Pak. J. Engg. & Appl. Sci. Vol. 10, Jan., 2012 (*p.* 57-67)

Handwritten Character Recognition Using Multiclass SVM Classification with Hybrid Feature Extraction

Muhammad Naeem Ayyaz [1], Imran Javed [2] and Waqar Mahmood [3]

1. *Department of Electrical Engineering, University of Engineering and Technology, Lahore, Pakistan. (e-mail: mna@uet.edu.pk)*
2. *Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, Lahore, Pakistan. (e-mail: imran.javed@kics.edu.pk)*
3. *Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, Lahore, Pakistan. (e-mail: drwqar@kics.edu.pk)*

Abstract

In this paper, we describe hybrid feature extraction for offline handwritten character recognition. The proposed technique is a hybrid of structural, statistical and correlation features. In the first step, the proposed technique identifies the type and location of some elementary strokes in the character. The strokes to be looked for comprise horizontal, vertical, positive slant and negative slant lines–as we observe that the structure of any character can be approximated with the help of a combination of simple straight line strokes. The strokes are identified by correlating different segments of the character with the chosen elementary shapes. These normalized correlation values at different segments of the character give correlation features. For making feature extraction more robust, we add in the second step certain structural/statistical features to the correlation features. The added structural/statistical features are based on projections, profiles, invariant moments, endpoints and junction points. This enhanced, powerful combination of features results in a 157-variable feature vector for each character, which we find adequate enough to uniquely represent and identify each character. Prior, handwritten character recognition problem has not been addressed the way our proposed hybrid feature extraction technique deals with it. The extracted feature vector is used during the training phase for building a support vector machine (SVM) classifier. The trained SVM classifier is subsequently used during the testing phase for classifying unknown characters. Experiments were performed on handwritten digit characters and uppercase alphabets taken from different writers, without any constraint on writing style. The obtained results were compared with some related existing approaches. Owing to the proposed technique, the results obtained show higher efficiency regarding classifier accuracy, memory size and training time as compared to these other existing approaches.

Key Words: character recognition, feature extraction, correlation function, SVM

1. Introduction

Handwritten character recognition (HCR) is the computer based identification of handwritten numerals and alphabets. HCR is a step towards the automation of human interaction with machines. HCR has applications for assisting visually-impaired people; for automatic database recording and filtering of written documents; writer identification and signature verification etc. [1]. Despite its tremendous scope of application, HCR is a difficult object classification task because each writer has its own way of writing characters and writing fashion varies for a single writer too.

1.1 Feature extraction and related work

One of the most important phases in successfully achieving character recognition is the task of *feature extraction*. Feature extraction stage identifies and extracts various attributes from characters that help distinctly and uniquely distinguish different characters. A number of different feature extraction methods have been proposed in literature in accordance with different character representations. For example, different sets of features have been defined to best represent character shapes, boundaries, their skeletons and strokes etc. Trier et al. [2] comprehensively describe different types of features and methods for character recognition task. Among these methods, there are statistical feature extractors and structural feature extractors. Statistical features consider the statistical distribution of pixel values. Major statistical features used for handwritten character recognition task include zoning, projections, profiles, and crossings etc. Structural features consider the geometry and topology of character samples such as number of loops, end points, junction points, aspect ratio, type of strokes and their directions etc. Some feature extraction methods are based on different transformations such as those based on Fourier transform, wavelet transform, central moments, and Zernike moments etc. In [3], the authors describe a zoning based feature extractor to recognize handwritten numerals of Indian Kannada script. Authors in [4] recognize handwritten numerals using Fourier descriptors and neural network. In [5], the authors recognize Chinese handwritten characters using gradient and wavelet based features. In [6], the authors extract moment based features in order to recognize handwritten Arabic letters. They use genetic algorithm for feature selection and use SVM to evaluate the classification error for the chosen feature subset.

Instead of focusing on feature vector based on a single representation of a character, it is a trend now of combining different types of features extracted from different representations of the same character. The advantage of combining, and harnessing, such different kinds of features is that it can offer wider range of identification clues to help improve the accuracy of recognition. For example, Heutte et al. [7] combine different statistical and structural features for recognition of handwritten characters. They construct a 124-variable feature vector comprising following seven families of features: 1) intersection of the character with horizontal and vertical straight lines, 2) invariant moments, 3) holes and concave arcs, 4) extremas, 5) end points and junction points 6) profiles, and 7) projections. Aurora et al. [8] combine different feature extraction techniques such as intersection based features, shadow features, chain code and curve fitting features for Indian Devnagari language script. Kimura et al. [9] propose a genetic algorithm based strategy for finding a suitable combination of features from a large pool of features with the objective criteria to minimize the classification error.

