Volume 4, Issue 11, November 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Efficient Handwritten Alphabet Recognition Using LBP based Feature Extraction and Nearest Neighbor Classifier

Amrita Hirwani, Neelmani Verma, Sandeep Gonnade
Dept. of Comp. Science, MATS School of Engg & IT MATS University,
Raipur (C.G.), India

Abstract- Handwritten Alphabet Recognition is one of the artificial intelligence applications which provide an important fundamental for various advanced applications, such as information retrieval and human-computer interaction applications. An off-line handwritten alphabetical character recognition system using local binary pattern (LBP) as feature descriptor and nearest neighbor classifier is described in the project work. In conventional hand written alphabet recognition technique mostly vertical and horizontal or diagonal blocks average value is used as a feature value of alphabets. Most of the time these techniques failed to prove itself as an efficient feature and hence results poor alphabet recognition efficiency.

This work brought forward a robust feature descriptor for alphabets using local binary pattern. At the classifier side a nearest neighbor classifier is proposed. Nearest neighbor classifiers are a class of non-parametric methods used in statistical classification (or pattern recognition). This method classifies objects based on closest training examples in the feature space.

A set of seventy alphabets including capital letters, small letters and numbers are used for testing of the proposed method. The proposed recognition system performs efficiently yielding higher levels of recognition accuracy compared to the systems employing the conventional feature extraction methods.

Keywords — Handwritten alphabet recognition, local binary pattern (LBP), feature Descriptor, nearest neighbor classifier.

I. INTRODUCTION

Handwriting recognition has been one of the most attractive and demanding research areas in field of image processing and pattern recognition in the recent years. It contributes massively to the advancement of automation process and can improve the interface between man and machine in numerous applications. Most of the research works have been focusing on new techniques and methods that would reduce the processing time and provide higher recognition accuracy [1], [2].

The first important step in any handwritten recognition system is pre-processing followed by segmentation and feature extraction. Pre-processing comprises with the steps that are required to shape the input image into a form suitable for segmentation. In segmentation step, the input image is segmented into individual characters and then, each character is resized into m x n pixels towards training the network [3].

The selection of appropriate feature extraction method is probably the single most important factor in achieving high recognition performance. Several methods of feature extraction for character recognition have been reported in the literature [4]. The widely used feature extraction methods are Template matching, Deformable templates, Unitary Image transforms, Graph description, Projection Histograms, Contour profiles, Zoning, Zernike Moments, Spline curve approximation, Fourier descriptors, Gradient feature and Gabor features.

In this project work, a vertical and horizontal blocking based LBP feature extraction scheme for the off-line handwritten alphabet recognition system is proposed. In the feature extraction process, resized individual character of size 70x50 pixels is further divided into 35 equal zones, each of size 10x10 pixels. The features are extracted from the pixels of each zone by moving along vertical and horizontal directions and determining average of LBP values for each 10x10 block. This procedure is repeated for all the zones leading to extraction of 35 features for each alphabet. A set of seventy alphabets including capital letters, small letters and numbers (between 0 to 9) are used for testing of the proposed method. The proposed recognition system performs efficiently yielding higher levels of recognition accuracy compared to the systems employing the conventional feature extraction methods.

Gita Sinha, Anita Rani, Prof. Renu Dhir, Mrs. Rajneesh Rani proposed Zone based approach which is the combination of image centroid zone and zone centroid zone of numeral/character image. In image centroid zone character is divided into n equal zone and then image centroid and the average distance from character centroid to each zones/grid/boxes present in image is calculated. Similarly, in zone centroid zone character image is divided into n equal zones and centroid of each zones/boxes/grid and average distance from zone centroid to each pixel present in block/zone/grid is calculated. SVM used for subsequent classifier and recognition purpose. This method obtained 95.11% recognition accuracy [5].

Juarez Paulino da Silva J´unior, Marcus Vinicius Lamar, Jacir Luiz Bordim addresses the problem of recognizing the American Sign Language (ASL) hand alphabet relying only on depth information acquired from an RGB-D sensor. They proposed a novel Iterative Closest Point (ICP) based recognition methodology that comprehensively analyzes the inputs and outputs of the alignment as efficiency and accuracy determinants. Next, a novel classification technique, denoted Approximated KB-fit, is proposed to efficiently handle the space complexity of the database template matching [6].

