Deep Learning Based Recognition of the Script of a Language

A Case of the Devanagari Script

Ruchika Sah

Mar 13 , 2020

Agenda

- Problem Statement
- The Devanagari Language
- Characters
- CNN Model
- Results
- Evaluation
- Conclusions

Problem Statement

The need to recognize Handwritten Text is a challenging problem especially in the context of pattern recognition. Character classification is an important part of Optical Character Recognition like Licence Plate Recognition, House Number Recognition, finding fraudulent checks etc. This project focuses on reviewing the existing methods for Character recognition of handwritten characters of the Hindi Language using Machine Learning Methods. The script of the Hindi language is very different from the English / Latin one and is called the Devanagari script. I will be using the Convoluted Neural Network (CNN) model for image recognition. The model will be evaluated using the Accuracy score.

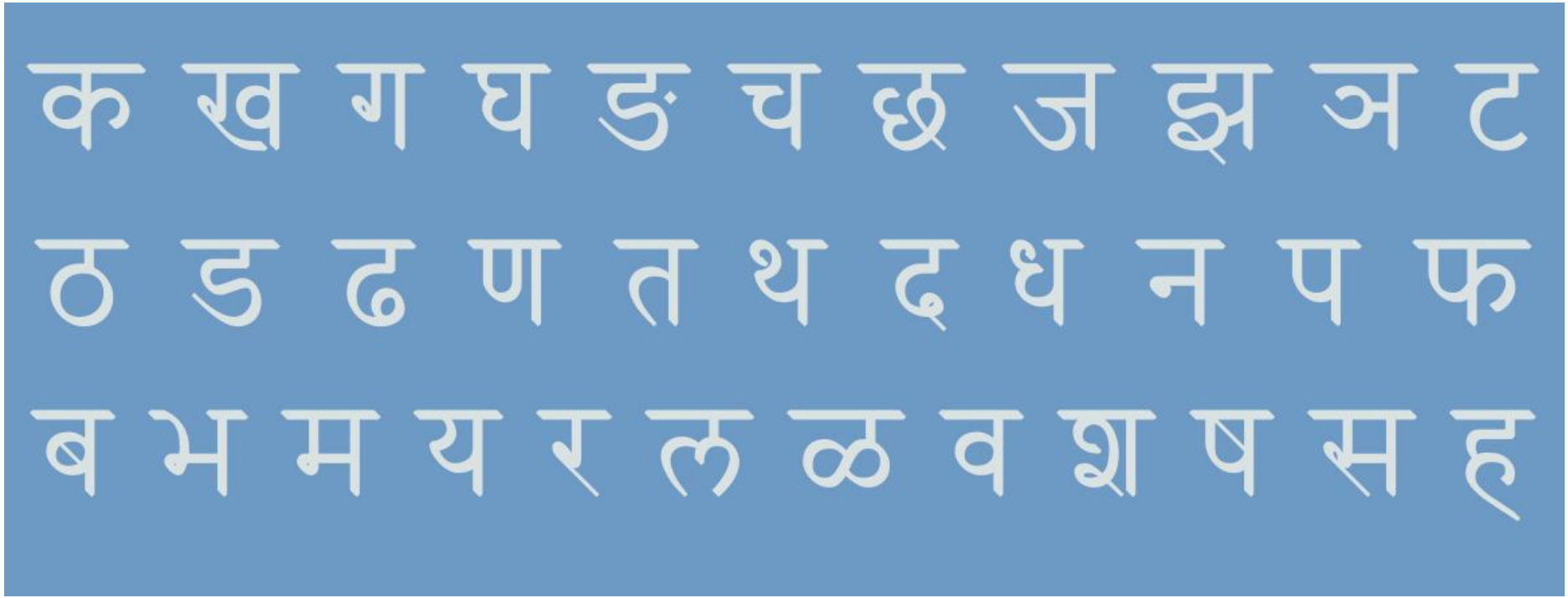
Devanagari Script

- Hindi is the popularly used language of India and is the third most popular language in the world,which is written and encoded using Devanagari script.
- The Devanagari script was developed in ancient India from the 1st to the 4th century BC and was in regular use by the 7th century BC.
- Unlike the Latin alphabet, the script has no concept of letter case (Upper or Lower).
- It is written from left to right, has a strong preference for symmetrical rounded shapes within squared outlines and is recognizable by a horizontal line that runs along the top of full letters.