1.2 Pattern classification and related work

The second most important component in successfully achieving handwritten character recognition is the *pattern classification* stage. This stage will assign an unknown character sample to one of possible classes by utilizing the information of feature extraction stage. Different types of classifiers can be built based on the nature and type of data samples and the extracted features. Classifiers used for character recognition problem include k-nearest neighbor classifier, hidden Markov model (HMM), support vector machine (SVM), and artificial neural network (ANN) etc. Jain et al. [10] give a review of statistical pattern recognition techniques. In [11], Pal and Singh train neural network to recognize uppercase handwritten characters based on Fourier descriptors of character boundaries as features. In [12], recognition of handwritten alphabets using neural network and zoning based diagonal features is addressed. In [13], Shubhangi and Hiremath recognize English handwritten characters and digits by extracting structural micro features for SVM classifier. Nasien et al. [14] also use SVM classifier to recognize handwritten alphabets by employing Freeman Chain codes as the features. In [15], Train et al. recognize accented handwritten French characters based on a combination of structural and moment features for SVM classifier. In [16], Liu and Nakagawa give a review of learning methods for nearest neighbor classifiers. [17] and [18] build HMM to recognize, respectively, offline handwritten Chinese characters and online English characters.

1.3 Present Work

In this paper we propose a different hybrid feature extraction technique which comprises a group of 100 correlation features alongside with another 57 structural/statistical features. Our correlation features are based on Pearson’s correlation coefficient [19-20] which has been widely applied for the purpose of measuring similarity or disparity among the images. The value of correlation coefficient indicates the extent to which two images are similar. Here, we seek the application of Pearson’s correlation coefficient in a different way so as to identify the basic elementary strokes in handwritten characters. For this, we compute the correlation coefficient among different character segments and the chosen elementary shapes. We transform the character images in frequency domain then we normalize their energy values as it is a well known fact in signal processing theory that the correlation in spatial domain is simply the multiplication in frequency domain. Shioyama and Hamanaka [21] extract similar correlation function based features for the problem of Chinese hand-printed character recognition. They however perform their classification based on minimum distance decision rule. We, on the contrary, perform final classification based on support vector machine (SVM). The biggest challenge, in achieving high accuracy results for SVM classification problems, is the extraction of robust features from the data samples. Shioyama and Hamanaka [21] argue that since the correlation function is based on power spectral density of character images, it is invariant under a translational transform and hence can absorb the local variation in hand-printing. In this paper, we test the application of this correlation function based approach to the domain of English handwritten alphabets and numerals. To the best of our knowledge, such Fast Fourier Transform (FFT) based correlation approach has not been yet applied for the classification of English handwritten character samples, though some significant work on fuzzy rules based identification of lines and curve strokes in the characters does exist [22-23].

In our case of unconstrained handwritten character recognition problem, these correlation features alone did not give satisfactory accuracy for SVM classification. To make the feature vector more robust, with regard to capability of better distinguishing the characters, we combine correlation function based features with a number of structural or statistical features. Some structural uncovered features are end points and junction points which we add to the correlation features. Finally we add profiles, projections, and moment features to our correlation features as these are based on binary images of characters whereas correlation features are based on skeletonized characters. This sophisticated combination of features has resulted in a feature vector which, as results show, has proven robust to character style variations and shows better results as compared to related existing approaches. To the best of our knowledge, such proposed hybrid feature extraction technique also has not been yet applied the same way for character recognition problem. After feature extraction stage, we train SVM classifier on extracted features of training data. The built SVM model is subsequently used for the recognition of unknown characters during the testing phase.

1.4 Paper Organization

This paper is organized as follows: in section 2, we describe in detail our proposed methodology for handwritten character recognition. Section 3 presents experimental results, analysis and discussion. Section 4 and section 5, respectively, describe future work and conclusion of our research.

2. Proposed Methodology

Our proposed work presents a complete handwritten character recognizer. The system can be split into three stages (as shown in figure 1): a) pre-processing, b) proposed feature extraction scheme, and c) SVM-based training and classification. In the following we will describe each of these sub-stages in detail.

