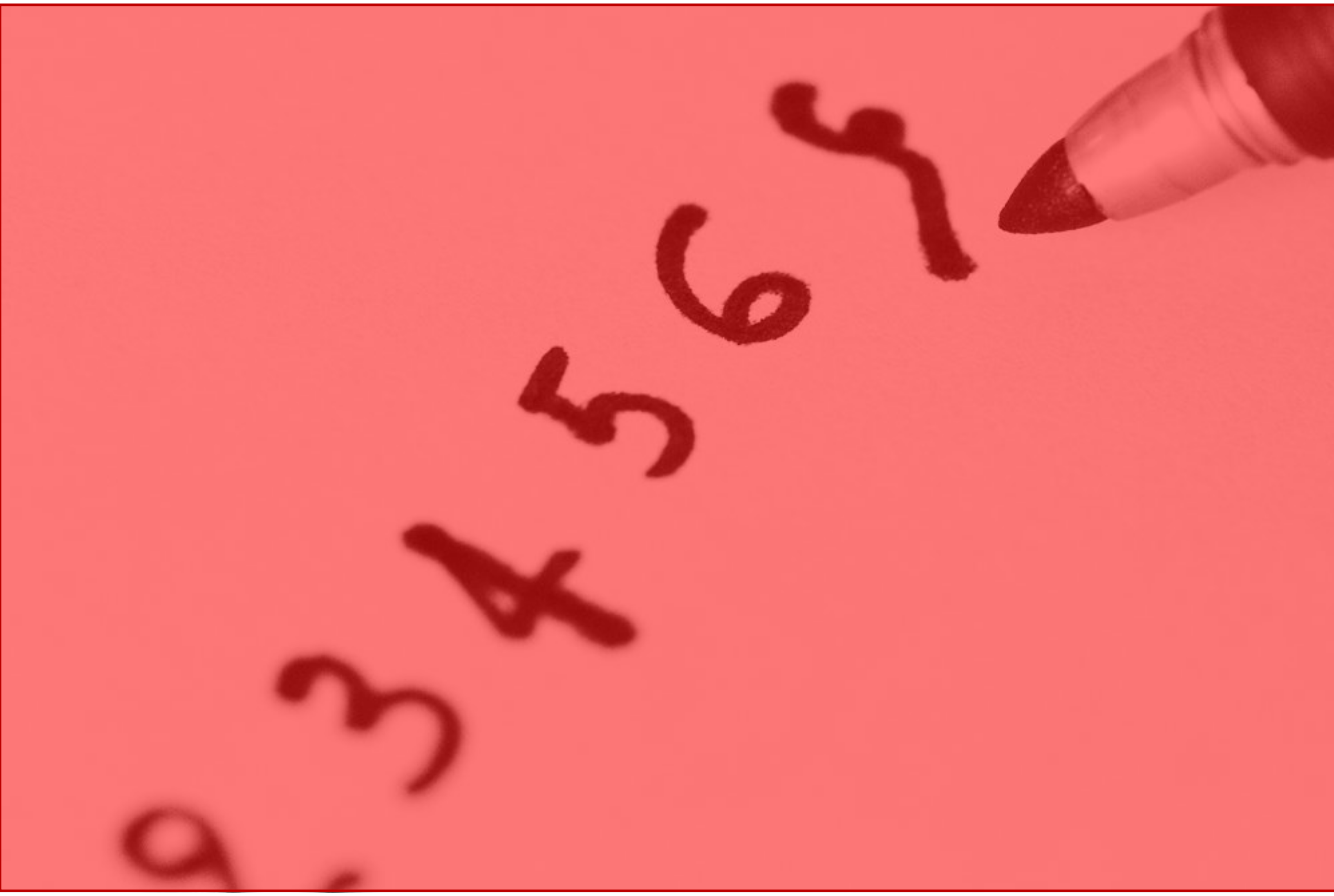
MNIST DIGITS RECOGNIZATION

-BY SHIVAM KUMAR GIRI

AIM

The objective of this study is to classify the handwritten digit which are categorized into 10 classes i.e. 0-9

