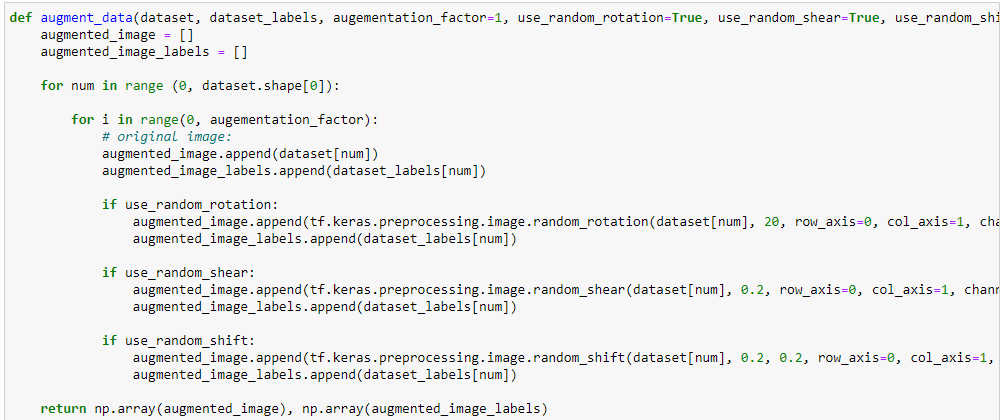


Figure 2: Data Augmentation Function

With the data augmentation, I increased the training data sample amount to 240,000. The accuracy improved by around %5.

**Revisiting the Model**

Even though I managed to get a good accuracy boost. I still wanted a better accuracy rating. Therefore, I revisited the model I built. Up until this moment, I made the models with only testing, not really understanding the layers, their purpose, and their limitations. So, I started to understand why the layers worked the way they do.

* I ended up building the convolution layers so that I would get 3x3 2d convolution outputs.
* I decreased the 2d max pooling layers to cut the batches less in half.
* I ended up increasing the filter amount with each convolution layer instead of keeping it as a constant amount.
* I realized that batch normalization basically normalizes the values of the outputs of a layer, so I simply put them after convolution layers and removed the ones after the normal dense layer.
* I decreased the dense layers to 2.
* I realized that the Flatten layer was in the wrong place, so I put it right after the convolution layers.

With more understanding of the layers and the model, I trained the model.

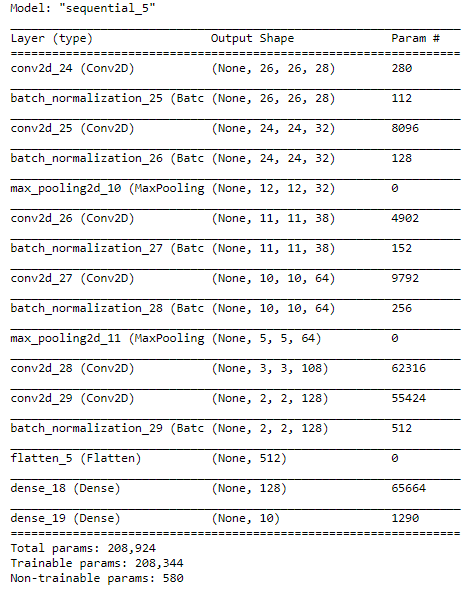


Figure 3: The Final Model Summary

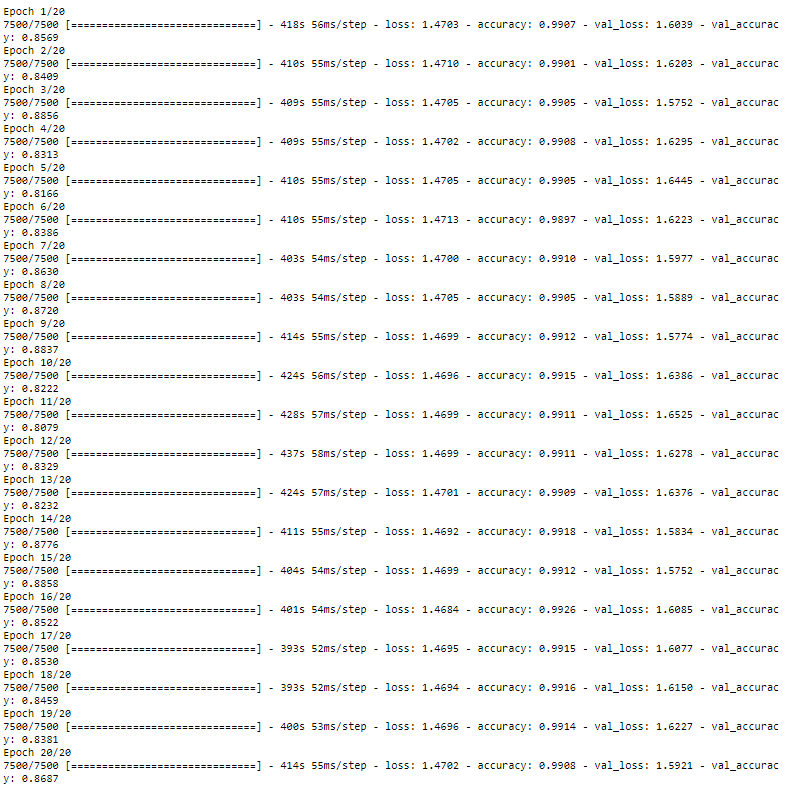


Figure 4: The Training Report

This model managed to get %86.9 success on the evaluation dataset. Given the time it took to train the model, I decided to be done with working on the model. With a better hardware, I would try to get a better accuracy than this. However, the point of this exercise is to show the details I consider while creating a model. This exercise taught me a lot when it comes to machine learning and it gave me something to work on, so I am happy for it. Also, after finishing this model, I decided to look up the models on the discussion and try to train one of them. My computer could not handle it very well. Thus, I decided to stay with this model period.