Journal of Engineering and Applied Sciences 12 (Special Issue 12): 9359-9364, 2017
ISSN: 1816-949X

Feature Extraction Techniques for Handwritten Character Recognition using Neural Networks with Non-Uniform Background

G. Roshan, M. Akhil Kumar, R.S.V. Vishnu and G. Virajit
Department of Electronic and Communication Engineering, Amrita School of Engineering,
Amrita University, Amrita Vishwa Vidyapeetham, 690525 Amritapuri, India

Abstract: Handwritten character recognition has been one of the most active and challenging areas of research in the field of image processing and pattern recognition. This study is a brief survey and comparison of two different feature extraction methods used in recognition of uppercase English alphabets in a given handwritten scanned text from sentences. These techniques were tested using feed-forward neural networks which are trained using back propagation algorithm with the cost function as Mean Square Error (MSE). The resesrch also provides substantial evidence that enhancement of recognition and reduction of misclassification is dependent on the type of features used. We extended our researcch towards extraction of characters from colored and non-uniform backgrounds using Maximally Stable External Regions (MSER) feature algorithm.

Key words: Handwriting recognition, neural network, MSERF, feature extraction, binarization, algorithm, scanned

INTRODUCTION

In the present age of automation, pattern recognition, deep learning and machine learning algorithms are playing key roles is structuring the rapid digitalization of the world. Character recognition has been one of the most compelling and demanding fields which contributes immensely to automation process and improves the interface between man and machine in various applications. Neural networks are important information-processing paradigm inspired by human brain itself. They are composed of simple elements which operate in parallel similar to the biological nervous system and their function is determined by the connections between these elements. We can train a neural network to perform a function by adjusting the values of the connection links also called as weights (Richard *et al*., 1973). The neural network links are manipulated until particular input results in a known correct output. The optimum weights can be obtained by adopting various algorithms (Sharma and Chaudhary, 2013). Feature extraction is an important step in pattern recognition engines, there is extensive research being carried out in this field to determine which features can provide optimum classification. In such context, we are comparing the results of two such feature extraction methods. Characters can be from various backgrounds such as walls, textured papers and textured charts etcetera. With different backgrounds the extraction of characters from the image becomes difficult, thus we have extended our work to create a system that can extract characters from the non-uniform background using MSERF algorithm (Neumann and Matas, 2012).

Neural network and pre-processing of image

The structure of neural network: The neural network architecture is comprised of the input layer, hidden layers, and an output layer. Each layer is comprised of neurons. Number of input layer neuron nodes depends on number of features we are going to use to classify the objects, number of output neuron nodes depends on number of classes, therefore for classification of digits output layer will require 10 neuron nodes with each node representing each digit, similarly classification of alphabets will require 26 output neuron nodes with each node representing each English alphabet. There are 2 different neurons which can be used.

Perceptron: This neuron can be perceived as a logic gate. They only provide a scalar output of 1 or 0. All the inputs to the neurons are coupled with the respective weights and then their sum is calculated and compared with a threshold. If the result of the sum is greater that the threshold it provides an output of 1 else 0.

Sigmoid neurons: Unlike perceptron's these can respond to small changes in input and also reflect them in the output as the output is an exponential function of the

Corresponding Author: G. Roshan, Department of Electronic and Communication Engineering, Amrita School of Engineering, Amrita University, Amrita Vishwa Vidyapeetham, 690525 Amritapuri, India

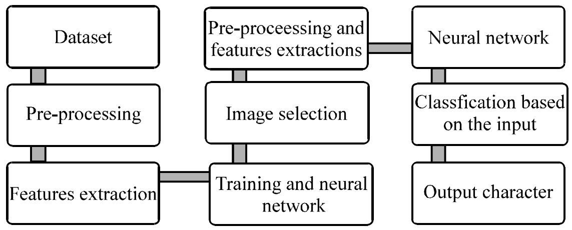

Fig. 1: Steps for character recognition

input. As we are dealing with pixel content in the image using this type of neuron is useful as we can track small changes in the structure of the character (Fig. 1).

Steps for character recognition

Pre-processing of the image

Background noise removal: In real time scenarios there is a higher probability of noise present in images. Most of the time it will be electronic noise, these noises will add spurious information to the image (Rani *et al*., 2016). So, it has to be eliminated, there are many digital filters to help us achieve this, we used the property of median filter to eliminate the spurious elements in the image such as salt and pepper noise or unnecessary blobs.