Source : Wikipedia

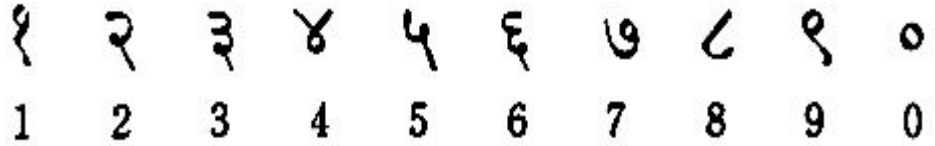
Devanagari Script

Consonants



Devanagari Script

Numbers



Vowels

An example of a sentence

मेरा नाम रुचिका है

My name is Ruchika

Our Data

- The 92000 images of this dataset were generated by imaging the characters written by many individuals resulting in a wide variation in the way each character was written.
- Each of the dataset images of the characters is unique.
- The dataset is randomly split into a training set with 85% of the data and a testing set with 15% of the data.
- The Training Set consists of 78200 images and the testing set consists of 13800 images.
- Each image is 32 x 32 pixels with the actual character centered within 28 x 28 pixels.

Challenges of the Data

There are many pairs of characters in the Devanagari script that are very similar. They are differentiated by structures like dots, horizontal lines, loops etc. Some examples of that are:

ढ and द

छ and ६

क and फ

This problem is more intense with handwritten characters

घ

ध

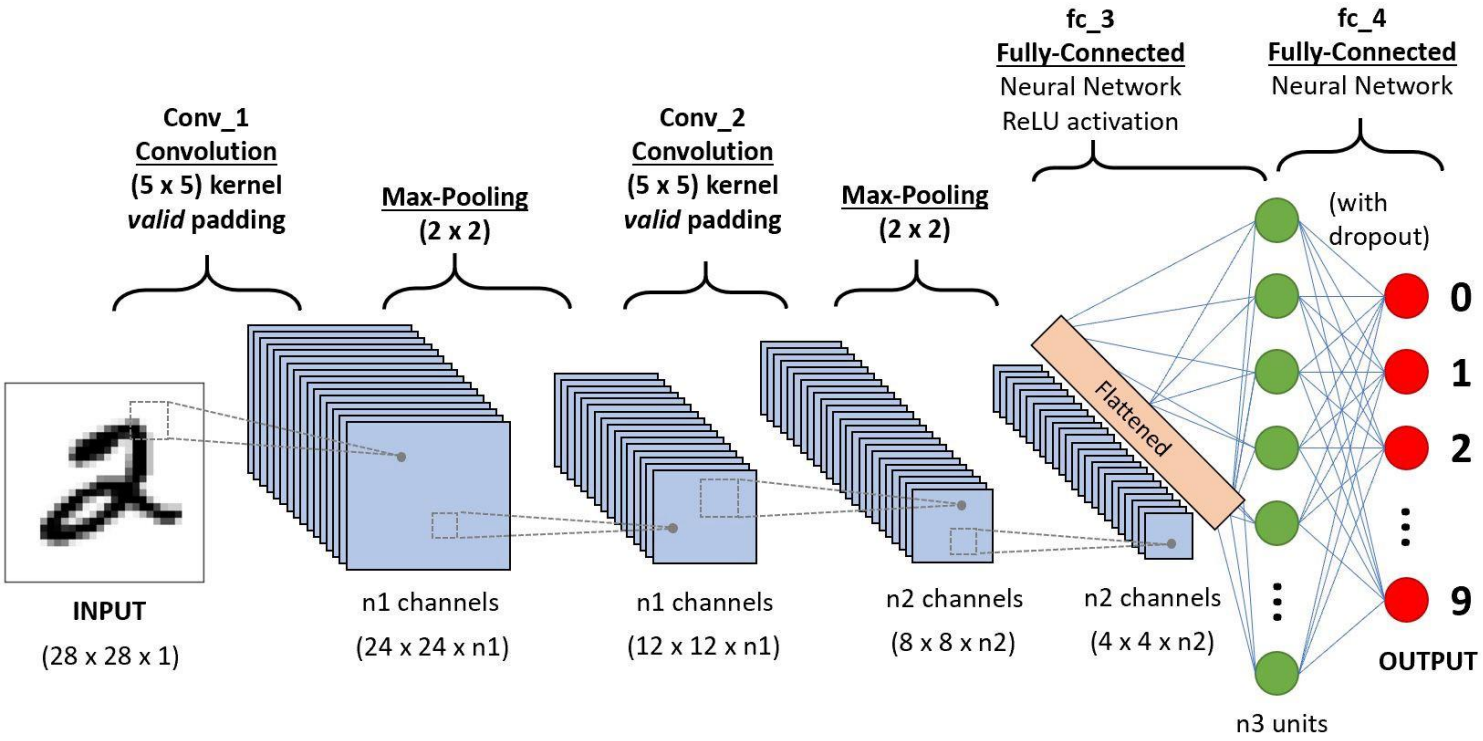
प

य

How does the CNN work

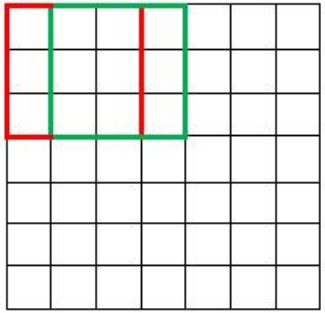
- An image is converted into an array of pixel values.
- The image is then passed through a series of convolutional, nonlinear, pooling (downsampling), and fully connected layers to get an output.
- The objective of the Convolution Operation is to **extract the high-level features** such as edges, from the input image.
- The Pooling layer is responsible for reducing the spatial size of the Convolved Feature and **decrease the computational power required to process the data** through dimensionality reduction. **Max Pooling** returns the **maximum value** from the portion of the image covered by the Kernel.
- Then we flatten the final output. This is fed to a feed-forward neural network and backpropagation is applied to every iteration of training. Over a series of epochs, the model is able to classify them using the **Softmax Classification** technique.

