

Character Recognition Technique using Neural Network

Harshal Bobade¹, Amit Sahu²

¹M.E (Scholar),

G.H.Raisoni College of Engineering and Management, Amravati.

²M.E (CSE),

G.H.Raisoni College of Engineering and Management, Amravati.

Abstract

Character Recognition (CR) has been extensively studied in the last half century and progressed to a level, sufficient to produce technology driven applications. The pre-processing of characters comprises bounding of characters for translation invariance and normalization of characters for size invariance. Now, the rapidly growing computational power enables the implementation of the present CR methodologies and also creates an increasing demand on many emerging application domains, which require more advanced methodologies. In this paper an attempt is made to develop neural network strategies for the isolated. Handwritten English characters (A to Z, a to z). The pre-processing of characters comprises bounding of characters for translation invariance and normalization of characters for size invariance. First, an overview of CR systems and their evolution over time is presented. Then, the available CR techniques with their superiorities and weaknesses are reviewed. Finally, the current status of CR is discussed and directions for future research are suggested. The variability in a character introduced by the rotation and deformation is the main concern of this paper. This variability has been taken into account by devising a neural logic based approach using normalized angle features. Now, the rapidly growing computational power enables the implementation of the present CR methodologies and also creates an increasing demand on many emerging application domains, which require more advanced methodologies. This material serves as a guide and update for the readers, working in the Character Recognition area, remove noises and feature extraction. Special attention is given to the off-line handwriting recognition, since this area requires more research to reach the ultimate goal of machine simulation of human reading.

Keyword- CR; neural network; normalization; character; off-line; Character Recognition; feature extraction.

1.Introduction

The desire to develop algorithms to match human performance for handwritten characters

recognition has led to intense research in this field during the last decades. There has been a particular interest in the last decade in the recognition of hand written characters both isolated and cursive. attempted to recognize a small vocabulary of key words on a word-level basis description rather than a letter-level basis. the chain based off-line cursive word recognition. the use of dynamic programming for matching between a word and a pre-stored model of a word. machine simulation of human functions has been a very challenging research field since the advent of digital computers. In some areas, which require certain amount of intelligence, such as number crunching or chess playing, tremendous improvements are achieved. On the other hand, humans still outperform even the most powerful computers in the relatively routine functions such as vision. Machine simulation of human reading is one of these areas, which has been the subject of intensive research for the last three decades, yet it is still far from the final frontier. In this overview, Character Recognition (CR) is used as an umbrella term, which covers all types of machine recognition of characters in various application domains. The overview serves as an update for the state of the art in the CR field, emphasizing the methodologies required for the increasing needs in newly emerging areas, such as development of electronic libraries, multimedia databases and systems which require handwriting data entry. The study investigates the direction of the CR research, analyzing the limitations of methodologies for the systems, which can be classified based upon two major criteria: the data acquisition process (on-line or off-line) and the text type (machine-printed or hand- written). No matter which class the problem belongs, in general there are five major stages in the CR problem:

1. Pre-processing,
2. Segmentation,
3. Representation,
4. Training and recognition,
5. Post processing.

The organization of the paper is as follows: Section 2 gives the details of preprocessing of characters which includes normalization, thinning of isolated handwritten English characters. Section 3 contains the representation and recognition techniques employed. Section 4 outlines the results of implementation of the proposed approach followed

by Section 5 which summarizes the results and provides suggestions for future work. The paper is arranged to review the CR methodologies with respect to the stages of the CR systems, rather than surveying the complete solutions. Although the off-line and on-line character recognition techniques have different approaches, they share a lot of common problems and solutions. Since it is relatively more complex and requires more research compared to on-line and machine-printed recognition, off-line handwritten character recognition is selected as a focus of attention in this article.

