ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

An Idea to Recognition of handwritten Characters using Genetic Algorithms

Samta Jain Goyal[1], Rajeev Goyal[2]

[1,2]**Amity University, Madhya Pradesh**

Abstract: Challenges in handwritten characters recognition is due to the variation and distortion of handwritten characters, since different people use different style and way of draw the same shape of the characters. This paper demonstrates the nature of handwritten characters, conversion of handwritten data into electronic data, and the neural network approach to make machine capable of recognizing hand written characters. This motivates the use of Genetic Algorithms for the problem. In order to prove this, we made a pool of images of characters. We converted them to graphs. The graph of every character was intermixed to generate new or unique styles intermediate between the styles of parent character. Character recognition involved the matching of the graph generated from the unknown character image with the graphs generated by mixing.

Key Words: Machine recognition, Handwriting recognition, neural networks, generic algorithms, graph theory, and coordinate geometry, offline handwriting recognition

I. Introduction:

Handwritten recognition has always been a special problem, especially in an offline mode. Handwritten recognition is a famous problem which involves the recognition of whatever input is given in the form of image, scanned paper, text, etc. There is lot of work has been done in this area in the past few years. In the other ways, handwritten recognition refers to the identification of the written characters. Here the recognition of character is of online character recognition as well as offline character recognition.

In online character recognition, utilize the digitizers which directly capture writing with the order of speed pen up, pen down and the stroke information. Online works on real time data only which is not much challenging rather than offline character recognition.

But in offline character recognition the final figure is given to recognize which is a challenging task in the research area. The complexity of the recognition is usually associated with the size of the language being considered also the various characters are written and the differences between the various characters at the same time. If the language contains more number of characters, the identification of those would be much more difficult than the language which contains less number of characters. In offline character recognition system, character image is converted into bit pattern by optically digitizing devices such as camera or scanner. Here the bit pattern data is used for recognizing offline characters.

This paper is used to present offline character recognition with the help of neural network. Since neural network techniques are insensitive to the missing data, to recognize those characters we will use four stages for the same - 1. Scanning 2. Pre-processing 3. Feature extraction 4. Recognition

In Scanning, a recognition system acquires a scanned image as an input image through a digital scanner or any other suitable digital input device. This image should have a specific format such as

JPEG, BMP etc. [17].In Preprocessing stage, the output of the image acquisition is fed as input. The raw data of handwritten characters will be subjected to a number of preprocessing steps to make it useable for the next steps. The preprocessing phase aims to extract the relevant features or parts and prepares them for next phases. Pre-processing of the image means applying a number of procedures for thresholding, smoothing, filtering, resizing, and normalizing so that successive algorithm to final classification can be made simple and more accurate [18].The pre-processing is a series of operations performed on the scanned input image. It essentially enhances the image rendering it suitable for segmentation. Binarization process converts a gray scale image into a binary image using global thresholding technique. Detection of edges in the binarized image using sobel technique, dilation the image and filling the holes present in it are the operations performed in the last two stages to produce the pre-processed image suitable for segmentation [21].

In Segmentation phase, an image is present in the form of sequence of characters which is decomposed into sub-images of individual character. In this system, labeling process is used for segmentation of pre-processed input image into isolated characters by assigning a number to each character. This labeling provides information about number of characters in the image [9].Since each character has differential features, which is very important for pattern recognition. So Feature Extraction describes the relevant information for the shape of character, contained in a pattern so that the task of classifying the pattern is made easy by a formal procedure. The main goal of feature extraction is to obtain the most relevant information from the original data and represent that information in a lower dimensionality space.

A term feature extraction is termed that transforms the input data into the set of features. The features set will be used to extract the relevant information from the input data in order to perform the desired task using this reduced representation .Feature extraction method used for the extraction [10, 15].The widely used feature extraction schemes are Template matching, Gradient feature extraction method, Directional feature extraction, Fourier descriptor, Zoning, Diagonal Based Feature Extraction.

Here we will use Genetic Algorithm to recognise offline hand written characters which gives better results as compared to other techniques. These results are based on recognition accuracy, training time and classification time.

II. Genetic Algorithms

Genetic Algorithms are very good means of optimization to recognize offline characters. They optimize the desired property by generating hybrid solutions. These hybrid solutions are added to the solution pool which is used to generate more hybrids. These solutions generally give better results than the already generated solutions. This is all done by the genetic operators. Her we will use set of graphs generated from training data for any character. Genetic Algorithms are used to mix two such graphs and generate new graphs. This newly generated graph may happen to match the character better than the existing graph. So we can say that Genetic Algorithms are good means of optimization.