Binarization/thresholding: This process of converting a pixel image to a binary image is necessary because it reduces the computation burden as we deal with pixel values of only 1 or 0's. At first, the colored image is converted into gray-scale. Then a threshold is applied, this threshold can either be set fixed or adaptive using a clustering algorithm.

Morphological processing of image: There is a possibility that we may lose some of the vital pixel content during the thresholding process which might be useful for recognizing the character to overcome this liability we apply morphological techniques to enhance the image. We use morphological opening operation on an image (Gonzalez and Wintz, 1977) where the image is first eroded and then dilated using structuring elements. This process removes all the spurious elements and joins any broken gaps (Bruhwiler *et al*., 2007) in the character skeleton. We also apply region thinning operation to extract the character skeleton of one-pixel width.

The universe of discourse: It may be defined as the shortest matrix that can enclose the entire region of interest. This provides us a matrix which only extends as far as the character skeleton is spread as shown in Fig. 2.

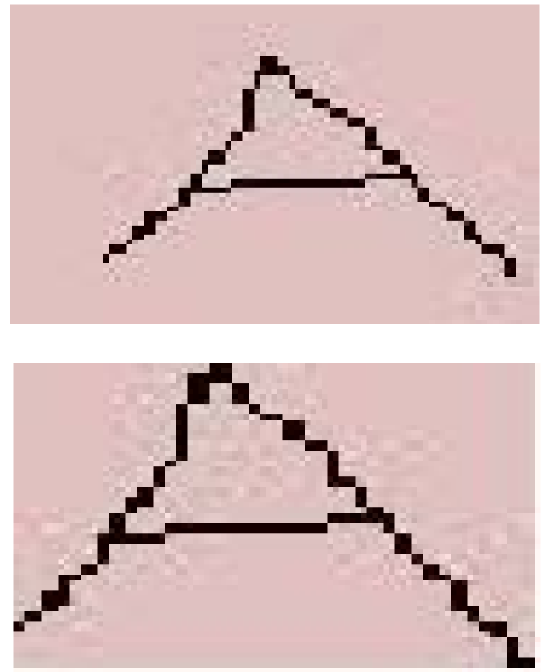

Fig. 2: Universe of discourse

This process is important because we do not want our classification to be affected by the position of the character in the image (Dileep, 2012).

Zoning: The full image is divided into zones for additional accuracy. This provides uniqueness for each character. The type of lines and the features which were dependent on the complete character skeleton are now dependent on the part of the character skeleton contained in a particular zone, thus increasing the ease for differentiation between characters and thus increasing accuracy in classification. Here, we have used a 3×3 zoning technique (Fig. 3).

Line segmentation: We know that character recognition solely developed for individual characters is not applicable as we have characters as part of words and sentences. Thus our system must be able to extract characters from an image containing multiple sentences. In order to accommodate this functionality we apply the universe of discourse to the image containing multiple sentences and then we parse through all the rows to check if there exists a row with no pixel content of interest, the first row to exist as the above criteria will represent the break in the sentence, we save the pixel content till this row as a new line and delete it from the original image and this iteration is repeated again to extract the second line and is continued until all the sentences are processed.

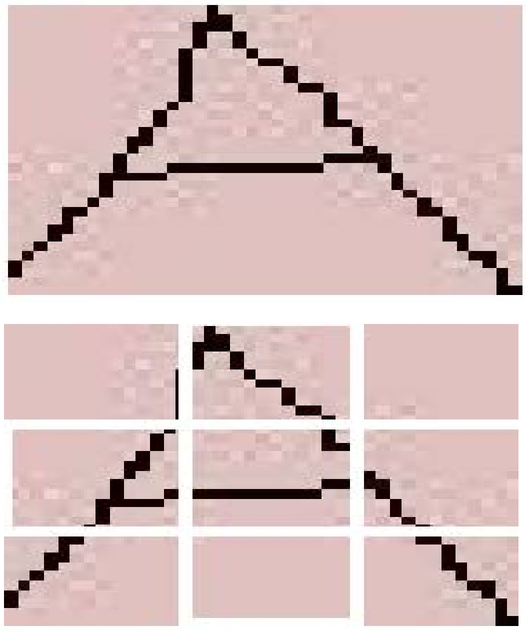

Fig. 3: Zoning

MATERIALS AND METHODS

Feature extraction techniques and classification: There are several feature extraction and classification techniques that can be adopted for recognition of handwritten characters. We will discuss and compare two methods to provide evidence that the type of features directly impacts the accuracy and efficiency of the classification.