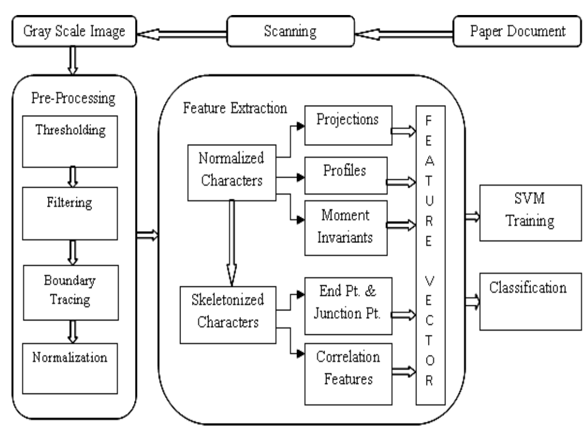

Fig.1: Block diagram of complete HCR system

2.1 Preprocessing

We collected data samples of handwritten alphabetic characters and digit characters from different writers on plain white papers. The writers were allowed to write the characters without any constraint on writing style or equipment. The written documents were then scanned at 150 dpi resolution and stored in the PC as gray-scale images. An example image is shown in figure 2.

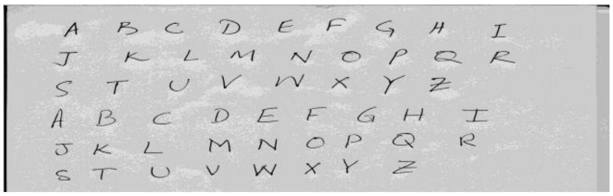

Fig. 2: Gray-scale image

Pre-processing step was performed on these images to remove the noise from these documents and also to minimize the variations in character styles. For example, we often observe long, slight shades at the edges of scanned images. Sometimes, the document was not clean that produced small dots in the images. This noise generated by shaded areas and dots must be filtered during pre-processing step. Moreover, the characters must be skew, slant and size normalized to minimize the writing variations. Typical steps performed during pre-processing were the following:

2.1.1 *Thresholding*

This is used to convert gray-scale image to binary image. A threshold is defined for this purpose; the pixels above this threshold are set to white and those below the threshold are set to black. We tested different values of threshold on the scanned images. Threshold value of 190 produced good quality binary images. The result of thresholding step is shown in figure 3.

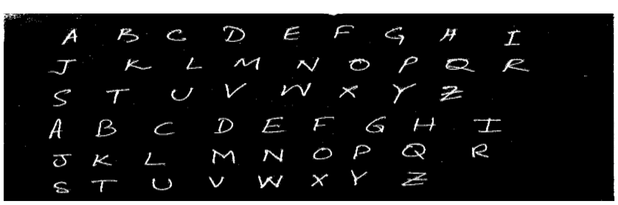

Fig. 3: Binary image

2.1.2 *Filtering*

This is used to remove the noise present in the binary image. To remove small black specks in the image and the black shade appearing at the edges, we performed median filtering. Further, the documents were cropped from the edges. We found that the thresholding and filtering steps often resulted in some broken characters. To rejoin the broken characters, we performed image dilation operation on the filtered images. An example filtered image is shown in Fig.4.

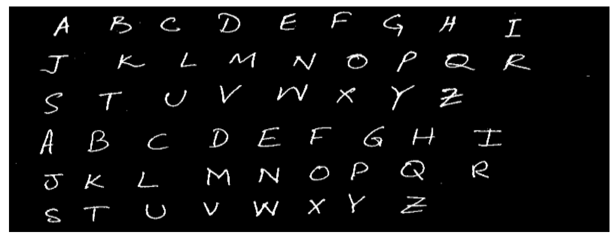

Fig. 4: Filtered image

2.1.3 *Boundary Tracing*

This step identifies the connected components i.e. characters in the filtered images and stores them in an array. To find the connected components, our algorithm starts by traversing the rows of filtered image. If in any row the algorithm finds a foreground pixel, it marks that pixel then it picks and marks all the neighbors of found pixel in different search directions till all the pixels of the potential character have been traversed and marked. If in any row the algorithm does not find any foreground pixel, it will continue its search in the next row. If the size of any picked connected component is too small (less than 10 pixels), the algorithm treats that component as noise and thus discards it.

2.1.4 *Normalization*

During normalization step, we remove slant in characters and resize them to a 30×30 window. Slant is the average divergence of the vertical strokes of the character from the right angle position. To remove the slant, we employed projection histogram based technique [24]. We limit the search of slant angle to [-11, 11] and apply the following horizontal shear transformation to the character image:

$$\begin{cases} x_5 = x - (y - y_c).\tan(\theta) \\ y_5 = y \end{cases} \quad (1)$$

In equation (1), each image pixel (*x, y*) is transformed to new pixel value (x_s, y_s), y_c is ycoordinate of centre and θ is the angle of transformation. At each angle then we calculate the sum of vertical projection profiles of the transformed character. The angle with the maximum sum of vertical projection profile is the required slant angle. Finally, we apply inverse shear transformation to the character image through the estimated slant angle. For the characters of figure 4, normalized images of characters are shown in figure 5.

