International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-4, April 2019

Offline Handwriting Recognition on Hindi using CNN RNN Hybrid Network

A. Balamurali, NacodeVidheesh Kumar, Aniket Yadav, Rushikesh Deshmukh

***Abstract*: Recognising the Scripts written in Hindi is an onerous task due to the delicacies and variations in the script. Handwriting of every script is diversified in writing styles and orientation angles. This requires a neural network with a large training dataset to successfully recognise the script. Till now the development of this handwriting recognisers have been obstructed due to the unavailability of hand-written public datasets in Hindi. In this paper we used CNN RNN hybrid network based on IIIT-HW-Dev dataset to differentiate and recognise handwritten scripts.**

***Index Terms*: CNN-RNN Hybrid Network, Handwriting Recognition, Hindi Handwriting**

I. INTRODUCTION

Hindi is a widely used language in India which is devanagiri based. It has 46 letters, of which 33 letters are consonants and rest 13 are vowels. Shirorekha is the horizontal line placed on top of the word formed. Above this shirorekha, different types of symbols which make each letter to pronounce differently. These are called 'Matrayen'. The possibility of recognising a specific character with traditional feedforward neural network is eventually low due to number of possible combinations formed with letters and Matrayen. The traditional recognition which usually recognises English handwriting can obtain upto certain level of efficiency such that overall word is recognised. But incase of Hindi, every letter depends Upon the previous letter and the matayen above the shirorekha. By using traditional recognising method on Hindi word it fails to recognise overall word effectively. Thus instead of Feedforward Neural Network we used Convolution and Revolution Hybrid Neural Network. The dataset used here for training is IIIT-HW-Dev dataset.

II. EXISTINGSYSTEM

II.aHow computer reads an Image?

In recognising an image consisting , basically there will be three channels one will be red, another will be green and finally we have blue channel this is popularly known as RGB. So each of these channels will have their own respective pixel value as referred to figure 2.

Manuscript published on 30 April 2019.
* Correspondence Author (s)
Nacode Vidheesh Kumar*, (Student, B.Tech), Computer science and engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, T.N., India
Aniket Yadav (Student, B.Tech), Computer science and engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, T.N., India
Rushikesh Deshmukh, (Student, B.Tech), Computer science and engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, T.N., India

While measuring size go an image it is read as AxBx3. It means the image has A rows B columns and 3 channels. For example for if an image is of size 28x28x3 then it means that image has 28 rows, 28 column and 3 channels. This is how the computer sees an image. This type of recognition is for coloured images. Incaseof Black and White images, it has only 2 channels. Let's see why can't we use fully connected networks for image classification. Consider an Image with 28x28x3 pixels, when we feed in this image to a fully connected network like figure 3 then the total number of ways require in the first hidden layer will be 2352. I.e., by multiplying the numbers. But in real life the images are not that small. Most of the images are of size over 200x200x3 pixels. Then when this image then feed it to the fully connected network, the number of bits required the first hidden layer itself will be 120000. We need to deal with such huge amount of parameters and obviously we require more number of neurones so that they can eventually lead to overfitting. So thats why we cannot use a fully connected network for image classification. Now let's see why we need convolutional neural network. In CNN a neuron in the layer will only be connected to a small region of the layer before it. Like one neuron will be connected to 3 neurons unlike in the fully connected network one neuron will be connected to all the neurons in the next layer. The advantage is that because of this we need to handle less amount of weights and in turn we need less number of neurons as well.

III. PROPOSEDSYSTEM

Let us understand what exactly Convolutional neural network and its background. They are the special type of feed forward artificial neural networks which is inspire from visual cortex. Visual cortex is nothing but a small region of brain which consists of a small regions of cells that are sensitive to specific region of visual field. Some individual neuronal Cells in the brain respond only in the presence of edges of a certain orientation. For example some neurons fire when exposed to vertical edges and some other neurons will fire up when exposed to horizontal edges. This is the motivation behind CNN.

HOW CNN WORKS

Generally, Convolutional Neural Network has 3 layers. Convolution, ReLU layer, Pooling and Fully connected Layer. Let's understand each layer one by one. Let's take an example of a classifier that can classify an image of an X and O. In empirical situation instead of X and 0, letters of hindi from the dataset are considered. With this example we'll be understanding all these four layers. As in the figure 1, X can be represented in the four forms as shown in the figure. These are nothing but the deformed images of X similarly for O's well. These deformed images need to be classified as either X or O.