Method 1: In this method, we apply all the image pre-processing techniques except for universe of discourse to the image. In this method, we resize the input image to the size of 28×28 before implementing the pre-processing techniques. Then we convert the zoned image into an input array of size (784×1) and provide this as the input vector to the neural network for classification. Here, we can observe that we feed the pixel values of the pre-processed image as inputs to the neural network and we train the neural networks using stochastic gradient algorithm against targeted outputs. Thus in this method, the background is represented as zero pixel values and the character skeleton is represented with a high-intensity value of 1 (binary), this forces the neural network to adjust weights at the input nodes at which the character skeleton is projected. When the neural network is tested with an image of the character similar to that of the database, we obtain an output at the recognized character node in the output layer. Though, this method seems to be effective it has more limitations which will be discussed in the next section.

Method 2: A character skeleton is made of various line segments we can extract properties of such line segments such as a number of lines and length of these line segments that can be used as input features for the neural net for classification (Blumenstein *et al*., 2003). This process involves extracting of various line segments that build up the character skeleton and to determine the type of lines if they are horizontal, vertical, right diagonal or left diagonal lines and also to determine the length of the line segments. A detailed explanation of this technique is provided by Blumenstein *et al*. (2003). From the above-stated feature extraction method, we get the following features as input to the neural network.

Features considered per zone are total number and lengths of horizontal, vertical, right diagonal and left diagonal and area of character skeleton. The above features are extracted per zone. To find the line segments and length of the lines and type of the line segment found we use the algorithm explained clearly by Dileep (2012). This algorithm makes use of three important properties to extract line segments, namely starters, intersections and minor starters and we use a reference direction matrix to differentiate the type of the extracted lines.

The below-mentioned features are extracted from the entire character skeleton are appended to the feature vector extracted from all the zones.

Features applied on the total skeleton:

- Eccentricity: the eccentricity is the ratio of the distance between the foci of the ellipse and its major axis length. In this algorithm, it is the value of eccentricity of the ellipse which has the same second-moments as the region
- Euler number: it is the difference between the number of connected components and the holes in the image
- Orientation: it is the angle between the major and minor axis of the ellipse which has the same second-moments as the region
- Extent (similar to the area of the region)

Thus, per character skeleton, we will have 9 feature points from 9 zones which give us a total of 81 feature points. And there are 4 features extracted from the skeleton of the image which is appended to the 81 zonal features, thus giving us a total of 85 feature points per character. After we extract the total features we normalize them into the fractional value of the range -1 to +1. This provides us to reduce the deviation errors due to differentiation in handwriting.

Comparison b/w methods: We now discuss the merits and limitations as well as a comparison of the results of the two methods stated in the previous section.

Method 1

Merits: This method has less computational burden during the pre-processing of the image. As the pixel content is directly fed to input neurons the accuracy rate in classification is better than the latter method provided that the font and thickness of the character skeleton are similar to those in the dataset.

Limitations: It fails to recognize and classify characters which have a bold font. As the pixel content is directly fed as inputs it becomes mandatory to resize the test image to 28×28 size, thus, resulting in loss of important data which will be further damaged due to thresholding. It has a high computational burden as the number of input nodes is more, thus increasing the number of weights to be processed during training and simulation of the neural network. The classification becomes erroneous if the position of the character in the image is different from that of the database images. This method cannot be extended to classify alphabets and digits simultaneously as it will result in classification between digit "0" and alphabet "O", similarly the algorithm might fail to recognize the difference between digit "8" and alphabet "B".

Method 2

Merits: This method has a less computational burden in terms of simulating the neural network. This method is independent of the size of the image and the font and boldness of the character. It is independent of the position of the character and the inclination of the sentence. Since, the features are independent of the pixel content and only dependent on the character skeleton the range of correct classification was increased and the error rate was observed to decrease. The optimum weights of the neural links were obtained at a faster rate and with fewer iterations.

Limitations: It fails to recognize and classify characters which have similar skeleton structure and small variations in such as uppercase I and J, X and Y etcetera. It increases the preprocessing burden for extracting features from the character. Similar to the method I this also cannot classify difference between digit "0" and alphabet "O" and digit "8" and alphabet "B".