Fig. 5: Normalized characters

2.1.5 Skeletonization

This step reduces the thickness of character image to one-pixel by removing the pixels on the boundary of the character but without breaking the character. A number of efficient algorithms for skeletonization have been proposed in literature [25-26]. We performed thinning operation on the handwritten characters as for example shown in figure 6. We extract some features on character skeletons such as those of endpoints, junction points and features based on correlation function.

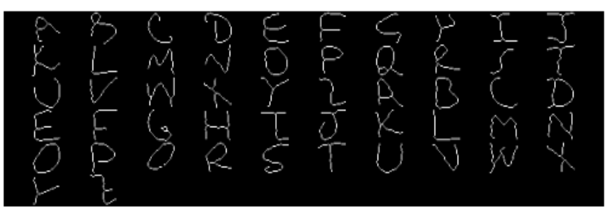

Fig.6: Skeletonized characters

2.2 Proposed feature extraction

Our proposed feature vector for handwritten data samples has a length of 157 variables. It is a hybrid of correlation function based features and some structural/statistical features as described earlier. To the best of our knowledge such hybrid scheme of feature composition (*i.e.* correlation function based features combined with some structural/statistical features) has not yet been applied and tested the same way for the problem of handwritten character data samples. We have successfully applied and tested such combination of features to capturing character variations in experimental data of several different writers. The scheme has demonstrated high recognition accuracy on the testing data, as described later in section 3. In the following, we describe our proposed feature extraction technique in detail.

2.2.1 *Correlation function based features*

We extract correlation function based features from the skeletonized characters. The normalized character images are of size 30×30. We divide each character into different segments, each of size 9×9. We define four elementary shapes, each of the size same as that of segments, *i.e.* 9×9. The four elementary shapes chosen are horizontal line, vertical line, positive slant line and negative slant line. These entire four elementary shapes were located at the central position of the 9×9 window. Our feature extraction stage identifies and determines the type and location of the elementary shapes in the character samples. It does so by correlating the different segments of the character sample with the chosen elementary shapes through Pearson’s correlation function [19] in frequency domain. To compute the correlation, we need to define the correlation points in the 30×30 window of the normalized character samples. Experimental simulations show that the following correlation points give good classification results on the data samples of handwritten English alphabets and digits:

Horizontal axis of normalized 30×30 character image:

$$m = [5;10;15;20;25] \qquad (2)$$

Vertical axis of normalized 30×30 character image:

$$n = [5;10;15;20;25] \qquad (3)$$

Combining the axes in equation 2 and 3, we obtain following correlation points around which we take the different segments of each character:

$$mn = [5\ 5;5\ 10;5\ 15;5\ 20;5\ 25;\ 10\ 5;10\ 10;10\ 15;10\ 20;10\ 25;\ 15\ 5;15\ 10;15\ 15;15\ 20;15\ 25;\ 20\ 5;20\ 10;20\ 15;20\ 20;20\ 25;\ 25\ 5;25\ 10;25\ 15;25\ 20;25\ 25]; \quad (4)$$

Now, we divide the character into different segments. Each segment is of size 9×9 with its centre at the points of '*mn*' vector in equation (4). The graphical form of this whole procedure is shown in figure 7 which contains four elementary shapes on the left side and an example handwritten character on the right side. The example handwritten character is divided into segment windows (two segment windows shown as example in figure 7). The centre of each segment window is indicated by a cross ($\times$).

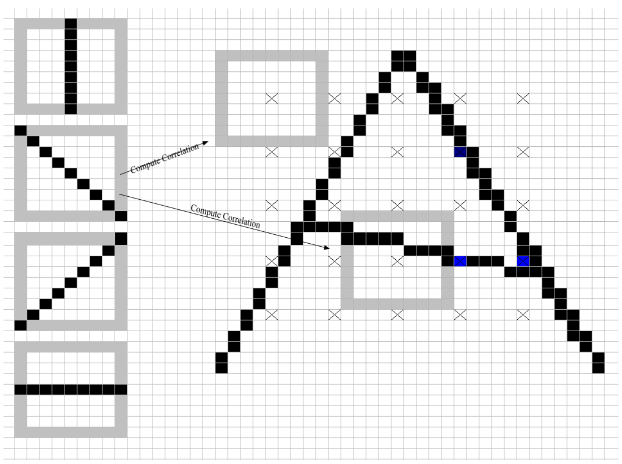

Fig.7: Computing correlation of elementary shapes with different segments of handwritten character