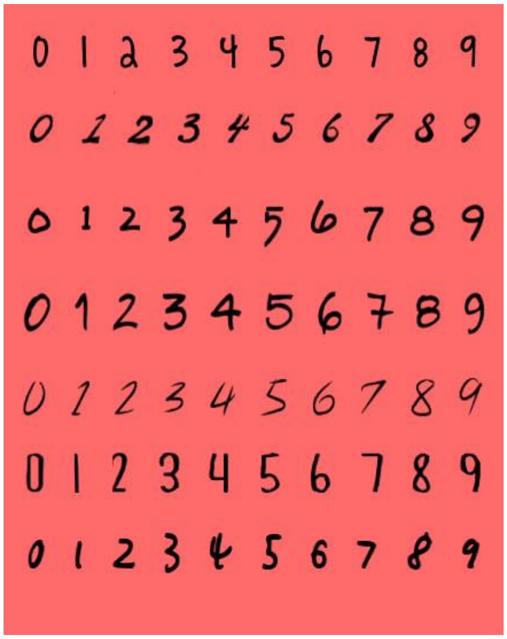
ABSTRACT



Here I designed a Machine Learning model using python which can detect the handwritten digit image at a particular time, based on the specs of digit, i.e. Curves and lines. It takes the image/ pixels of image as input and output the digit based on image specifications. There are 10 digits: 0-9 which are detected by this application.

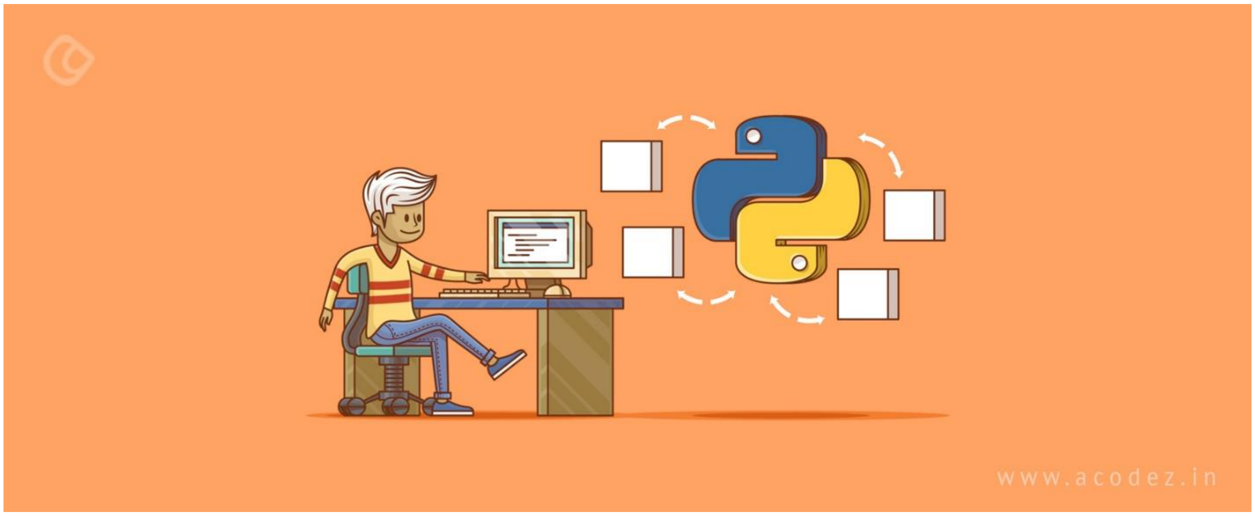
DATASET

MNIST ("Modified National Institute of Standards and Technology") is the de facto “hello world” dataset of computer vision. Since its release in 1999, this classic dataset of handwritten images has served as the basis for benchmarking classification algorithms. As new machine learning techniques emerge, MNIST remains a reliable resource for researchers and learners alike. The MNIST database contains 60,000 training images and 10,000 testing images. The dataset consists of 785 columns with 1 column consist of class and other consist of image data. The image data consists of 28x28 pixel grayscale images of faces. The pixel values are stored in 784 (28*28) columns. These column names start with pixel. Along with pixel values, there is class column that say about digit depicted in the image.



There are 70000 rows of data, i.e. 70000 distinct data, each with the digit and pixel values of image. We will use this data for training, validation and testing with the ratio 8:1:1 of the data present in the dataset.