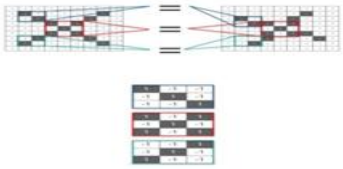

Figure 1. Selecting features or filters of image which are to be compared with input image

We know that computer understands an image using numbers at each pixels. In this case all the white spaces are assigned with -1 and all the black pixels are assigned as 1. When we use a normal techniques to compare these two images, one is a proper image of X and other is the deformed image of X we got to know that a computer is not able to classify the deformed image of X correctly. Because it is comparing it the proper image of X. When in these images, add the pixel values of both these images it outputs image which doesn't resembles X. Basically a computer is not able to recogniser whether it is an X or not. This traditional method is eliminated by using CNN. CNN compares the images by pieces. The pieces that it looks for are called features. By finding rough features matched, in roughly the same position in two images, CNN gets a lot better at seeing similarity than whole Image matching schemes. Features are small pieces of bigger image. We will be taking three features or filters as shown below in figure 4. We choose a feature and put it on the input image if it matches the images classified correctly.

Convolution layer:

The first layer is Convolution Layer. There are 4 steps of this particular layer.First we need to Line up the feature and the image. Secondly Multiply each image pixel by the corresponding feature pixel. Consider an example as shown in figure 2. Consider a feature as diagonal. This first diagonal feature that will take is placed on the image of X. Then we are going to multiply the corresponding pixel values such that the value in first row and first column of the feature will be multiplied with the value in first one first column of the image part. In this case 1 will be multiplied with 1 and -1 will be multiplied with -1 and so on. Then the corresponding multiplied values are placed in the new matrix of the same size as feature.

After the matrix is formed the The third step is to add all the values in Matrix and divided by divided by the number of pixels in the feature. In this case as showed in figure 2, the final value is 1 i.e., 9/9.

The next step is to create a map and put the value of filter(feature) at that particular place Similarly, these 4 steps are iterated by moving the filter through out the image.

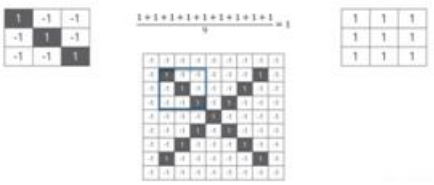

Figure 3. Output of convolutional layer for 3 filters for the input image

Figure 2 shows the result of the same feature moved to another location and performed filtering again. In this case, moved to centre of the image. The output is .55 which is placed in the matrix. Similarly, the features are moved to every position of the image and they are matched. Then the output matrix as shown in figure 3 is resulted. The same process of convolution is performed with every other filters by which 3 matrices are resulted(for 3 features.)

ReLU Layer:

In this layer we remove every negative values from the filtered image and replaces it zero's. This is done to avoid the values from summing upto zero. ReLU transform function only activates a node if the input is of above certain value. While the input is below then the output is zero.

Thus for each values obtained from previous convolution layer are passed to the ReLU layer. Then the negative values of each elements in the matrix are turned to 0.

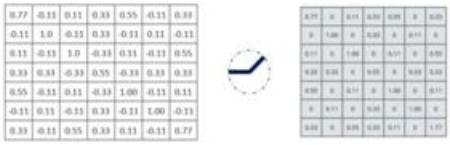

Figure 4. Placing the value 0 in place of the negative values

Pooling Layer:

In this layer we reduce the size of the image by selecting windows. For instance we set the window size of 2 then a 2x2 sized matrix is traced on the filtered image and the maximum value of the window is placed in the reduced matrix as shown in the figure 8. This process is continues till all the blocks in the reduced matrix are filled. The 7x7 filtered images shown in the previous figures are reduced to 4x4. This process is gone through by every other filtered image obtaining three 4x4 matrices. Once the pooling layer returns the output, that output Is considered as input in the convolution layer and again the whole process is repeated till the required size of the pooling layer output is obtained. In this case till 2x2 sized output is obtained.

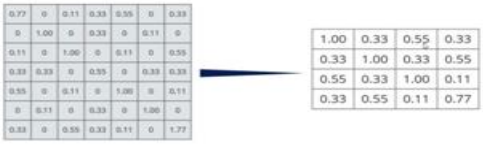

Figure 5. Reducing the 7x7 filter image to 4x4

Fully Connected Layer:

This is last layer where the classification is executed. Here all the filtered images are placed in form of singled column list as shown in figure 9.