We find the normalized correlation value of different segments with the elementary shapes by the following procedure:

Step *a*: Take 2-dimesional Fourier transform of both the t^{th} elementary shape and the different 9×9 character segments centered at mn^{th} position as follows.

$$t\_PIC(w1, w2) = FFT2\{t\_PIC(r, c)\} \quad (5)$$

$$mn\_SEG(w1, w2) = FFT2\{mn\_SEG(r, c)\} \quad (6)$$

Step *b*: Now compute the normalized correlation value by Pearson's correlation function as in eqn. 7.

$$h(t,m,n) = \frac{\sum_{w1,w2=1}^{N} t\_PIC(w1,w2) \times mn\_SEG(w1,w2)}{\sqrt{\sum_{w1,w2=1}^{N} |t\_PIC(w1,w2)|^2} \sqrt{\sum_{w1,w2=1}^{N} |mn\_SEG(w1,w2)|^2}} \quad (7)$$

Step c: Repeat steps *a* and b for all the elementary shapes, *i.e.* $t = 1, 2, 3, 4$.

This gives a total of 100 correlation features, 25 for each elementary picture. An important property of these features is that, since they are based on power spectral density [21] of character images, so they are invariant under a translational transform in the character images.

2.2.2 *Structural/statistical features*

For our unconstrained handwritten character recognition problem, our results show that the extracted correlation features alone do not give satisfactory accuracy for SVM classification. To make the feature vector more robust with respect to capability of better distinguishing characters, we needed to provide some more identification clues to the SVM classifier. For this purpose, we combined correlation function based features with some structural/statistical features. On this new combination of features, the SVM classifier showed high recognition accuracy during the testing phase. The 57 more statistical/structural features added to the correlation features are described in some detail below:

2.2.2.1 *End points and junction points:*

These features are extracted from the thinned characters. End points are those having only one neighbor, while junction points have at least 3 neighbors. We select the number of end points, the number of junction points, and the x-y locations of these points as the features to be stored. Since the number of end points and junction points can vary from one character type to another, we need some strategy to convert these features into fixed length vector. For this purpose, we employ the strategy of [7]. Maximum number of end points and junction points, call it *p*, are noted down from the training data, and their average value with corresponding x-y position is computed. If any character has less than *p* pointes, then empty row in feature vector is filled with the average value. If during testing phase, the character happens to have greater number of points than *p*, then extra points are simply discarded. All these features are normalized in range [0, 1].

2.2.2.2 *Projection histogram:*

We compute the normalized features as in [7] from cumulative vertical and horizontal projection histogram of the characters. The Y-axis of both histograms is divided into 11 equal parts and 10 corresponding points on the X-axis are taken as the features.

2.2.2.3 *Profiles:*

We compute normalized features as in [7] from the top, bottom, left and right profiles. After computing the profiles, we find first-difference profiles by taking the difference of a point from its previous one. The maximas in four difference profiles are stored as features. Then the difference between left and right profiles at 1/3, 2/3, and 5/6 of the character's height is recorded as features. Similarly, the difference between top and bottom profiles at 1/3, 2/3, and 5/6 of the character's width are stored as features.

2.2.2.4 *Invariant moments:*

The invariant moments [27] measure pixel distribution around the center of character image. These moments are invariant to position, size and orientation of the character. We compute seven invariant moments for all the characters and store them as features.

2.3 SVM based classification

Once the feature extraction stage is complete, our next phase was to build an intelligent classifier on the extracted feature vector of all the data samples. In this research we have chosen the SVM classifier [28-29] for training and classification purpose. SVM is a 2-class classifier that separates the data samples of two classes by computing a maximum-margin boundary between them. The solution for this separating boundary is expressed in the form of a mathematical optimization problem and it is well-established in SVM literature [29]. In case, the data is nonlinearly separable, SVM makes the data linearly separable using kernel functions. A kernel function maps the input data patterns to some high dimensional space to make the points linearly separable in high dimensional space. Common kernel functions used for classification are Gaussian radial basis function, hyperbolic tangent, polynomial kernel, etc. The separating boundary between the two classes is defined as

$$w^T x_i + b \geq 1 - \xi_i \qquad y_i = 1 \tag{8}$$

$$w^T x_i + b \leq -1 + \xi_i \qquad y_i = -1 \tag{9}$$

$$\xi_i \geq 0 \qquad \forall i \tag{10}$$

where ξi counts the number of misclassified samples. For training data vector $x_1 \varepsilon R^n$ $i=1,2,..m$ and class labels $y_i, y_i \in \{1,-1\}$, the SVM solves the following optimization problem:

$$\text{Minimize:} \quad \frac{\|w\|^2}{2} + c\sum_{i=1}^{l} \xi_i \tag{11}$$

$$\text{subject to: } y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0 \tag{12}$$

where $c > 0$ is used to impose penalty for error terms. Based on Lagrange multipliers, equation 11 and 12 can be expressed in their dual forms as

$$\text{Maximize: } W(\alpha) = \sum_{i=1}^{n} \alpha_i - 1/2 \sum_{i=1, j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{13}$$

$$\text{subject to:} \quad \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{14}$$