LIBRARIES USED



Here is developed the application using python. I used the following Libraries for implementation of the mood detector:

Numpy: for Numerical computing

Pandas: for data manipulation and analysis.

Tensorflow: for dataflow and differentiable programming

Keras: for neural-network

Pillow: for Image Processing.

Matplotlib: for data visualization

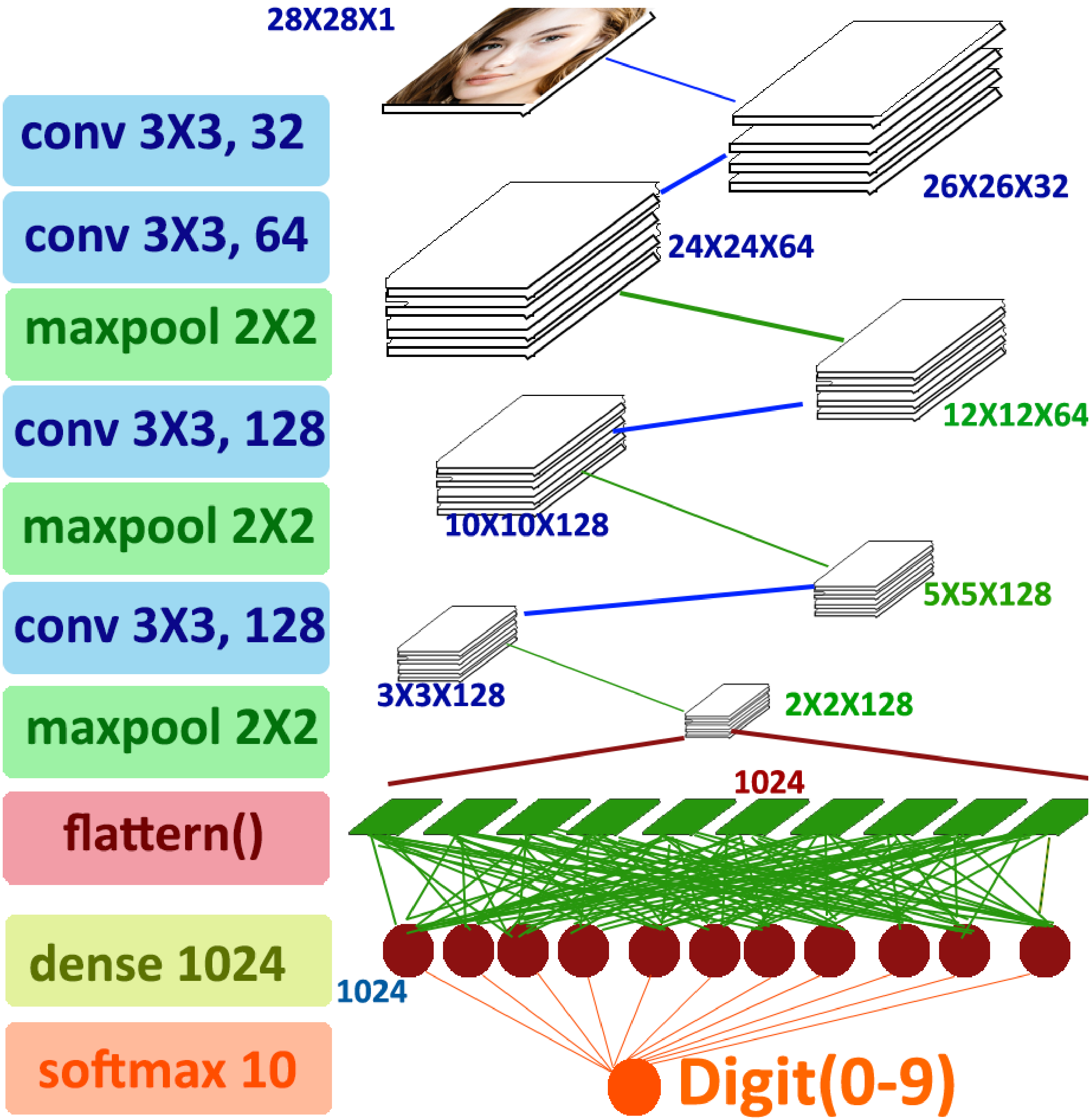
Seaborn: for data visualization and heat maps.