Convolutional Neural Network Model

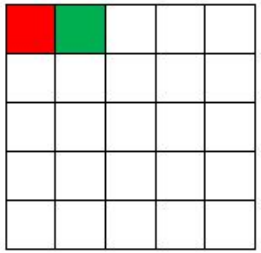


The Stride Parameter

7 x 7 Input Volume



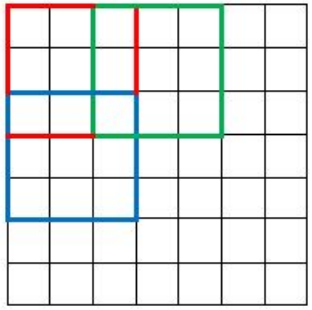
5 x 5 Output Volume



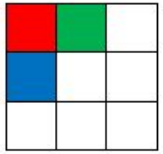
A 3 x 3 filter with stride=1

A 3 x 3 filter with stride = 2

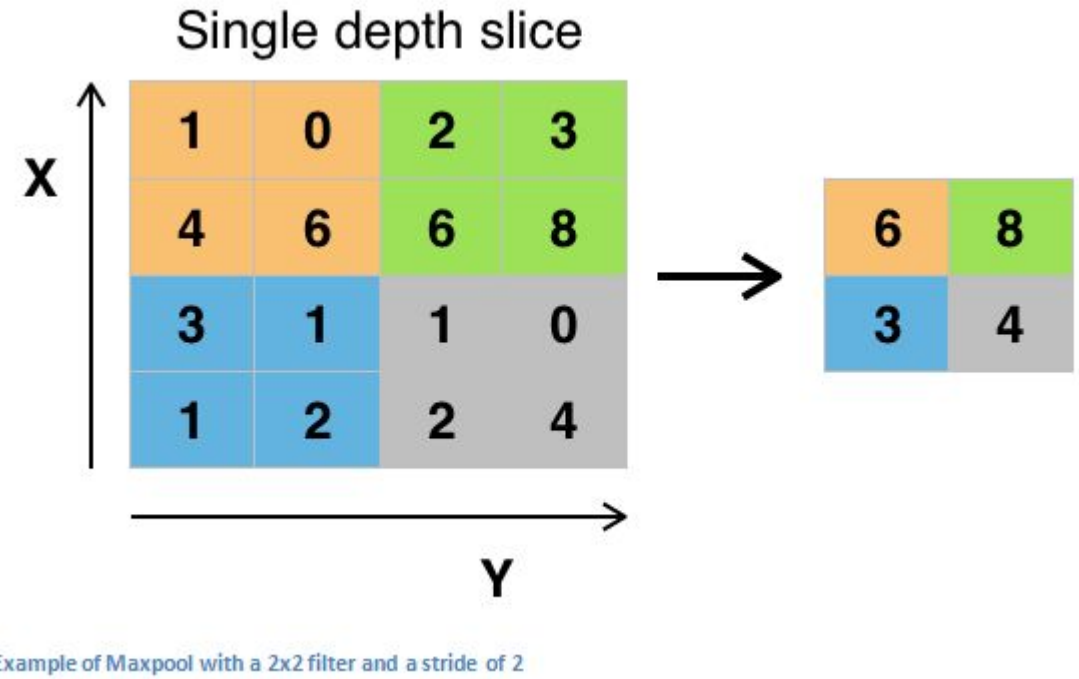
7 x 7 Input Volume



3 x 3 Output Volume

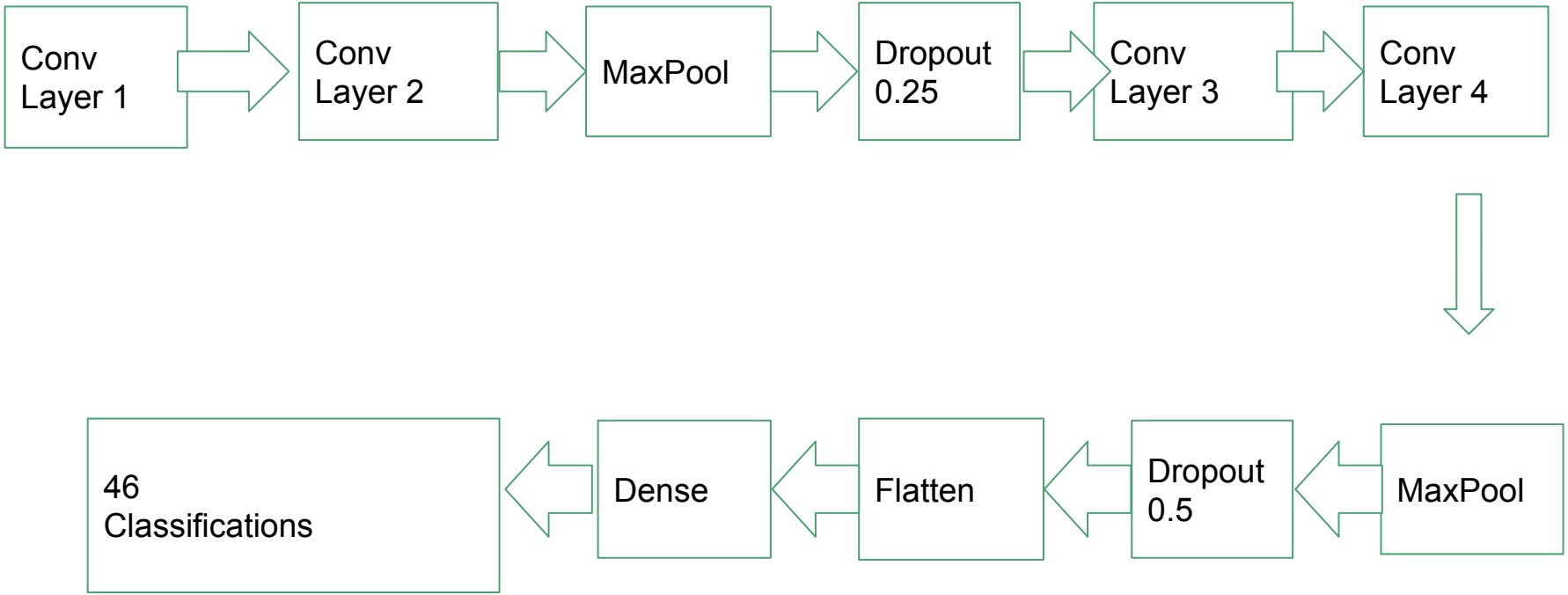


MaxPooling Layer

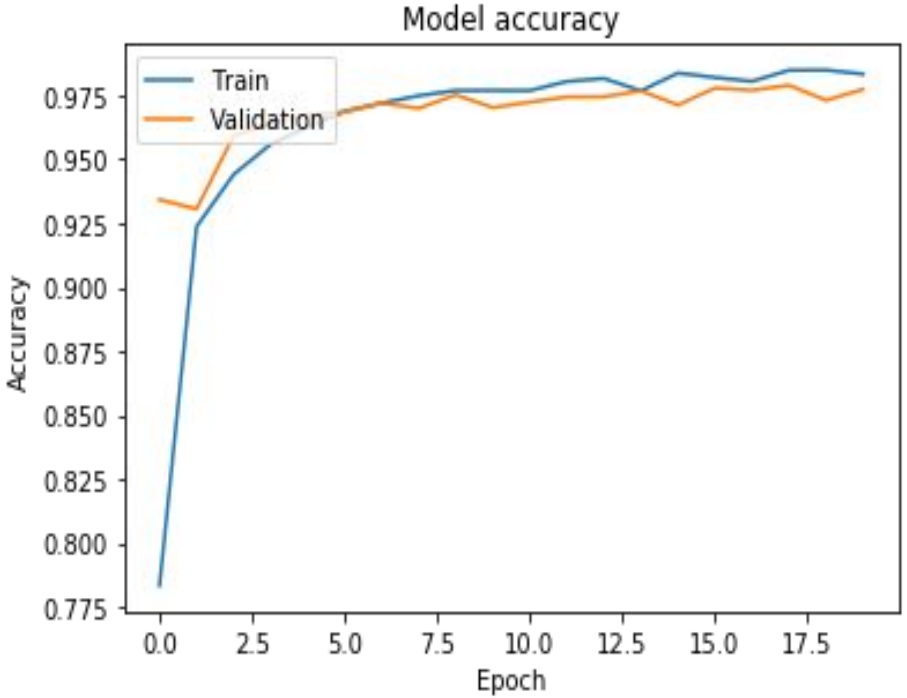


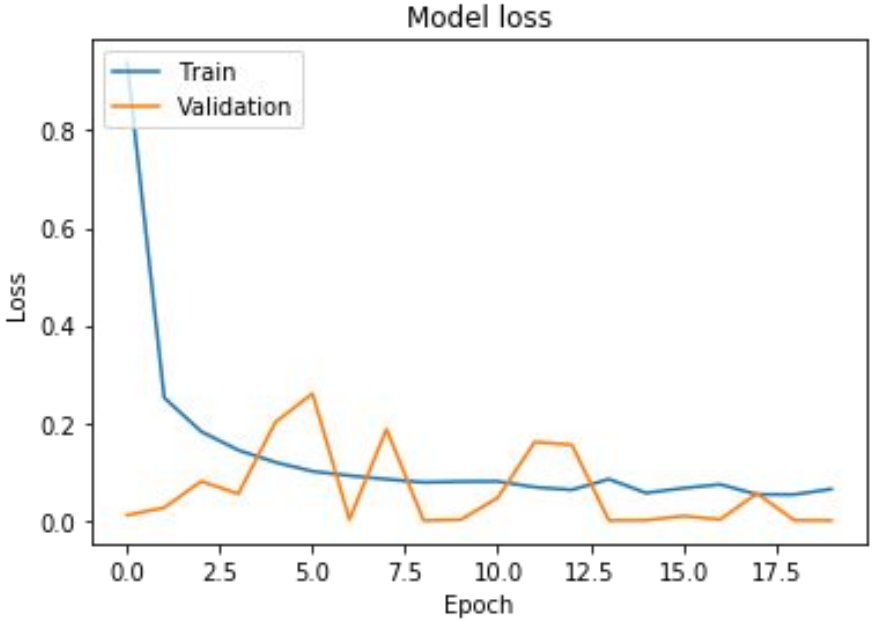
Example of Maxpool with a 2x2 filter and a stride of 2