α_1 have nonzero values for data points close to the decision boundary and are called Support Vectors (SV). These SV determine the separating hyperplane. The binary SVM classification problem can be converted to multi-class classification by building a number of 2-class SVM classifiers for different class pairs and then taking the final classification decision based on different strategies such as max-wins strategy, winner-takes-all strategy etc. Max-wins strategy is the majority-voting decision of all the 2-class SVM classifiers. In winner-takes-all strategy, the binary classifier with highest output function takes the decision of classification. Common existing approaches [30-32] for multi-class classification problem are one-against-one (OAO), one-against-all (OAA), binary tree of SVM and directed acyclic graph (DAG) etc. In this research, we have chosen OAO technique for multi-class classification.

3. Results, Analysis and Discussion

We tested this system on handwritten characters taken from 30 different writers, who were allowed to write in their natural style. The whole system was implemented in MATLAB. After the pre-processing stage, we extracted a total of 6092 characters for handwritten uppercase alphabets and 2279 handwritten digits from the scanned documents. Data samples were divided into two parts: a two-third of data samples was reserved for training purpose while one-third of data samples was reserved for testing purpose. Accordingly, alphabets training data consisted of 4067 characters while alphabets testing data comprised 2025 characters. Similarly, digits training data consisted of 1857 numerals while digits testing data consisted of 922 numerals. Feature vectors of dimension 157 were extracted for the training data of handwritten characters and numerals. One SVM model was trained on 157×4067 feature matrix of alphabets and another was trained on 157×1857 feature matrix of handwritten digits. SVM parameters on training data were fine-tuned using 3-fold cross-validation. Once the SVM models of handwritten alphabets and digits were trained, we checked performance of the recognition system on reserved testing data sets. Out of the testing data, only 32/922 digits and 80/2025 alphabets were misclassified. This gives 96.5% recognition accuracy on chosen digits data and 96% recognition accuracy on chosen alphabets data. The system showed 100% accuracy on training data of both alphabets and numerals. We compared performance of our proposed technique with [7] and [21] on the same handwritten data samples. The SVM classifiers were trained and tested on the feature extraction techniques of [7] and [21]. The results are shown in Table 1.

Table 1 shows that our proposed technique is more accurate and more efficient compared to the other two approaches. Its training time and memory size of found classifier is much less compared to the other two approaches. The system has also higher recognition rate as compared to other two approaches. We further examined the performance of our system on data samples of a new writer not originally among the 30 writers on whom the system was trained and tested. Figure 8 and figure 9 show performance of the system on this new writer. We observed during the feature extraction stage that the thinning process sometimes eliminates important character strokes which cause some characters to get misclassified. The system performance can therefore be further improved by refining the thinning stage.

4. Further Research

Our proposed hybrid feature extraction technique in conjunction with SVM classifier has shown good performance on handwritten digits and uppercase alphabets. In future, we intend to test the performance of proposed technique on lower case alphabets. To obtain satisfactory accuracy on lowercase characters, our hybrid technique might adjust the window size and shape of elementary segments including, if necessary, some micro structural features specific to lower case characters.

Table-1: Comparison of different approaches on the collected handwritten data set

Approach	Number of SV's Digits	Number of SV's Alphabets	Training Time (Sec) Digits	Training Time (Sec) Alphabets	Size of Classifier (MB) Digits	Size of Classifier (MB) Alphabets	% Accuracy Digits	% Accuracy Alphabets
Heutte [7]	358	1111	216	8830	0.565	2.55	94.37	95.44
Shioyama [21]	424	1779	350	3357	0.786	1.10	93.33	89.15
Ours	548	1743	68	2010	0.366	1.16	96.5	96