III.Terms and work related with Genetic Algorithms

a. Fitness function
b. Crossover function

III-A.In Genetic Algorithm, fitness function is used to test the “goodness of the solution”. The fitness function tests the level of goodness of any of the solution from the solution pool. For our problem we have used fitness function to measure the deviation of the graph of the solution for the unknown input. If the two graphs are very similar, the deviation would be low and hence the value of fitness function would be automatically low. The lower the value of the fitness function, the better would be matching. Hence the graph with the lowest value of fitness function would be the most probable answer. We first derive a formula to find the deviation between any two edges of a graph which would be then used as means of finding the deviation between two graphs.

Deviation of a graph: To found deviation of a graph easily, first to know the deviation of an edge with another edge of a graph. By pairing up of edges and iterating through all the edges, we can found this deviation. The design of algorithm is as follows-

Deviation (G1, G2)

```
Step1: Here we initialize dev ← 0

Step2: Taking two graphs G1 & G2.While G1, G2 has no edges

Step3: Find the edges e1 from G1 and e2 from G2 such that deviation between e1 and e2 is the Minimum for any pair of edges e1 and e2

Step4: Add its deviation to dev

Step5: Remove e1 from G1 and e2 from G2

Step6: For all edges e1 left in G1.

Step7: Add deviation of e1 and null to dev

Step8: For all edges e2 left in G2

Step9: Add deviation of null and e2 to dev

Step10: Return dev
```

Here we are trying to reduce total deviation. For this we select the pair of edges, one from each graph such that their deviation is minimal. We keep such pairs till one graph gets empty. In the end we add deviation of all the left edges or curves to find the minimum deviation between the two graphs.

III-B.Crossover: In genetic algorithms, crossover is a genetic operator used to mix two solutions to form a new solution. This may result better than the existing ones. The crossover operation also helps in optimization. The basic motive of using this operation is to mix styles. If the two graphs are in different styles, we would be able to mix them and form a new one that is intermediate between the existing ones. Many solutions are possible for every combination of parents. In this algorithm, we try to generate all the forms and add it as a solution. Hence we can get many solutions .The following is the algorithm used for the crossover of the two graphs

CrossOver (G1, G2)

```
Step1: match ← MatchPoints (G1, G2) (Selects corresponding points from G1 & G2)

Step2: W1 ← GenerateAdjacency(G1) (Creates adjacency matrix from graph G1)

Step3: W1 ← GenerateAdjacency(G2) (Creates adjacency matrix from graph G2)
Step4: if W1 ≠ W2 (Compares adjacency matrices generated by above two steps)

Step5: FindPaths(W1) (Finds all paths between all pairs of nodes in W1)

Step6: FindPaths(W2) (Finds all paths between all pairs of nodes in W1)

Step7: p1 ← path between vertices v1 and v2 in G1 of length l, for all v1, v2 in W1 and length l= 1 or 2 or 3 or 4

Step8: p2 ← path between vertices match[v1] and match[v2] in G2 of length l2=1 or 2 or 3 or 4

Step9: If p1 and p2 results true and if p1 ≠ p2

Step10: Remove all common edges of p1 and W1 from W1

Step11: All the edges of p2 are added to W1

Step12: g ← MakeGraph(W1,W2)(Creates graph from Adjacency matrix of W1)

Step13: g is added to solution set.

Step14: else g ← MakeGraph (W1, W2)

Step15: Add g to solution set
```

IV. Conclusion

In this paper we proposed the use of genetic algorithm and graph theory for solving the problem of offline handwriting recognition. We had given the input in the form of images. The idea of designed algorithm was based on the training data that was initially present in the database. The training data consisted of at least two training data sets per character in the language. In our idea, we used the graph theory and coordinate geometry to convert the images to graphs. We saw that these conversions changed the whole problem of handwriting recognition to the problem of graph matching. When a pure graph matching was done, sufficiently fine results were obtained. The algorithm could recognize unknown characters given as input. But the efficiency improved drastically when we applied genetic algorithms. This algorithm helped in both style optimization and distance optimization. In style optimization, it helped us to mix two different styles to generate a new one that was in between the two. We also saw how the algorithm helped in distance optimization. It transformed the start and end point of vertices in such a way that it could match better with the unknown data input.

This was done by taking mean coordinates of the vertices of parents.