My CNN model

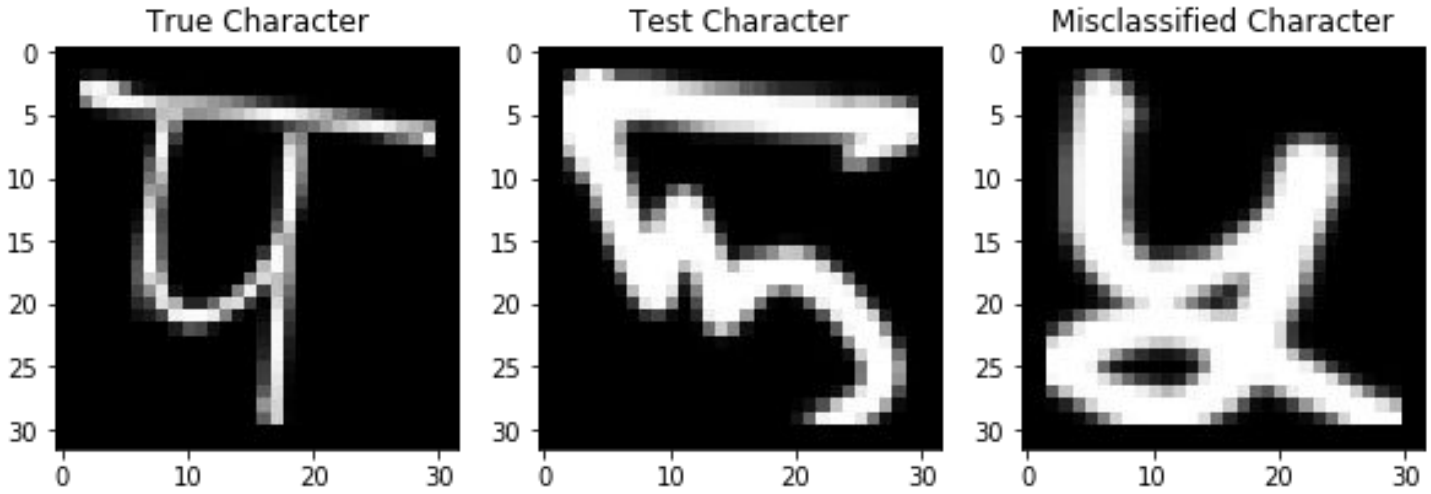

Conv Layer 1
Conv Layer 2
MaxPool
Dropout 0.25
Conv Layer 3
Conv Layer 4
MaxPool
Dropout 0.5
Flatten
Dense
46 Classifications

Accuracy Score - 98.3 %. From a test set of 13,800 images, the model got only 312 wrong