Sklearn: for classification algorithms and other Machine Learning metrics.

TRAINING DATA

The training data, validation data and test data are in ratio 8:1`:1. I used Keras sequential model for modelling my Convolution Neural Network.

I used following CNN model in my application:

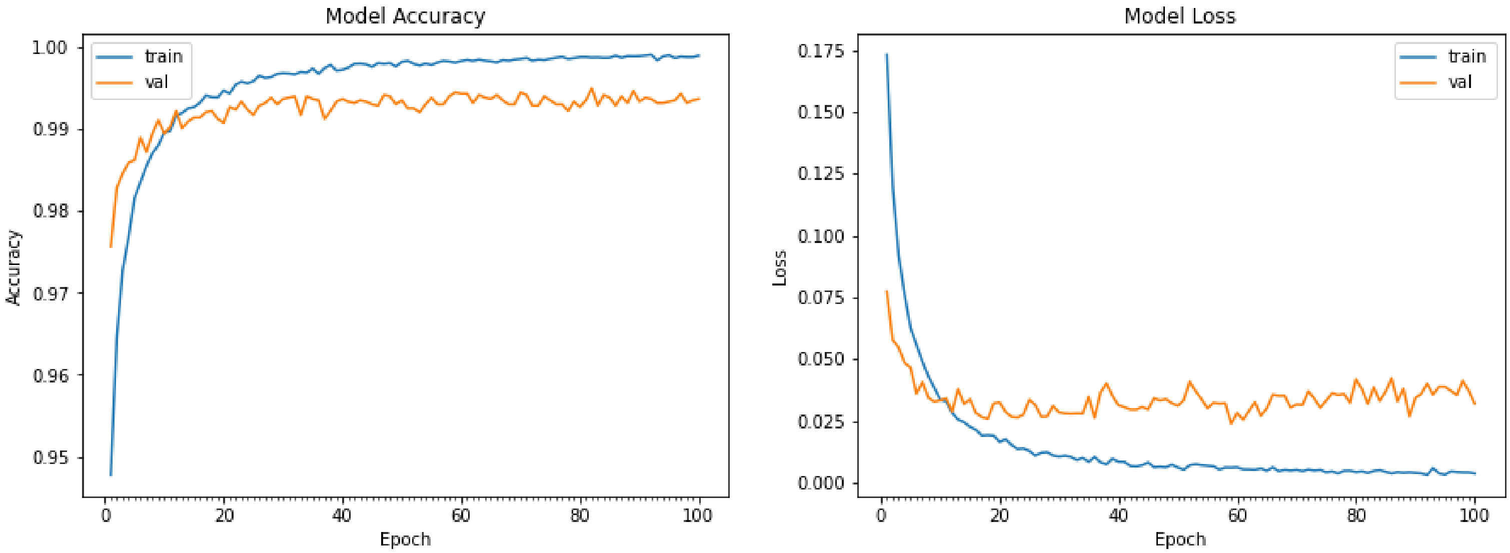


The Model is hence trained with 100 epochs to get maximum accuracy.

VALIDATION & TESTING DATA

10% of dataset is used for validation and 10% for testing.

For testing the following accuracy and losses at each step of validation is recorded below:



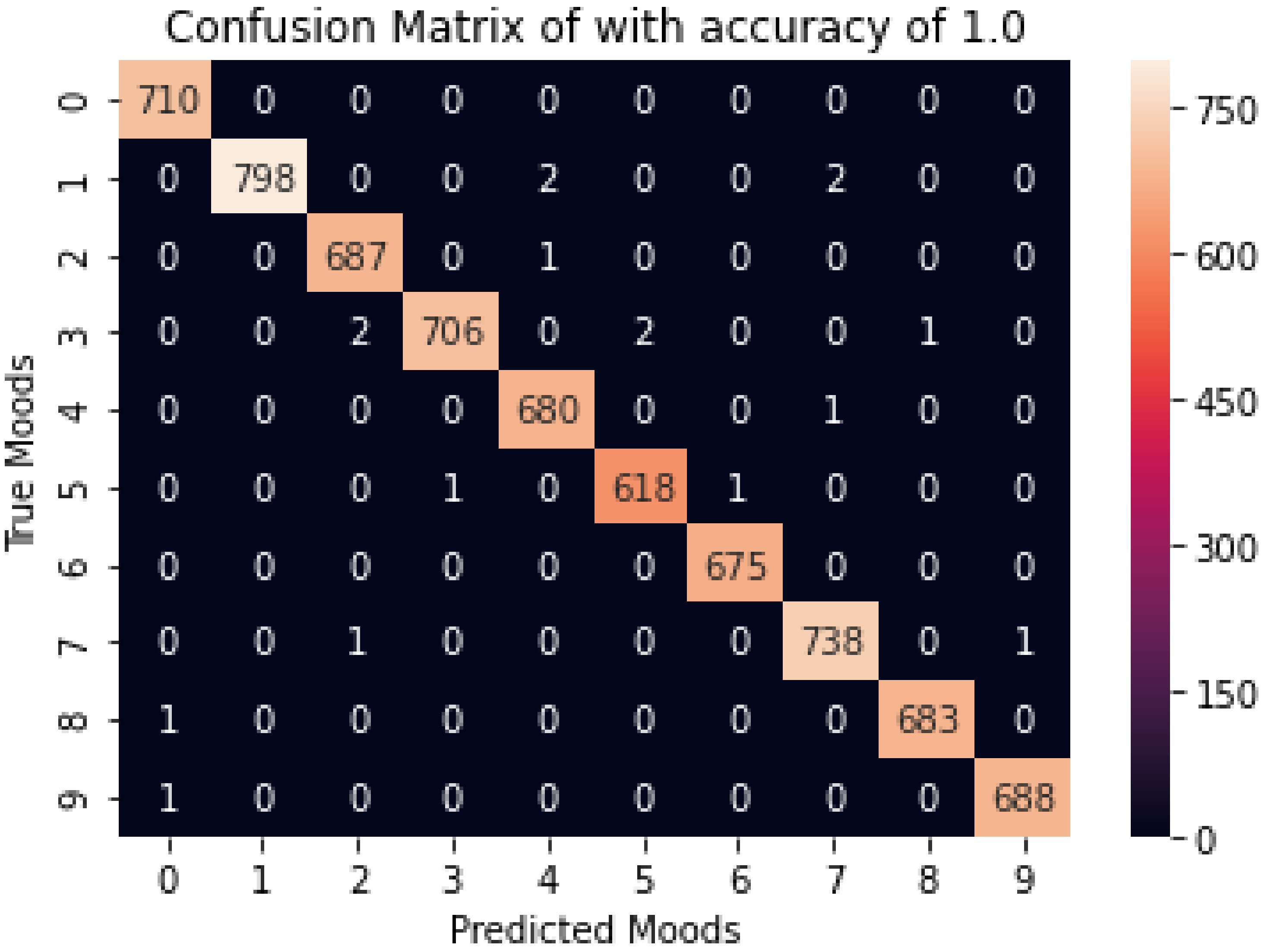
Hence, we saw that accuracy for the graph is reaching to the peak upto 1.0 i.e. a very good accuracy achieved with validation reaching upto 0.99.

Also, the losses for each of them is minimal and can be view upon the loss per epoch graph above

Since our epoch is 100, it measured validation in 100 steps, and plotted the graph.

We saved the weights of train model in file model.h5. For testing we imported the weight and used.

We found the accuracy of the model as 99.75% and the confusion matrix as printed as below.



	0	1	2	3	4	5	6	7	8	9
0	710	0	0	0	0	0	0	0	0	0
1	0	798	0	0	2	0	0	2	0	0
2	0	0	687	0	1	0	0	0	0	0
3	0	0	2	706	0	2	0	0	1	0
4	0	0	0	0	680	0	0	1	0	0
5	0	0	0	1	0	618	1	0	0	0
6	0	0	0	0	0	0	675	0	0	0
7	0	0	1	0	0	0	0	738	0	1
8	1	0	0	0	0	0	0	0	683	0
9	1	0	0	0	0	0	0	0	0	688

CONCLUSION

Our application of face detection is ready with accuracy of 99%. Hence it can detect the digit very accurately from 0-9.

THE END