2. Literature survey

Character recognition task has been attempted through many different approaches like template matching, statistical techniques like NN, HMM, Quadratic Discriminant function (QDF) etc. Template matching works effectively for recognition of standard fonts, but gives poor performance with handwritten characters and when the size of dataset grows. It is not an effective technique if there is font discrepancy [4]. HMM models achieved great success in the field of speech recognition in past decades, however developing a 2-D HMM model for character recognition is found difficult and complex [5]. NN is found very computationally expensive in recognition purpose [6]. N. Araki et al. [7] applied Bayesian filters based on Bayes Theorem for handwritten character recognition. Later, discriminative classifiers such as Artificial Neural Network (ANN) and Support Vector Machine (SVM) grabbed a lot of attention. In [3] G. Vamvakas et al. compared the performance of three classifiers: Naive Bayes, K-NN and SVM and attained best performance with SVM. However SVM suffers from limitation of selection of kernel. ANNs can adapt to changes in the data and learn the characteristics of input signal [8]. Also, ANNs consume less storage and computation than SVMs [9]. Mostly used classifiers based on ANN are MLP and RBFN. B.K. Verma [10] presented a system for HCR using MLP and RBFN networks in the task of handwritten Hindi character recognition. The error back propagation algorithm was used to train the MLP networks. J. Sutha et al. in [11] showed the effectiveness of MLP for Tamil HCR using the Fourier descriptor features. R. Gheroie et al. in [12] proposed handwritten Farsi character recognition using MLP trained with error back propagation algorithm. Computer Science & Information Technology (CS & IT) 27 similar shaped characters are difficult to differentiate because of very minor variations in their structures. In [13] T. Wakabayashi et al. proposed an F-Ratio (Fisher Ratio) based feature extraction method to improve results of similar shaped characters. They considered pairs of similar shaped characters of different scripts like English, Arabic/Persian,

Devnagri, etc. and used QDF for recognition purpose. QDF suffers from limitation of minimum required size of dataset. F. Yang et al. in [14] proposed a method that combines both structural and statistical features of characters for similar handwritten Chinese character recognition. As it can be seen that various feature extraction methods and classifiers have been used for character recognition by researchers that are suitable for their work, we propose a novel feature set that is expected to perform well for this application. In this paper, the features are extracted on the basis of character geometry, which are then fed to each of the selected ML algorithms for recognition of SSHMC.

3. Methodology for feature extraction

A device is to be designed and trained to recognize the 26 letters of the alphabet. We assume that some imaging system digitizes each letter centered in the system's field of vision. The result is that each letter is represented as a 5 by 7 grid of real values. The following figure shows the "perfect" pictures of all 26 letters.



Figure 1: The 26 letters of the alphabet with a resolution of 5×7 .

However, the imaging system is not perfect and the letters may suffer from noise:



Figure 2: A "perfect" picture of the letter "A" and 4 noisy versions (standard deviation of 0.2).

Perfect classification of ideal input vectors is required, and more important reasonably accurate classification of noisy vectors. Before OCR can be used, the source material must be scanned using an optical scanner (and sometimes a specialized circuit board in the PC) to read in the page as a bitmap (a pattern of dots). Software to recognize the images is also required. The character recognition software then processes these scans to differentiate between images and text and determine what letters are represented in the light and dark areas. Older OCR systems match these images against stored bitmaps based on specific fonts. The hit-or-miss results of such pattern-recognition systems helped establish OCR's reputation for inaccuracy. Today's OCR engines add the multiple algorithms of neural

network technology to analyze the stroke edge, the line of discontinuity between the text characters, and the background. Allowing for irregularities of printed ink on paper, each algorithm averages the light and dark along the side of a stroke, matches it to known characters and makes a best guess as to which character it is. The OCR software then averages or polls the results from all the algorithms to obtain a single reading. OCR software can recognize a wide variety of fonts, but handwriting and script fonts that mimic handwriting are still problematic, therefore additional help of neural network power is required. Developers are taking different approaches to improve script and handwriting recognition. As mentioned above, one possible approach of handwriting recognition is with the use of neural networks. Neural networks can be used, if we have a suitable dataset for training and learning purposes. Datasets are one of the most important things when constructing new neural network. Without proper dataset, training will be useless. There is also a saying about pre-processing and training of data and neural network: "Rubbish-in, rubbish-out". So how do we produce (get) a proper dataset? First we have to scan the image. After the image is scanned, we define processing algorithm, which will extract important attributes from the image and map them into a database or better to say dataset. Extracted attributes will have numerical values and will be usually stored in arrays. With these values, neural network can be trained and we can get a good end results. The problem of well defined datasets lies also in carefully chosen algorithm attributes. Attributes are important and can have a crucial impact on end results. The most important attributes for handwriting algorithms are:

1. Negative image of the figure, where the input is defined as 0 or 1. 0 is black, 1 is white, values in between shows the intensity of the relevant pixel.
2. The horizontal position, counting pixels from the left edge of the image, of the center of the smallest rectangular box that can be drawn with all "on" pixels inside the box.
3. The vertical position, counting pixels from the bottom, of the above box.
4. The width, in pixels, of the box.
5. The height, in pixels, of the box.
6. The total number of "on" pixels in the character image.
7. The mean horizontal position of all "on" pixels relative to the center of the box and divided by the width of the box. This feature has a negative value if the image is "leftheavy" as would be the case for the letter L.
8. The mean vertical position of all "on" pixels relative to the center of the box and divided by the height of the box.
9. The mean squared value of the horizontal pixel distances as measured in 6 above. This attribute will

have a higher value for images whose pixels are more widely separated in the horizontal direction as would be the case for the letters W or M.

10. The mean squared value of the vertical pixel distances as measured in 7 above.
11. The mean product of the horizontal and vertical distances for each "on" pixel as measured in 6 and 7 above. This attribute has a positive value for diagonal lines that run from bottom left to top right and negative value for diagonal lines from top left to bottom right.
12. The mean value of the squared horizontal distance times the vertical distance for each "on" pixel. This measures the correlation of the horizontal variance with the vertical position.
13. The mean value of the squared vertical distance times the horizontal distance for each "on" pixel. This measures the correlation of the vertical variance with the horizontal position.
14. The mean number of edges (an "on" pixel immediately to the right of either an "off pixel or the image boundary) encountered when making systematic scans from left to right at all vertical positions within the box. This measure distinguishes between letters like "W" or "M" and letters like "I" or "L."
15. The sum of the vertical positions of edges encountered as measured in 13 above. This feature will give a higher value if there are more edges at the top of the box, as in the letter "Y."
16. The mean number of edges (an "on" pixel immediately above either an "off pixel or the image boundary) encountered when making systematic scans of the image from bottom to top over all horizontal positions within the box.
17. The sum of horizontal positions of edges encountered as measured in 15 above.

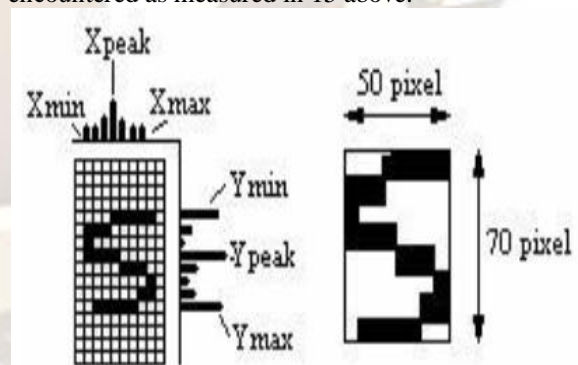


figure 3: Example of image defined attributes

4. Feature Extraction

The sub-images have to be cropped sharp to the border of the character in order to standardize the sub-images. The image standardization is done by finding the maximum row and column with 1s and with the peak point, increase and decrease the counter until meeting the white space, or the line

with all 0s. This technique is shown in figure below where a character "S" is being cropped and resized.

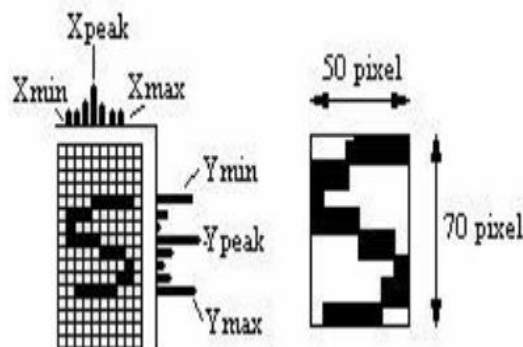


Figure 4: Cropped and resized picture

The image pre-processing is then followed by the image resize again to meet the network input requirement, 5 by 7 matrices, where the value of 1 will be assign to all pixel where all 10 by 10 box are filled with 1s, as shown below:

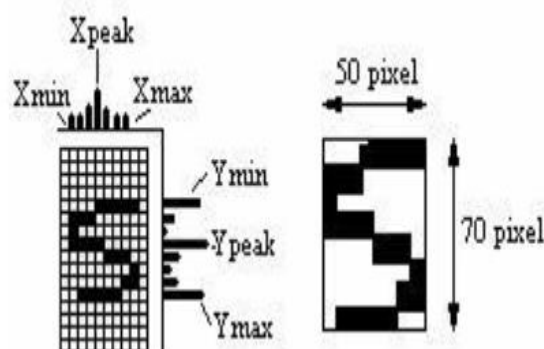


Figure 5: Image resize again to meet the network input requirement

Finally, the 5 by 7 matrices is concatenated into a stream so that it can be feed into network 35 input neurons. The input of the network is actually the negative image of the figure, where the input range is 0 to 1, with 0 equal to black and 1 indicate white, while the value in between show the intensity of the relevant pixel. By this, we are able to extract the character and pass to another stage for future "classification" or "training" purpose of the neural network.

5. Using a Neural Network to solve the problem

The script appcr1.m which is part of the Neural Network Toolbox demonstrates how character recognition can be done with a acknowledgment propagation network. The twenty-six 35-element input vectors are defined in the function as a matrix of input vectors called alphabet. The target vectors are also defined in this file with a variable called targets. Each target vector is a 26-

element vector with a 1 in the position of the letter it represents, and 0's everywhere else. For example, the letter "C" is to be represented by a 1 in the third element (as "C" is the third letter of the alphabet), and 0's everywhere else. The network receives the 5x7 real values as a 35-element input vector. It is then required to identify the letter by responding with a 26-element output vector. The 26 elements of the output vector each represent a letter. To operate correctly, the network should respond with a 1 in the position of the letter being presented to the network. All other values in the output vector should be 0. In addition, the network should be able to handle noise. In practice, the network does not receive a perfect letter (see Fig.1) as input. Specifically, the network should make as few mistakes as possible when classifying vectors with noise of mean 0 and standard deviation of 0.2 or less (see Fig.2).

6. Estimating the System Performance

The reliability of the neural network pattern recognition system is measured by testing the network with hundreds of input vectors with varying quantities of noise. For example we create a noisy version (SD 0.2) of the letter "J" and query the network:

```
noisyJ = alphabet(:,10)+randn(35,1) * 0.2;
plotchar(noisyJ);
A2 = sim(net,noisyJ);
A2 = compet(A2);
answer = find(compet(A2) == 1);
plotchar(alphabet(:,answer));
```

Here is the noisy letter and the letter the network picked (correctly).



Figure 6: The network is tested with a noisy version of the letter "J".

The script file appcr1 tests the network at various noise levels, and then graphs the percentage of network errors versus noise. Noise with a mean of 0 and a standard deviation from 0 to 0.5 is added to input vectors. At each noise level, 100 presentations of different noisy versions of each letter are made and the network's output is calculated. The output is then passed through the competitive transfer function so that only one of the 26 outputs (representing the letters of the alphabet), has a value of 1. The number of erroneous classifications is then added and percentages are obtained.

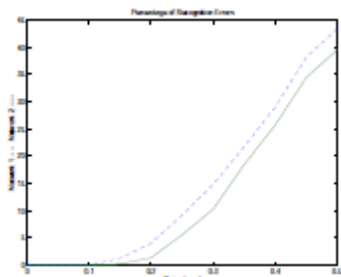


Figure 7: Performance for the network trained with and without noise.

The solid line on the graph shows the reliability for the network trained with and without noise. The reliability of the same network when it had only been trained without noise is shown with a dashed line. Thus, training the network on noisy input vectors greatly reduces its errors when it has to classify noisy vectors. The network did not make any errors for vectors with noise of std 0.00 or 0.05. When noise of std 0.2 was added to the vectors both networks began making errors. If a higher accuracy is needed, the network can be trained for a longer time or retrained with more neurons in its hidden layer. Also, the resolution of the input vectors can be increased to a 10-by-14 grid. Finally, the network could be trained on input vectors with greater amounts of noise if greater reliability were needed for higher levels of noise.

7. CONCLUSION

Feature extraction is the most crucial & important part of handwritten character recognition. With feature extraction we also implemented steps of pre-processing to normalize an image of a character. We selected Back propagation neural network for classification purpose. Comparative analysis of different feature extraction methods in terms of accuracy is done in our work. Cross-corner, diagonal, direction methods are most accurate methods according to the confusion matrices & 13-region have least recognition rate. Combining different feature vector into a single feature vector for hybrid method & result shows that hybrid method have higher recognition rate compared to its individual feature extraction method in case of accuracy. In the future work we are going to introduce a hybrid approach for the feature extraction techniques with using the pros and cons of different feature extraction technique.