Padmavathi S, Saipreethy.M.S, Valliammai.V, proposes a method to convert the Indian Sign Language (ISL) hand gestures into appropriate text message. In this the hand gestures corresponding to ISL English alphabets are captured through a webcam. In the captured frames the hand is segmented and the neural network is used to recognize the alphabet. The features such as angle made between fingers, number of fingers that are fully opened, fully closed or semi closed and identification of each finger are used as input to the neural network. Experimentation done for single hand alphabets [7].

Binod Kumar Prasad, Goutam Sanyal**,** proposed a recognition model based on Hidden Markov Models (HMMs) supported by novel feature extraction technique. Post-processing is done to enhance the recognition rate further. A data-base of 13000 samples is collected from 100 writers written five times for each character. 2600 samples have been used to train HMM and the rest are used to test recognition model. This recognition system achieved a good average recognition rate of about 93.24 percent [8].

After the complete study of the literatures available on alphabet recognition systems following problems was identified: First, lack of robust and efficient features of the alphabets which can define uniquely an alphabet. Second, an efficient feature classifier for recognition of alphabets which can provide higher recognition efficiency.

II. RECOGNITION SYSTEM

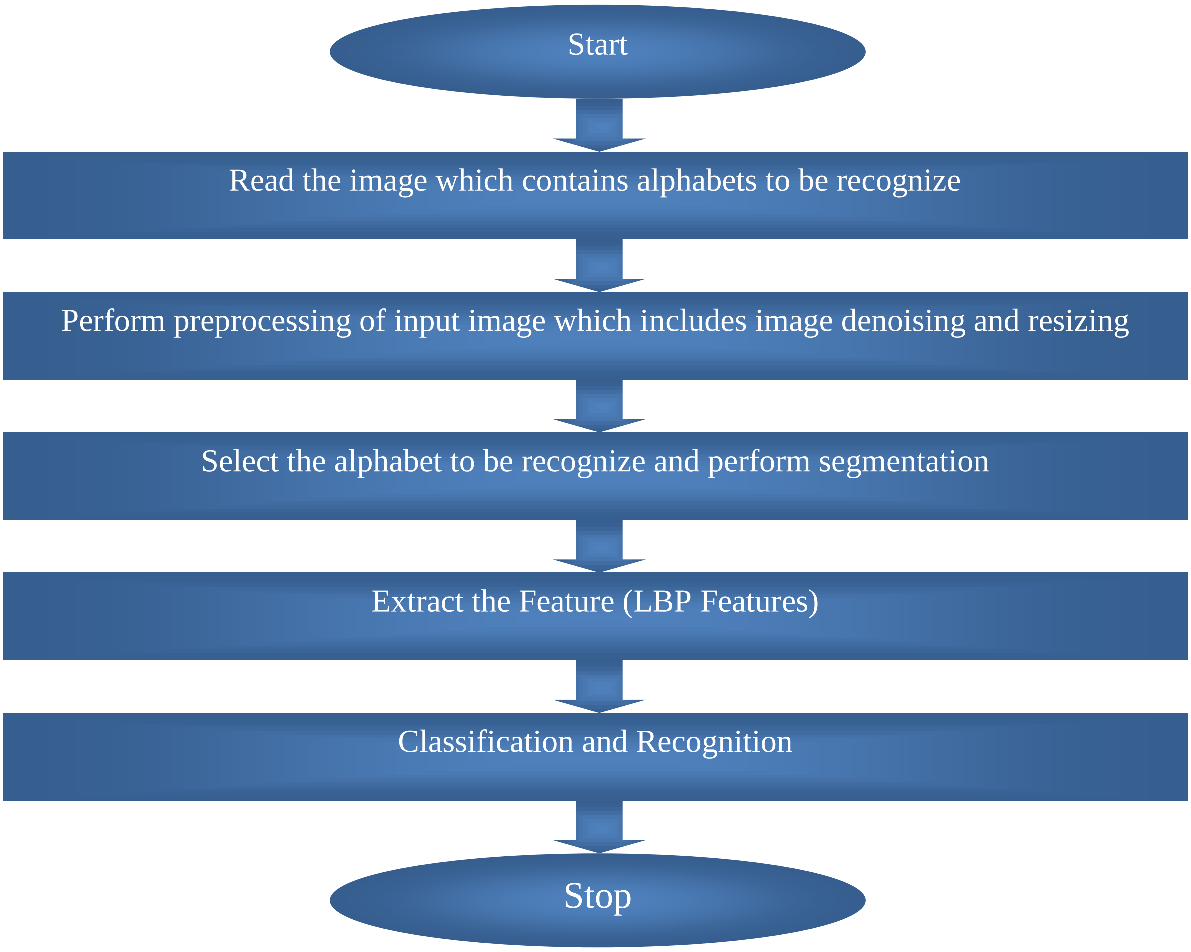

Figure-1 Schematic diagram of the proposed system

In this section, the proposed alphabet recognition system is described in brief. A typical handwriting recognition system consists of pre-processing, segmentation, feature extraction, classification and recognition, and post processing stages. The schematic diagram of the proposed recognition system is shown in Figure 1.

III. FEATURE EXTRACTION