RESULTS AND DISCUSSION

The graph in Fig. 4a depicts the performance plot of the training algorithm for a neural network in method 1 as we can observe the error at the last epoch or iteration,

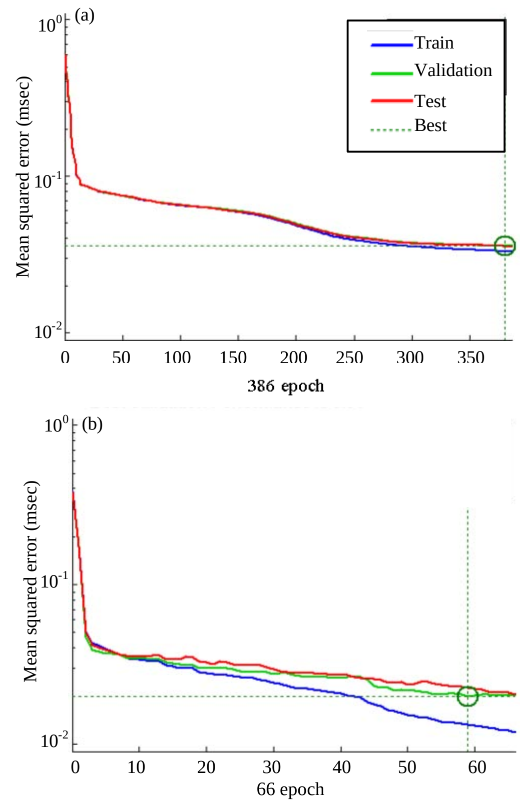

Fig. 4: a) Performance plot of method 1, best validation performance is 0.035824 at epoch 380 and b) Performance plot of method 2, best validation performance is 0.019774 at epoch 59

which is 386th iteration is 0.035824 which is approximately double of the error obtained when method 2 is implemented. The graph in fig depicts the performance plot of the training algorithm for a neural network in method 1 as we can observe the error at the last epoch or iteration, that is 59th iteration is 0.019774 which is approximately half of the error obtained when Method 1 is implemented.

Test case: The test case covers all the possibilities for the validation of the system. As we can observe the similarities between letter O and D have taken precedence than the differences, thus leading to recognition of handwritten letter O as D (Table 1).

Extraction from non-uniform backgrounds: Handwritten characters can be available from different backgrounds. In such a scenario we have a problem to extract the region of interest and remove those undesired objects in the background. For solving this problem we are using a computer vision algorithm called Maximally Stable External Regions Features (MSERF) which help in blob detection in images for matching correspondence in two images with varied view points. The MSERF detector

Table 1: Comparison of the two methods

	Feature extraction technique	
Font size	Method 1	Method 2
Number of epochs for optimum weights	386	59
Mean square error at the last iteration	0.035824	0.019774
Accuracy of recognition of characters similar to properties in dataset	88.05%	89%
Accuracy of recognition with differences in Bold and font	48%	79.96%
Time elapsed to train the neural network (approx.)	2 min 30 sec	1 min
Number of hidden layers nodes used	35	39
Number of input nodes	784	85
Training algorithm	Stochastic gradient	Stochastic gradient

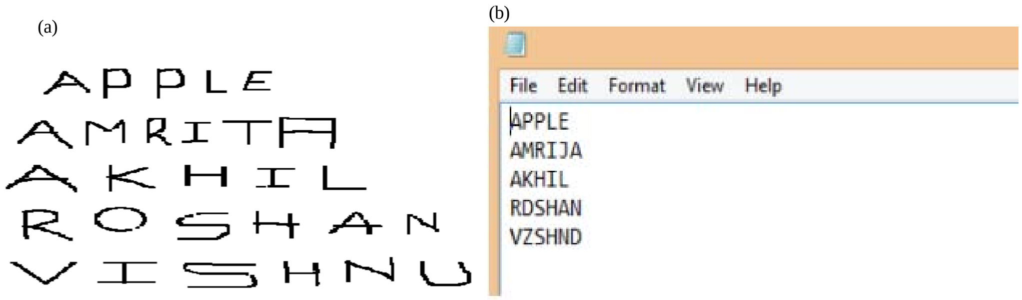

Fig. 5: a) Input and b) output in text document

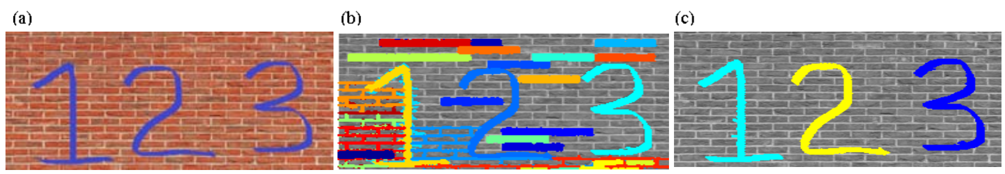

Fig. 6: a) Input image b) Post MSERF algorithm and c) Output

incrementally steps through the intensity range of the input image to detect stable regions. The threshold delta parameter in MSERF function in MATLAB determines the number of increments the detector tests for stability. The MSERF object checks the variation of the region area size between different intensity thresholds and this variation can be put as a threshold parameter, so that when the variation is less than the threshold region is considered stable. Using this algorithm stereo matching and object detection algorithms have improved. We can also use MSERF for text recognition when combined with canny edges algorithm (Chen *et al*., 2011). As MSERF gives us the connected regions any undesired edges formed from canny are removed, this also improves the blurry text caused by MSERF when canny edges are added to it.