REFERENCES

[1] J.Pradeep, E.Srinivasanand, S.Himavathi, "Diagonal based feature extraction for handwritten alphabets recognition system using neural network," International Journal of Science & Information Technology (IJCSIT), vol. 3, no. 1, pp. 27-37, Feb 2011.

[2] Nafiz Arica and Fatos T. Yarman-Vural, "An overview of character recognition focused on off-line handwriting," IEEE transactions on systems, man, and cybernetics-part c: applications and reviews, vol. 31, no. 2, pp. 216-233, May 2001.

[3] M. D. Garris, R. A. Wilkinson, and C. L. Wilson, "Methods for enhancing neural network handwritten character recognition," International Joint Conference on Neural Networks, IEEE, Seattle, vol. 1, pp. 695-699, July 1991.

[4] R. Plamondon and S.N. Srihari, "On-line and off-line handwriting recognition: A comprehensive survey," IEEE transactions Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 63-84, Jan. 2000.

[5] Lera, G. and Pinzolas, M., "Neighborhood based Levenberg-Marquardt algorithm for neural network training," IEEE Transaction on Neural Networks, vol. 13, no.5, pp. 1200-1203, Sept 2002.

[6] Satish Kumar and Chandan Singh, "A study of zernike moments and its use in devnagari handwritten character recognition," International Conference on Cognition and Recognition, pp. 514-520, 2005.

[7] Cheng-Lin Liu, "Normalization-cooperated gradient feature extraction for handwritten character recognition," IEEE Transaction on pattern analysis and machine intelligence, vol. 29, no. 8, Aug. 2007.

[8] Ján Dolinský and Hideyuki Takagi, "Analysis and modelling of naturalness in handwritten characters," IEEE transactions on neural networks, vol. 20, no. 10, pp. 1540-51, October 2009.

[9] E.Lecolinet and R.G. Casey, "A survey of methods and strategies in character segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no.7, pp. 690-706, July 1996.

[10] Cheng-Lin Liu, "Normalization-cooperated gradient feature extraction for handwritten character recognition," IEEE Transaction on pattern analysis and machine intelligence, vol. 29, no. 8, Aug. 2007.

[11] Anita Pal and Dayashankar Singh, "Handwritten english character recognition using neural network," International Journal of Computer Science & Communication vol. 1, no. 2, pp. 141-144, July-December 2010.

[12] Chang-Lin Liu, "Normalization-cooperated Gradient Feature Extraction for Handwritten character recognition." IEEE transactions on Pattern Analysis and machine intelligence, vol. -29, Issue8, pp. 1465-1469, Aug 2007.

[13] Walid A. Salameh and Mohammed A. Otair, "Online handwritten character recognition using an optical back propagation neural network," Issues in Informing Science and Information Technology, vol. 3, pp. 787-795, 2005.

[14] Maier, H. R. and Dandy, G. C., "The effect of internal parameters and geometry on the performance of back propagation neural networks: an empirical study," Environmental Modelling and Software, Vol. 13, no. 2: pp. 193-209, 1998.

[15] Anil K. Jain and Torfinn Taxt, "Feature extraction methods for character recognition-A survey," Pattern Recognition, vol. 29, no. 4, pp. 641-662, April 1996.

[16] James A. Freeman, David M. Skapura, "Back-propagation algorithm," in Neural Networks: Algorithms, Applications, and Programming Techniques (Computation and Neural System), Addison-Wesley Pub (Sd), 1991.

[17] Dougherty, E.R., "Random processes for image and signal processing," IEEE Press, New York, 2000.

[18] Azizah Suliman, Mohd. Nasir Sulaiman, Mohamed Othman, Rahmita Wirza, "Chain coding and pre processing stages of handwritten character image file," Electronic Journal of Computer Science and Information Technology (eJCSIT), vol. 2, no. 1, pp. 6-13, 2010.

[19] Dayashankar Singh, Maitree Dutta and Sanajay Kumar Singh "Handwritten character recognition using twelve directional feature input and Neural Network." International journal of computer applications. (0975-8887) volNo.3, 2010.

[20] Shuchita Upadhyaya et. al, "Handwritten Character Recognition Using Diagonal Based Feature Extraction and Neural Network", National Conference on Advanced Computing Technology, vol-No. 2, pp. 615-618, March 2013.

[21]Rafael C.Gozalez,Richard E. woods and Steven L.Eddins,Digital Image Processing using MATLAB, Pearson Education, Dorling Kindersley, SouthAsia,2004