Optical Character Recognition using Neural Networks by Classification based on Symmetry

A Mohammad Azam[1], Abhishek V Anil[2], Aishwarya G[3]

[1,2,3]Dept of CSE, BNM Institute of Technology, Bangalore

Abstract - *OCR allows us to extract data from a print medium by obtaining a scanned image of it, or a picture. We employ three main modules namely classification, line and character extraction and neural network. Every module contributes toward increasing the efficiency of the OCR. The classification based on symmetry property of characters reduces the training time and recognition time as well. The efficiency of the proposed system is compared to traditional methods of OCR. We employed multiple backpropagation neural networks to perform OCR. We trained the neural network with various images of three different fonts namely, Arial, Times New Roman, and Liberation Serif.*

***Key Words*: OCR, Artificial neural network, Character extraction, character classification.**

1. INTRODUCTION

Various papers explored in the field of OCR with neural networks have differences in the aspects of preprocessing methods, training data classification criteria, model of the neural network used and the training dataset. This variation reflects in the performance achieved by different authors, while some authors strive for a balance between training time and recognition time, others concentrate only on reducing the recognition time. The training data classification criteria include the symmetry property, the Euler number feature and curvature property of characters. Different models of neural networks range from simple multilayer perceptron neural network with backpropagation to a hybrid of convolutional and LSTM (Long Short Term Memory) neural network models.

Preprocessing methods such as binarization, character extraction to extract and process characters from sentences are used in [1]. Training phase uses a multi-layer perceptron neural network. The multi-layer perceptron neural network has an input layer, a hidden layer and an output layer.

[2] uses preprocessing techniques such as digitization, noise removal, binarization, line segmentation and character extraction. This network consists of 96 input neurons and 62 output neurons. The neural network was trained with 10 samples for each character.

[3] uses the curvature properties of the characters for differentiating between the characters, this is done by looking for black pixel from each corner of the extracted character, through certain angles called seeking angles, the difference between the seeking angles being 15°. A smaller seeking interval makes the input larger and makes the recognition process accurate but at the same time increase the calculations needed. Thus reducing the speed of the algorithm.

The method proposed in [4] reduces the training time considerably while increasing the recognition time. The author suggests classifying the training data according to the Euler number feature and a multi stage approach is used to deal with various types of inputs. The training time grows exponentially with respect to the size of training data. In the conventional neural network based character recognition system, the input nodes take the pixel values of the source image to process.

In [5] a hybrid convolutional-LSTM implementation is used. The author explores the performance differences between different combinations of geometric normalization, 1D LSTM, deep convolutional networks, and 2D LSTM networks. Result obtained is that deep hybrid neural networks using line normalization have the better performance among the combinations compared.

2. PROPOSED METHOD

In the proposed model for OCR, a neural network is being used. The neural network being used is a multi-layer perceptron network with backpropagation for learning. The input is the pixel data from the images. The three phases in the proposed model are classification phase, training phase and recognition phase.

2.1 Classification phase

The classification phase involves classifying the training data according to the symmetry property of the image. The four symmetry types are total symmetry, vertical symmetry, horizontal symmetry and no symmetry.

Vertical symmetry is checked by comparing the image of a character with a replica that is flipped along the vertical axis. Similarly, horizontal symmetry is checked by comparing the original image with a replica that is flipped along the horizontal axis.

Total symmetry occurs when an image is classified as both vertically and horizontally symmetrical. Whereas, no symmetry is when an image doesn't classify as either horizontally or vertically symmetrical.

The technique adopted for classifying the characters is flipping the matrix representation of the characters. They are flipped either horizontally or vertically. Based on the classification, the training data is stored separately.

Fig 1. shows the various steps in the algorithm for finding symmetry of images. The algorithm for symmetry checking was executed multiple times to tweak the threshold values that determined the symmetry of images.

In the algorithm it is assumed that the image is of size 28x28 pixels. The image is converted into a matrix of 1's and 0's. The matrix is then flipped either vertically or horizontally and the difference between the original and flipped matrix is computed. The absolute value of the result matrix is taken and the number of 1's are computed. The ratio of number of 1's and the size of the matrix is taken to check symmetry. It classifies as symmetry if the ratio is less than a certain threshold value that is obtained by repeatedly running the algorithm to achieve best classification.