LBP based vertical and horizontal feature extraction scheme for recognizing off-line handwritten characters is proposed in this work. Every character image of size 70x 50 pixels is divided into 35 equal zones, each of size 10x10 pixels (Fig. 2(c)). The features are extracted from each zone pixels by moving along the vertical and horizontal directions of its respective 10x10 pixels. Each zone has 5 horizontal and 7 vertical lines.

LBP is determined for each 10x10 block and averaged to form a single feature value and placed in the corresponding zone (Fig. 2 (b)). This procedure is sequentially repeated for the all the zones. Finally, 35 features are extracted for each character.

In addition, 7 and 5 features are obtained by averaging the values placed in zones row wise and column wise, respectively. As result, every character is represented by 47, that is, 35 +12 features.

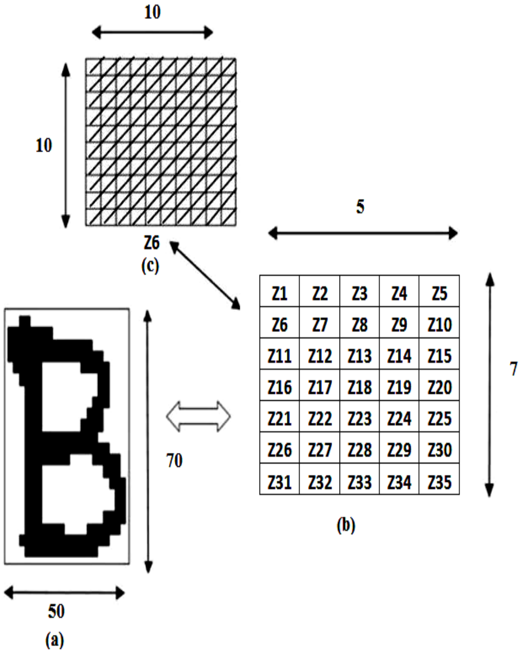

Figure 2 Feature Extraction process

A. Local Binary Patterns (LBP)

LBP is a type of feature used for classification in computer vision. LBP define the particular case of the Texture Spectrum model and proposed in 1990. In 1994, LBP was first described. It is a powerful feature for texture classification.

The LBP feature vector, in its simplest form, is created in the following manner:

- Divide the examined window into cells (e.g. 4x4 pixels for each cell).
- For each pixel in a cell, compare the pixel to each of its 2 neighbors (on its left-top, left-middle, left-bottom, right-top, etc.). Follow the pixels along a circle, i.e. clockwise or counter-clockwise.
- Where the center pixel's value is greater than the neighbor's value, write "1". Otherwise, write "0". This gives an 8-digit binary number (which is usually converted to decimal for convenience).
- Compute the histogram, over the cell, of the frequency of each "number" occurring (i.e., each combination of which pixels are smaller and which are greater than the center).
- Optionally normalize the histogram.
- Concatenate (normalized) histograms of all cells. This gives the feature vector for the window.