We can observe in Fig. 6a, numerical characters have a bricked background and using MSERF algorithm we are able to find most text part in the image but also can find some stable connected components, now our goal is to remove undesired regions. Here, we can use geometric properties of text to remove undesired regions (Gonzalez *et al*., 2012) the properties used are.

- Eccentricity: eccentricity for general characters well be approximately >0.99
- Euler number: euler should be <4
- Extent: returns a scalar that specifies the ratio of pixels in the region to pixels in the total bounding box. Computed as the area divided by the area of the bounding box and extent can be a value in range of 0.2- 0.9
- Solidity: returns a scalar specifying the proportion of the pixels in the convex hull that is also in the region. computed as area/convex Area and this value can be <0.3

So, by setting the above threshold to properties, we can eliminate non-text regions. And this can be still improved by adding stroke width parameter for thresholding.

CONCLUSION

The accuracy of the classification of characters and the rate of recognition is directly dependent on the type of features upon which the classification is made. This study provides an additional reference for selection of the relevant feature extraction technique which plays an important role in the performance of the system. The method 1 can be used when we are sure that our dataset covers all the conditions in which our test cases can lead to correct classification. In the case of uncertainty in the thickness of the character skeletons and if the position of characters is random then it is advantageous to use the method 2 for feature extraction. We can use the MSERF algorithm discussed above to extract the characters or digits from the non-uniform background. This review establishes a complete system that is capable of converting a scanned document containing multiple lines of handwritten characters to digitized text documents. We hope that this material will serve as a guide and update for readers working in the area of hand-written character recognition.

REFERENCES

Blumenstein, M., B. Verma and H. Basli, 2003. A novel feature extraction technique for the recognition of segmented handwritten characters. Proceedings of the 7th International Conference on Document Analysis and Recognition, August 6, 2003, IEEE, Edinburgh, UK., ISBN:0-7695-1960-1, pp: 137-141.

Bruhwiler, E., A. Miyamoto and M.A. Konno, 2007. Automatic crack recognition system for concrete structures using image processing approach. Asia J. Inform. Technol., 6: 553-561.

Chen, H., S.S. Tsai, G. Schroth, D.M. Chen and R. Grzeszczuk *et al.*, 2011. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. Proceedings of the 2011 18th IEEE International Conference on Image Processing, September 11-14, 2011, IEEE, Brussels, Belgium, ISBN:978-1-4577-1304-0, pp: 2609-2612.

Dileep, D., 2012. A feature extraction technique based on character geometry for character recognition. MSc Thesis, Department of Electronics and Communication Engineering, Amrita School of Engineering, Kollam, India.

Gonzalez, A., L.M. Bergasa, J.J. Yebes and S. Bronte, 2012. Text location in complex images. Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), November 11-15, 2012, IEEE, Tsukuba, Japan, ISBN:978-1-4673-2216-4, pp: 617-620.

Gonzalez, R. and P. Wintz, 1977. Digital Image Processing. Addison-Wesley Publishing Co., Logman, ISBN: 0471018880, pp: 750.

Neumann, L. and J. Matas, 2012. Real-time scene text localization and recognition. Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, June 16-21, 2012, IEEE, Providence, Rhode Island, USA., ISBN:978-1-4673-1226-4, pp: 3538-3545.

Rani, D.A.N.S., P. Vineeth and D. Ajith, 2016. Detection and removal of graphical components in pre-printed documents. Intl. J. Appl. Eng. Res., 11: 4849-4856.

Richard, D.O., H.E. Peter and S.G. David, 1973. Pattern Classification. Vol. 2, Wiley, New York, USA.,.

Sharma, A. and D.R. Chaudhary, 2013. Character recognition using neural network. Intl. J. Eng. Trends Technol., 4: 662-667.

Sorace, R.E., V.S. Reinhardt and S.A. Vaughn, 1997. High-speed digital-to-RF converter. Patent and Trademark Office, Washington, DC. USA. https://www.google. com/patents/US5668842