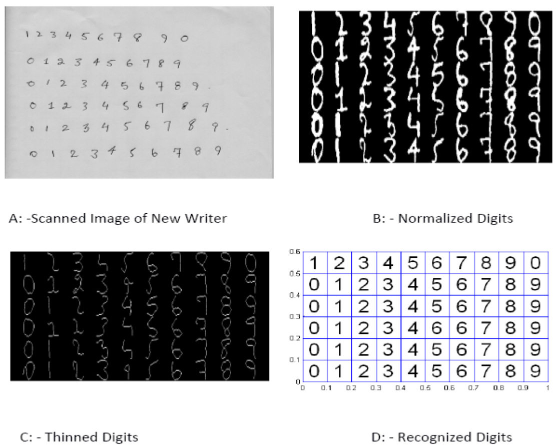

Fig.8 System's performance on digit samples of new writer

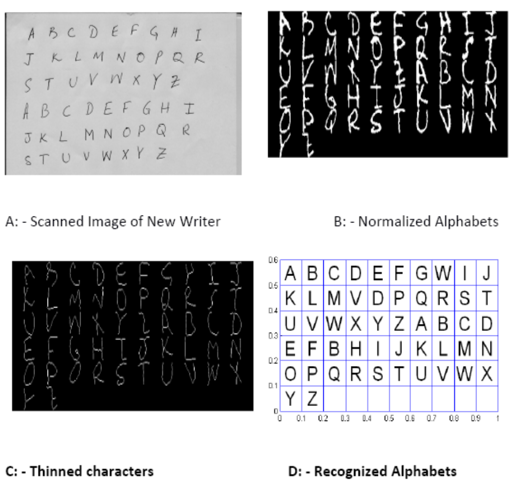

Fig. 9: System's performance on alphabet samples of new writer

5. Conclusion

A complete offline handwritten character recognition system based on a hybrid feature extraction technique has been presented. The system comprised three main stages, *i.e.* pre-processing, feature extraction technique, and SVM based training/classification. The proposed hybrid feature extraction technique, as experiments revealed, proved to capture local and global variations in handwritten character styles. The extracted feature vector was a combination of correlation function based features and some statistical/structural features. The correlation features identified the straight line strokes in characters which were then equipped with some statistical/structural features to make the feature vector more robust. Owing to the proposed feature extraction technique, the trained support vector machine classifier has shown higher efficiency with respect to speed, memory, and classification accuracy as compared to other related approaches dealing the handwritten character recognition problem.

6 References

[1] N. Shanthi and K. Duraiswamy, "A novel SVM based handwritten Tamil character recognition system," *Springer Journal on Pattern Analysis & Applications*, Feb 2009.

[2] O. D. Trier, A. K. Jain and T. Taxt, " Feature extraction methods for character recognition: a survey" *Pattern Recognition* 29 (4), 641-662. 1996.

[3] S. V. Rajashekararadhya and P. V. Ranjan, "Zone based feature extraction algorithm for handwritten numeral recognition of Kannada script," *IEEE International Advance Computing Conference (IACC)*, Patiala, India, March 2009.

[4] Y. Y. Chung and M. T. Wong, "Handwritten character recognition by Fourier descriptors and neural network," *IEEE TENCON, Speech and Image Technologies for Computing and Telecommunications*, 1997.

[5] W. Zhang, Y. Y. Tang and Y. Xue, "Handwritten character recognition using combined gradient and wavelet features," *International Conference on Computational Intelligence and Security*, pp. 662-667, Guangzhou, Nov. 2006.

[6] G. Abandah and N. Anssari, "Novel moment features extraction for recognizing handwritten Arabic letters," *Journal of Computer Science*, vol. 5, issue 3, pp. 226-232, 2009.

[7] L. Heutte, J. V. Moreau, T. Paquet, Y. Lecourtier, and C. Olivier, " Combining structural and statistical features for the recognition of handwritten characters," *Proceedings of 13th International Conference on Pattern Recognition*, Vienna, Austria, 1996, Vol. 2, pp. 210-214.

[8] S. Arora, D. Bhattacharjee, M. Nasipuri, D. K. Basu and M. Kundu, "Combining multiple feature extraction techniques for handwritten Devnagari character recognition," *IEEE Region 10 Colloquium and 3rd International Conference on Industrial and Information Systems*, Dec. 2008.

[9] Y. Kimura, A. Suzuki, K. Odaka, "Feature selection for character recognition using genetic algorithm," *IEEE Fourth International Conference on Innovative Computing, Information and Control (ICICIC)*, Kaohsiung , pp. 401-404, Dec. 2009.

[10] A. K. Jain, P. W. Duin, and J. Mao, "Statistical pattern recognition: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, Jan. 2000.

[11] A. Pal and D. Singh, "Handwritten English character recognition using neural network," *International Journal of Computer Science and Communication*, vol. 1, no. 2, pp. 141-144, July-Dec 2010.