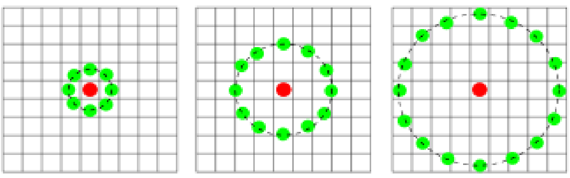

Figure 3 three neighborhood examples used to define a texture and calculate a local binary pattern (LBP)

IV. CLASSIFICATION AND RECOGNITION

For classification of alphabets this project work utilizes modern highly efficient K-nearest neighbor classifier. Following subsection briefly describes basics of K-nearest neighbor classification technique.

A. K-Nearest Neighbor Classifier

In pattern recognition, the K-Nearest Neighbors algorithm (or k-NN for short) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression:

- In k-NN classification, the output is a class membership. A majority vote of an object's neighbor, is used to classify an object and with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.

- In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors.

K-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.

Both for classification and regression, it can be useful to weight the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor.

The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, but explicit training step is not required.

B. Algorithm of K-Nearest Neighbor Classifier

The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples.

In the classification phase, k is a user-defined constant, and an unlabeled vector (a query or test point) is classified by assigning the label which is most frequent among the k training samples nearest to that query point.

A commonly used distance metric for continuous variables is Euclidean distance. For discrete variables, such as for text classification, another metric can be used, such as the overlap metric (or Hamming distance). Often, the classification accuracy of k-NN can be improved significantly if the distance metric is learned with specialized algorithms such as Large Margin Nearest Neighbor or Neighborhood components analysis.

A drawback of the basic "majority voting" classification occurs when the class distribution is skewed. That is, examples of a more frequent class tend to dominate the prediction of the new example, because they tend to be common among the k nearest neighbors due to their large number. To overcome this problem weight the classification by taking into account the distance from the test point to each of its k nearest neighbors. The class (or value, in regression problems) of each of the k nearest points is multiplied by a weight proportional to the inverse of the distance from that point to the test point. Abstraction in data representation is the other way to overcome this problem. For example in a self-organizing map (SOM), each node is a representative (a center) of a cluster of similar points, regardless of their density in the original training data. K-NN can then be applied to the self-organizing map.

1) Parameter Selection

The best choice of k depends upon the data; generally, larger values of k reduce the effect of noise on the classification [9], but make boundaries between classes less distinct. A good k can be selected by various heuristic techniques (see hyper parameter optimization). The special case where the class is predicted to be the class of the closest training sample (i.e. when k = 1) is called the nearest neighbor algorithm.

The accuracy of the K-NN algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance. Much research effort has been put into selecting or scaling features to improve classification. A particularly popular approach is the use of evolutionary algorithms to optimize feature scaling. Another popular approach is to scale features by the mutual information of the training data with the training classes. In binary (two class) classification problems, it is helpful to choose k to be an odd number as this avoids tied votes. One popular way of choosing the empirically optimal k in this setting is via bootstrap method.

The proposed work is mainly divided into two broad categories; first phase of the proposed work is the extraction of character feature using proposed technique and creation of feature database for training. The second phase is the testing of developed system for various characters.