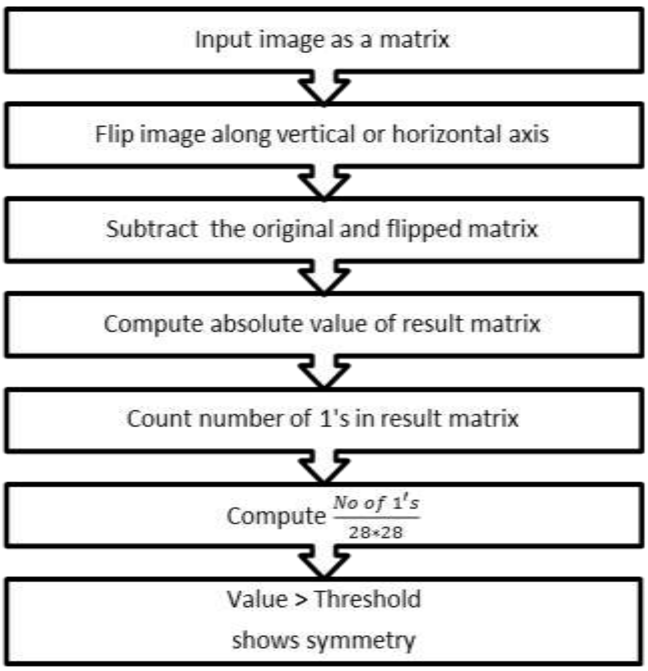

Fig -1: Steps in finding symmetry

2.2 Training phase

Training phase involves training different neural networks for these categorizations based on symmetry. Training is done by feeding the input to the neural network and then

calculating the cost function depending upon the output. Then the weights of the synapses are adjusted using a function like the gradient descent. These adjusted weights are then preserved by serializing the object of the network class.

2.3 Recognition phase

In the recognition phase, the input is the document which is read in either JPEG or PNG format. Pre-processing techniques used include line extraction, gray scale conversion, Gaussian adaptive thresholding and Character extraction.

Pre-processing techniques are used to provide cleaner input to the neural network. This helps in improving the recognition rate.

The aim of the character extraction module as shown in figure 2 is to first identify the text from an image. The module should be able to crop out individual characters which can be passed to the neural network. The image is sliced into different lines, by horizontally grouping the pixels. The characters are white pixels and the space between them is black. This is used to find the beginning and ending of a line, and character. Horizontal projection is used to find the spaces between lines.

Vertical projection is used to find the spaces between characters. The extraction module knows the end of a character when it finds that a certain amount of pixels are black in a row. The extracted characters have a specific border which is symmetric in all sides, to avoid changes or errors in the recognition phase.

Line and character extraction module also includes the space detection algorithm, the spaces are detected after detecting a character, the algorithm looks for spaces as a sequence of zeroes, with a particular threshold. The threshold size dictates the presence of a space or mere character spacing, the character spacing may be mistaken for spaces if the image is taken from a system the renders fonts differently. This can be overcome by dynamically adjusting the threshold in such a way that the algorithm can differentiate between character spaces and real spaces.

Limitations of character extraction lie with the fact that it cannot separate letters which are too close to each other, hence leading to a wrong recognition by the neural network.