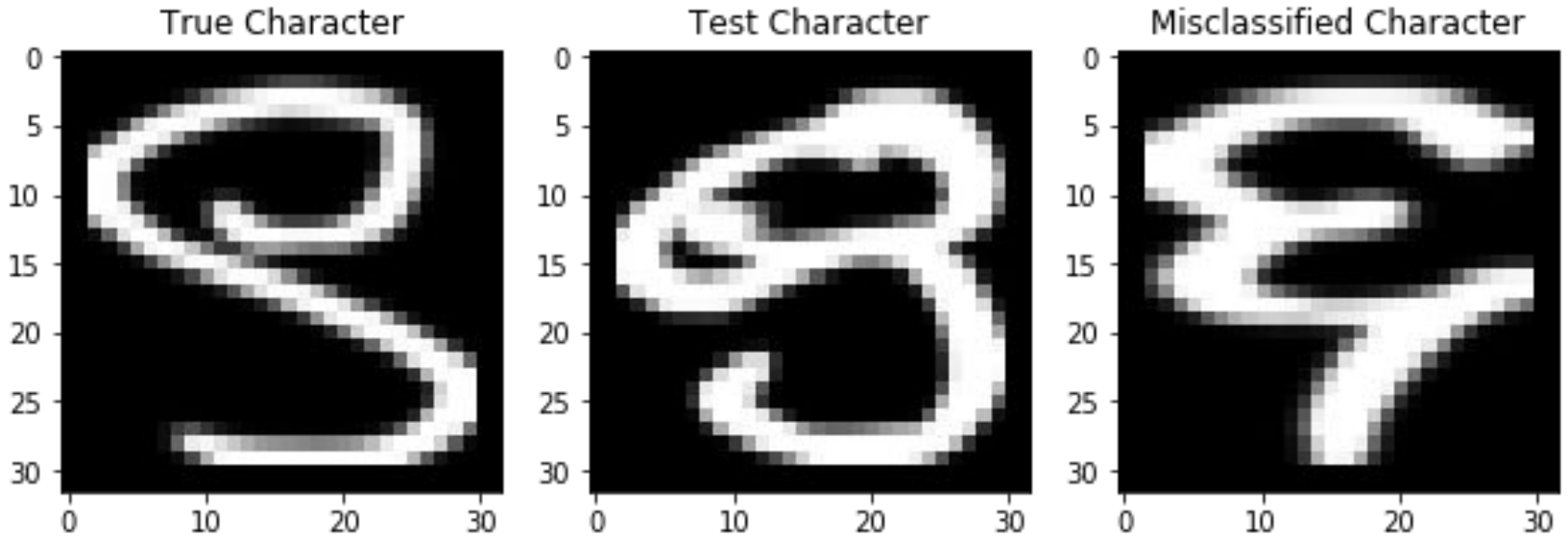




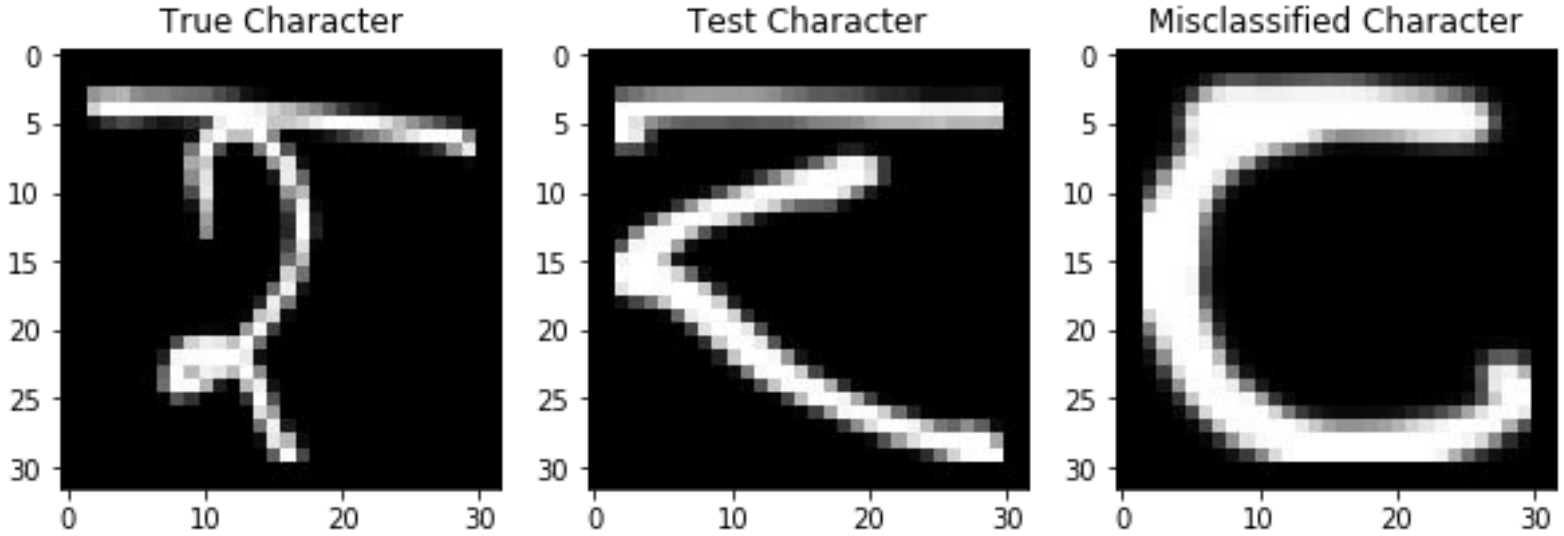
Misclassified characters - The test character is something that even a human would not understand