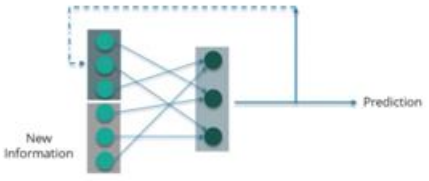

Figure 6. All layers in CNN stacked up. Repeating 3 layers of execution

When we look at our list formed, the image of X has some set of elements high and for image 0 has other set of elements as high i.e., 1. For the value of X 1,4,5,10 & 11 values of vector are high and for O 2,3,9,12. Basically then using this we can classify that for an input image. First compare the input vector with X. The values of the vector formed at 1,4,5,10 & 11 are added and divided by the sum of the values for the original vector X. Then the same is repeated with 0 vector. The maximum value between these two is selected and classified into their respective image i.e., X or 0. As shown in figure 10.

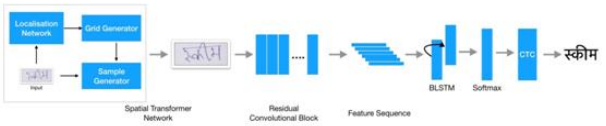

Figure 7. The filter is converted to single list and classified according to the high values in their respective list

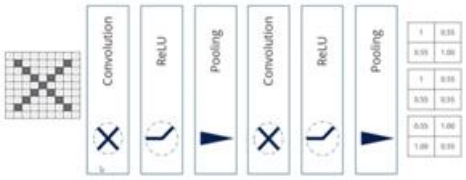

Figure 8. Final Classification

HOW RNN WORKS

RNN is a neural network is a type of back propagating neural network where its fed the output of previous step as input.

$h^{(t)} = g_h(w_i x^{(t)} + w_r h^{(t-1)} + b_h)$

$y^{(t)} = g_y(w_y h^{(t)} + b_y)$

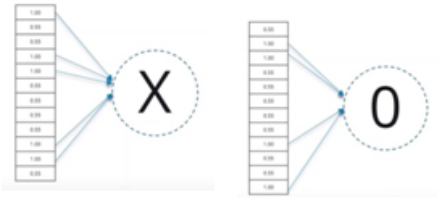

Figure 9. Output is taken as input of next neural network

The following diagram shows how this hybrid neural network is implemented practically.Figure 8 Shows the overall architecture of recognition system of the given input and results the identified output.

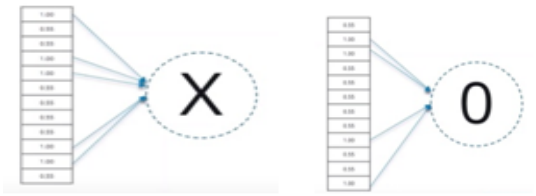

Figure 10. CNN RNN hybrid network visualisation for the given input sting

IV. SYSTEM ARCHITECTURE

As shown in the figure 8, the given input of hindi text goes through the CNN and RNN hybrid network. First step, the input gets processed in spatial transformer network and the new output is generated. The output acts as input to Residual Convolutional Block. Then the processing continues through Feature Sequence, BLSTM, Softmax, CTC. Then finally the recognized word is resulted with 87% Accuracy.

V.RESULT

The above mentioned proposed system is currently theoretical. The actual practical implementation may vary the accuracy after improvising.

VI. CONCLUSION

Thus we have used the dataset IIIT-HW-Dev as training set for the Hybrid Neural Network and recognised the Hand-written script. Additionally, for future aspects we plan to implement the system for other different languages which are based on Devanagiri Lipi.

REFERENCES

1. Anil.K.Jain and TorfinnTaxt, “Feature extraction methods for character Recognition-A Survey”, vol. 29, no. 4, pp. 641-662, 1996.
2. Dinesh, A. U. et al. (2007) “Isolated handwritten Kannada numeral recognition using structural feature and K-means cluster,” IISN, pp. 125-129.
3. Park, Jaehwa, VenuGovindaraju, and Sargur N. Srihari. "OCR in a hierarchical feature space." Pattern Analysis and Machine Intelligence, IEEE Transactions on 22.4 (2000): 400-407
4. Liu, Xia, and Zhixin Shi. "A format-driven handwritten word recognition system."2013 12th International Conference on Document Analysis and Recognition. Vol. 2. IEEE Computer Society, 2003.

AUTHORS PROFILE

NacodeVidheesh Kumar (Student, B.Tech), Computer science and engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, T.N., India

Aniket Yadav (Student, B.Tech), Computer science and engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, T.N., India

Rushikesh Deshmukh, (Student, B.Tech), Computer science and engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, T.N., India