[12] J. Pradeep, E. Srinivasan and S. Himavathi, "Diagonal based feature extraction for handwritten alphabets recognition system using neural network," *International Journal of Computer Science and Information Technology*, vol. 3, no. 1, Feb. 2011.

[13] D. C. Shubhangi and P. S. Hiremath, "Handwritten English character and digit recognition using multiclass SVM classifier and using structural micro features," *International Journal of Recent Trends in Engineering*, vol. 2, no. 2, Nov. 2009.

[14] D. Nasien, H. Haron and S. S. Yuhaniz, "Support vector machine for English handwritten character recognition," *2nd International Conference on Computer Engineering and Applications*, 2010.

[15] D. C. Tran, P. Franco and J. M. Ogier, "Accented handwritten character recognition using SVM: application to French," *12th International Conference on Frontiers in Handwriting Recognition*, 2010.

[16] C. L. Liu and M. Nakagawa, "Prototype learning algorithms for nearest neighbor classifier with application to handwritten character recognition," *Proc. 5th International Conference on Document Analysis and Recognition (ICDAR)*, Bangalore, India, Sep. 1999.

[17] B. Feng and X. Ding, "Offline handwritten Chinese character recognition with hidden Markov models," *Proc. 5th ICSP*, vol. 3, pp. 1542-1545, Beijing, China, 2000.

[18] S. R. Veltman and R. Prasad, "Hidden Markov models applied to online handwritten isolated character recognition," *IEEE Transactions on Image Processing*, vol. 3, issue 3, pp. 314-318, May 1994.

[19] A. Miranda Neto, L. Rittner, N. Leite, D. E. Zampieri, R. Lotufo and A. Mendeleck, "Pearson's correlation coefficient for discarding redundant information in real time autonomous navigation system", *16th IEEE International Conference on Control Applications, Part of IEEE Multi-conference on Systems and Control*, 1-3 October 2007, Singapore.

[20] Y.K. Eugene and R.G. Johnston, "The ineffectiveness of the correlation coefficient for image comparisons", *Technical Report LA-UR-96-2474*, Los Alamos, 1996.

[21] T. Shioyama, J. Hamanaka, "Recognition algorithm for handprinted Chinese characters by 2D-FFT," *Proceedings of 13th International Conference on Pattern Recognition*, Vienna, Austria, 1996, Vol. 3.

[22] *K.B.M.R. Batuwita and G.E.M.D.C. Bandar*, "Fuzzy recognition of offline handwritten numeric characters," *Proceedings of IEEE International Conference on Cybernetics and Intelligent Systems*, 2006.

[23] R. Ranawana, V. Palade and G. E. M. D. C. Bandara, "An efficient fuzzy method for handwritten character recognition" *Springer Verlag*, KES 2004, pp. 698-707.

[24] P. Nagabhushan, S. A. Angadi, B. S. Anami, "Geometric model and projection based algorithms for tilt correction and extraction of acsenders / descenders for cursive word recognition," IEEE *International Conference on Signal Processing, Communications and Networking (ICSCN '07),* Chennai, pp. 488-491, Feb. 2007.

[25] Lam L, Lee S-W, Suen CY, "Thinning methodologies: a comprehensive survey," *IEEE Trans Pattern Analaysis and Machine Intelligence (PAMI),* Vol. 14, No. 9, pp. 869-885, September 1992.

[26] Xuefang Zhu and Shuyi Zhang , "A shape-adaptive thinning method for binary images," *IEEE International Conference on Cyberworlds,* Hangzhou, pp. 721-724, Sep. 2008.

[27] M.K. Hu, "Visual pattern recognition by moment invariants", *IRE Transactions on Information Theory*, pp. 179-187, 1962.

[28] D. Elizondo, "The linear separability problem: some testing methods," *IEEE Transactions on Neural Networks,* vol. 17, no. 2, March 2006.

[29] Nello Cristianini and John Shawe-Taylor, "An introduction to support vector machines and other kernel-based learning methods", Cambridge University Press, New York, NY, 1999,

[30] Chih Wei Hsu and Chih Jen Lin, "A comparison of methods for multi-class support vector machines", *IEEE Trans. On Neural Networks*, Vol. 13, No. 2, March 2002.

[31] B. Fei, J. Liu, "Binary tree of SVM: a new fast multi-class training and classification algorithm," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, May 2006.

[32] S. Cheong, S. H. Oh, and S. Y. Lee, "Support vector machines with binary tree architecture for multi-class classification," *Neural Info. Process*. Lett., vol. 2, no. 3, Mar. 2004.

[33] H. Lei, V. Govindaraju, "Half-against-half multi-class support vector machine," *Journal of Machine Learning Research*, 2004.