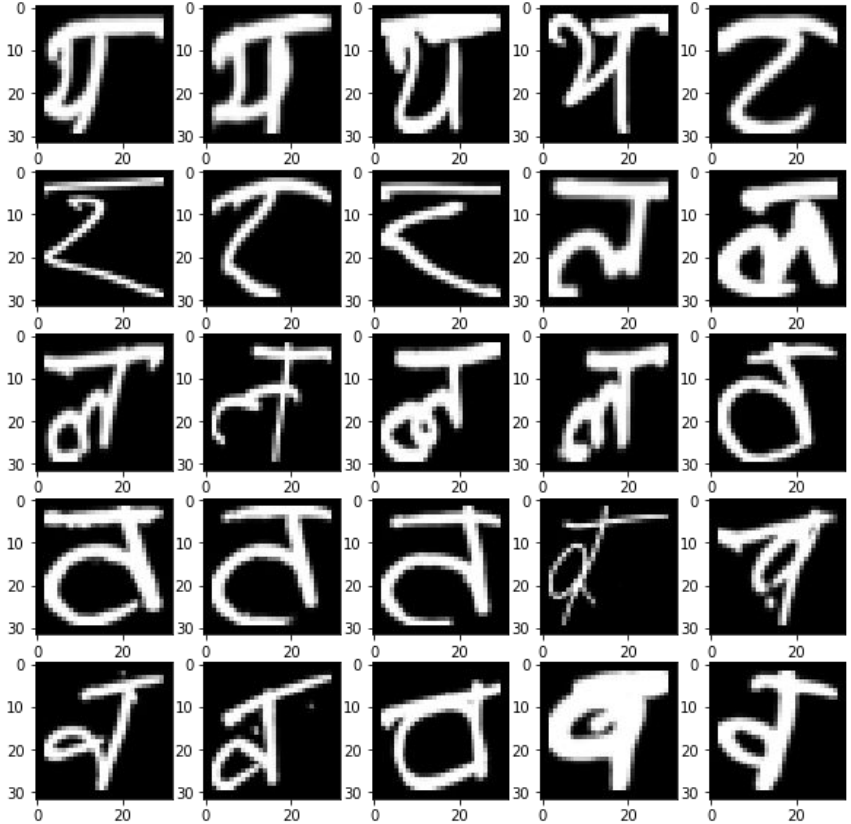
The True character is a 9 which was misclassified as a 6



Misclassified character - 8 instead of Ra



Some misclassified handwritten characters



Conclusions

- In spite of the characters in the language being so similar and complicated and given the variances in handwritten characters, the model performed very well at 98.3% accuracy. The validation accuracy was 97.7%

- Out of the 13800 test characters (with 300 per class and 46 classes of characters), it misclassified only 312.

- This shows that Deep CNN models with added Pooling layers and Dropout layers can result in a high test accuracy.

Next Steps

- This dataset included 46 characters, consonants and numbers, but did not include the vowels. For the future, this dataset could be extended to include the vowel characters as well.

- This model could also be extended to other languages like Mandarin or Japanese which have more difficult characterizations.

- It could also be implemented to design a complete handwritten document digitizing system.

Thanks - Questions?