Fig -2: Character extraction

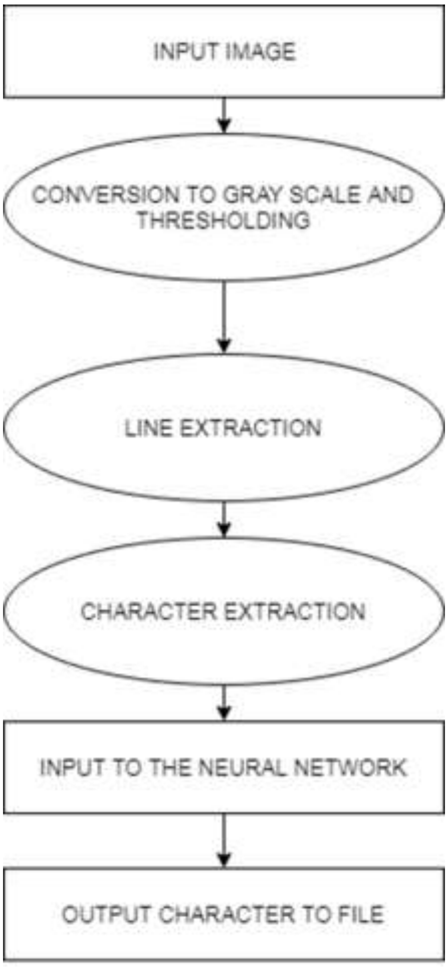

Fig -3: Steps in recognition phase

3. EXPERIMENTAL RESULTS

Table 1. shows the variation of accuracy of recognition with the introduction of punctuations in the input image. The proposed system is able to produce an accuracy of 99.2% in case of 'Arial' font without punctuations. This amount of accuracy cannot be achieved using the traditional approaches to OCR which compares the input images with a database containing the images of each character.

The 'Times New Roman' font without punctuation also produces a very good accuracy rate of 98.01% in recognizing the characters. In this case the accuracy decreases when punctuations are included. Our proposed system produces an accuracy rate of 99.2% in best case scenario with Arial font without punctuations whereas the system proposed in [6] produces at best 95.7%.

Table -1: Accuracy for fonts with punctuations and without punctuations

FONTS	WITH PUNCTUATION	WITHOUT PUNCTUATION
Times New Roman	92.56	**98.01**
Liberation Serif	70.27	71.03
Arial	90.04	**99.20**

However, there is no training time involved in traditional method. Our neural network requires 14.01 seconds to train all the four neural networks involved and to produce an output. Table 2. compares the traditional method used in [6] with our proposed model. It is observed that the recognition time of our model is almost 27 times lesser than the approach in [6].

Table – 2: Comparison with traditional method

APPROACHES	Top 1 accuracy rate (%)	Recognition time (msec/char)
Traditional method	95.7	1.45
Proposed method	99.2	0.055

4. CONCLUSIONS

We have proposed a method that uses multiple backpropagation neural networks to perform OCR. We train the neural network with various images of three different fonts namely, Arial, Times New Roman, and Liberation Serif. The training time is reduced drastically by classifying the training data using the symmetry property. The images are passed through a symmetry checking algorithm that returns details of whether the image has vertical symmetry, horizontal symmetry, total symmetry or no symmetry.

Finally, it is also observed that the inclusion of punctuations in the input image reduces the accuracy by a small margin. This happens because the extracted image of the punctuation marks are too small in size. When passed to the neural network, it fails to learn the features of this image.

REFERENCES

[1] M. Abdullah-al-mamun and T. Alam, "An approach to empirical Optical Character Recognition paradigm using Multi-Layer Perceptorn Neural Network," 2015 18th International Conference on Computer and Information Technology (ICCIT), Dhaka, 2015, pp. 132-137

[2] S. Afroge, B. Ahmed and F. Mahmud, "Optical character recognition using back propagation neural network,"

2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE), Rajshahi, 2016, pp. 1-4

[3] M. M. Farhad, S. M. N. Hossain, A. S. Khan and A. Islam, "An efficient Optical Character Recognition algorithm using artificial neural network by curvature properties of characters," 2014 International Conference on Informatics, Electronics & Vision (ICIEV), Dhaka, 2014, pp. 1-5.

[4] H. Y. Lin and C. Y. Hsu, "Optical character recognition with fast training neural network," 2016 IEEE International Conference on Industrial Technology (ICIT), Taipei, 2016, pp. 1458-1461.

[5] T. M. Breuel, "High Performance Text Recognition Using a Hybrid Convolutional-LSTM Implementation," 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 2017, pp. 11-16.

[6] Venu Govindaraju, and Sargur N. Srihari, “OCR in a Hierarchical Feature Space” IEEE transactions on pattern analysis and machine intelligence, vol. 22, no. 4, April 2000.