V. GUI OF RECOGNITION SYSTEM

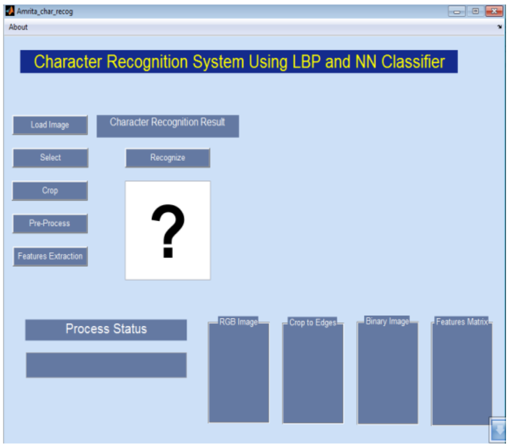

Figure-4 The screen shot of the developed GUI of Recognition system

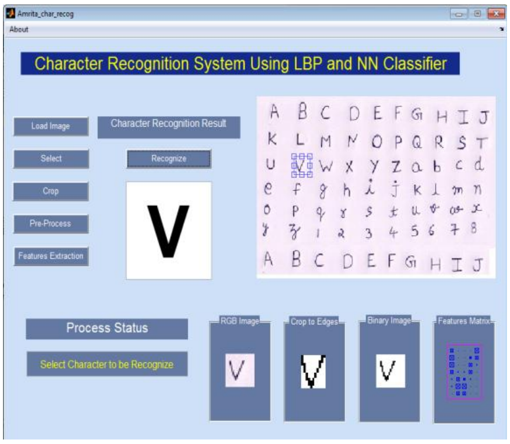

Figure-5 Single Character Recognition process

VI. RESULT AND DISCUSSION

The recognition system has been implemented using MATLAB 2012 (b).The scanned image is taken as dataset/ input and KNN classifier is used. A set of seventy alphabets including capital letters, small letters and numbers are used for testing of the proposed method. The proposed recognition system performs efficiently yielding higher levels of recognition accuracy compared to the systems employing the conventional feature extraction methods. Total 284 single characters have been tested for the recognition and out of 284 characters the developed system recognized 280 characters efficiently, and 4 characters were recognized wrongly by the developed system. Hence the character recognition efficiency showing by the developed system is 98.591 percent, which the highest character recognition efficiency as compare to all available character recognition systems.

VII. CONCLUSION AND FUTURE SCOPE

Handwritten Alphabet Recognition is one of the artificial intelligence applications which provide an important fundamental for various advanced applications, including information retrieval and human-computer interaction applications. For the future work this system can be further extending by fusion of some advance features like texture based features to increase the recognition efficiency further.

REFERENCES

[1] S. Mori, C.Y. Suen and K. Kamamoto, “Historical review of OCR research and development,” *Proc. of IEEE*, vol. 80, pp. 1029-1058, July 1992.

[2] V.K. Govindan and A.P. Shivaprasad, “Character Recognition – A review,” Pattern Recognition, *IEEE*, vol. 23, no. 7, pp. 671- 683, 1990.

[3] Mantas, J, “An overview of character recognition methodologies,” Pattern Recognition, vol.19, Issue 6, pp. 425-430, 1986.

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

[5] Gita Sinha, Anita Rani, Prof. Renu Dhir, Mrs. Rajneesh Rani, “Zone-Based Feature Extraction Techniques and SVM for Handwritten Gurmukhi Character Recognition”, *IJARCSSE*, vol. 2, Issue 6, June 2012.

[6] Juarez Paulino da Silva J´unior, Marcus Vinicius Lamar, Jacir Luiz Bordim, “Accuracy and Efficiency Performance of the ICP Procedure Applied to Sign Language Recognition” , *Clei electronic journal*, vol. 17, number 2, paper 10, august 2014.

[7] Padmavathi . S, Saipreethy.M.S, Valliammai.V, “Indian Sign Language Character Recognition using Neural Networks”, *IJCA Special Issue on “Recent Trends in Pattern Recognition and Image Analysis"* RTPRIA 2013.

[8] Binod Kumar Prasad, Goutam Sanyal, **“**A Model Approach to off-line English Character Recognition”, *International Journal of Scientific & Research Publications*, vol. 2, Issue 6, June 2012, ISSN 2250-3153.

[9] R. Plamondon and S. N. Srihari, “On-line and off- line handwritten character recognition: A comprehensive survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63-84, 2000.

[10] Pradeep, E.Srinivasan and S.Himavathi, " Diagonal Based Feature Extraction For Handwritten Alphabets Recognition System Using Neural Network", *IJCSIT*, Vol 3, No 1, Feb 2011.

[11] Rafael C. Gonzalez, Richard E. woods and Steven L. Eddins, *Digital Image Processing using MATLAB*, Pearson Education, Dorling Kindersley, South Asia, 2004.