References

[1] R.Tokas, A.Bhadu, "A Comparative Analysis Of Feature Extraction Techniques For Handwritten Character Recognition", International Journal of Advanced Technology & Engineering Research (IJATER), July 2012, pp. 215-218.

[2] M. Zafar, D. Mohamad, M.M. Anwar, "Recognition of Online Isolated Handwritten Characters by Back propagation Neural Nets Using Sub-Character Primitive Features", IEEE Multitopic Conference (INMIC), 2006, pp. 157 – 162

[3] G. Vamvakas, B. Gatos, S. Petridis, N. Stamatopoulos, "An Efficient Feature Extraction and Dimensionality Reduction Scheme for Isolated Greek Handwritten Character Recognition", IEEE Ninth International Conference on Document analysis and Recognition(ICDAR), 2007, vol. 2, pp. 1073 – 1077

[4] J.R. Prasad, U.V. Kulkarni, R.S. Prasad, "Offline handwritten character recognition of Gujrati script using pattern matching", IEEE 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication, 2009, pp. 611-615.


[5] H.S. Park, S.W. Lee, "An HMMRF-Based Statistical Approach for Off-line Handwritten Character Recognition", IEEE Proceedings of the 13th International Conference on Pattern Recognition, 1996, vol. 2, pp. 320 – 324.

[6] C.L. Liu, H. Sako, H. Fujisawa, "Performance evaluation of pattern classifiers for handwritten character recognition", International Journal on Document Analysis and Recognition (IJ DAR), 2002, vol. 4, pp. 191–204.

[7] N. Araki, M. Okuzaki, Y. Konishi , H. Ishigaki , "A Statistical Approach for Handwritten Character Recognition Using Bayesian Filter", IEEE 3rd International Conference on Innovative Computing Information and Control, 2008, pp. 194 – 194.

[8] N. Arica, F.T. Yarman-Vural, "An Overview Of Character Recognition Focused On Off-line Handwriting", IEEE Transactions on Systems, Man, and Cybernetics, 2001, vol. 31, pp. 216 –233.

[9] F. Kahraman, A. Capar, A. Ayvaci, H. Demirel, M. Gokmen, "Comparison of SVM and ANN performance for handwritten character classification", Proceedings of the IEEE 12th Signal Processing and Communications Applications Conference, 2004, pp. 615 – 618.

- 
- [10] B.K. Verma, "Handwritten Hindi Character recognition Using Multilayer Perceptron and Radial Basis Function Neural Networks," Proceedings of IEEE International conference on Neural Networks, 1995, vol. 4, pp. 2111-2115.
- [11] J. Sutha, N. Ramaraj, "Neural Network Based Offline Tamil Handwritten Character Recognition System", IEEE International Conference on Computational Intelligence and Multimedia Applications, 2007, vol.2, pp. 446 – 450.
- [12] R. Gharoie, M. Farajpoor, "Handwritten Farsi Character Recognition using Artificial Neural Network", International Journal of Computer Science and Information Security, 2009, vol. 4.
- [13] T. Wakabayashi, U. Pal, F. Kimura, Y. Miyake, "Fratio Based Weighted Feature Extraction for Similar Shape Character Recognition" IEEE 10th International Conference on Document Analysis and Recognition (ICDAR), 2009, pp. 196-200.
- [14] F. Yang, X.D. Tian, X. Zhang, X.B. Jia, "An Improved Method for Similar Handwritten Chinese Character Recognition", IEEE Third International Symposium on Intelligent Information Technology and Security Informatics (IITSI), 2010, pp. 419 – 422.
- [15] N.J. Nilsson, "Introduction to Machine learning", An early draft of a proposed textbook, Robotics Lab, Deptt of Computer Science, Stanford University, Stanford, CA 94305, 1998.
- [16] WEKA 3: Data Mining With Open Source Machine Learning Software in JAVA URL: <http://www.cs.waikato.ac.nz/ml/weka/>.
- [16] N. Friedman, D. Geiger, M. Goldszmidt, "Bayesian Network Classifiers, Machine learning, 1997, pp.131-163.