Title: **Classification of Letters in Indian Language (Devnagari)**

Name: Bhushan Suryawanshi

Date: 2021-03-02

# **Executive Summary**

Handwritten character recognition is an important part in computer vision domain. Handwritten character recognition is least explored for the Devnagari character (Indian Language). There are different techniques used in handwritten character recognition. Several researches have been done to develop a system which will provide a good accuracy. There are two techniques of character recognition system namely online and offline character recognition. Here we used offline handwritten character system. In offline character recognition system handwritten characters of user are available as image.

Handwritten character recognition can benefit multiple industries. Some of those are -

Consumer: With smartphone and tablets, consumer industry need something to minimize spelling mistakes on digital keyboards and handwritten character recognition can help in great way.

Education: Using handwritten technology, students can benefit from more than just the increased comprehension linked to taking handwritten notes. Handwritten technology is can be used in note taking, math and even music apps.

For this character classification problem, the dataset used is having 78,200 images of 46 Devnagari characters. The data is split into training and testing set for model training and verification. Using current advancement in Deep learning using Convolutional Neural Network (CNN) helped to achieve greater accuracy.

**Keywords**: Deep neural networks, Devanagari Handwritten character recognition, Machine Learning, Convolutional Neural Networks

## **Problem Background**

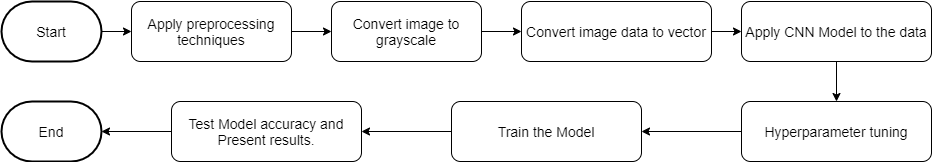
With advancement in Machine Learning technologies character recognition is done in many different forms for English language. However, same is not the case for Indian languages which are complicated in terms of structure and computations. Rapidly growing computational power may enable the implementation of Indian Language CR methodologies. Automating digital document and recognition of text using images has application to office and library automation, bank and postal services, publishing houses and communication technology.

## **Problem Statement**

Since the use of CNN most of the work is done for English language. Devnagari being the national language of India, spoken by more than 500 million people, should be given special attention so that document retrieval and analysis of rich ancient and modern Indian literature can be effectively done.

**Methods**

Convolution neural networks are a breakthrough in image recognition. They’re most commonly used to analyze visual imagery and are frequently working behind the scenes in image classification. We followed the CRISP-DM processes as close as possible.



The data was collected from UCI machine learning dataset[2]. The data provided was png images. I collected the data and created a gray scale vector of pixels. I found vector operation is faster than matrix operation. Each character has 1700 handwritten images in training data.

Letters available in dataset -



While looking at the Devnagari characters the structural challenges need to be noted are -

1. Character and Number having similar structure. Look at following where first one is character and second is number 6 in devnagari.



2.Two characters with difference of only one dot -



3.Two characters with difference of small line or circle.



The data is divided in to training and test set with 70% and 30% split. There are different models which can be used for the image classification. However will stick to Convolutional Neural Network (CNN).

The model used is having Conv2D and max-pooling layers. The input layer consists of raw pixel values from the 32x32 grayscale image. Following is the network summary for the reference.

Layer (type) Output Shape Param #

=================================================================

conv2d\_1 (Conv2D) (None, 30, 30, 32) 320

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_2 (Conv2D) (None, 28, 28, 4) 1156

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_1 (MaxPooling2 (None, 14, 14, 4) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_3 (Conv2D) (None, 12, 12, 4) 148

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_4 (Conv2D) (None, 10, 10, 4) 148

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_2 (MaxPooling2 (None, 5, 5, 4) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_2 (Flatten) (None, 100) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 20) 2020

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense11 (Dense) (None, 1024) 21504

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_2 (Dense) (None, 46) 47150

=================================================================

Total params: 72,446

Trainable params: 72,446

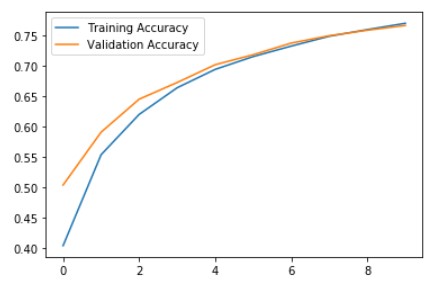
Non-trainable params: 0

The pooling layer reduces the feature map generated by conv2D layer. In this it uses maximum value of pixel resolution. This is helpful to learn the relative position of feature rather than absolute position. This makes the classifier immune to shift and distortion in images.

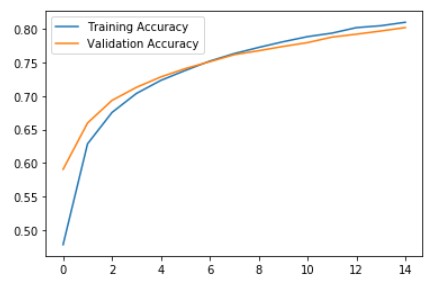
The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

**Results**

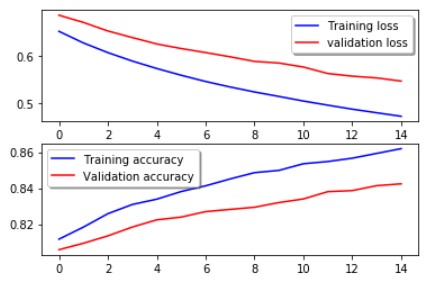
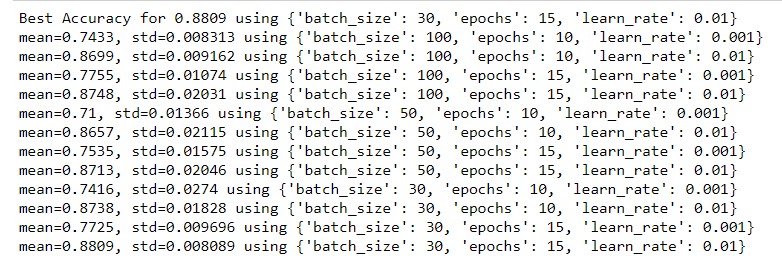
The model with batch size of 100 and 10 epoch gave accuracy of 76%.



When we used smaller batch size the model accuracy increased to 80% on the validation data.



We noticed hyperparameter tuning can improve the model performance. We used GridSearchCv to find the best set of hyperparameters. As per the tuning process it has been observed that model with {'batch\_size': 30, 'epochs': 15, 'learn\_rate': 0.01} has accuracy of 88%.



**Conclusion**

By using the Convolution neural network, we have created a model to classify handwritten devnagari characters. The accuracy of the model is approximately 88% for test data. The precision of the model can be further improved by using more image processing techniques like image segmentation. The overfitting of the model can be reduced by using more data along with some more image processing techniques like segmentation.

**Acknowledgments**

I would like to acknowledge efforts by Shailesh Acharya and Prasanna Gyawali for putting together this dataset. I would also like to acknowledge Prof. Brett Werner from Bellevue University for motivation and guidance on the project work. Also like to acknowledge Anthony LaRosa for his review and suggestion throughout the project life cycle.

**References**

1. Abbott, D. (2014). Applied Predictive Analytics: Principles and Techniques for the Professional Data Analyst. Indianapolis, IN: Wiley.

2. https://archive.ics.uci.edu/ml/datasets/Devanagari+Handwritten+Character+Dataset