PCA Based Handwritten Character Recognition System Using Support Vector Machine & Neural Network

Ravi Sheth
[1]*Information Technology Dept., A.D Patel Institute of Technology, New V V nagar-388120, Gujarat, India*
[1]*raviesheth@gmail.com*

N C Chauhan
[2]*Information Technology Dept., A.D.Patel Institute of Technology, New V V nagar-388121, Gujarat, India*

Mahesh M Goyani
[3]*Computer Engineering. Dept., L.D.college of engineering, Ahmadabad, Gujarat, India*

Kinjal A Mehta
[4]*Electronics and Communication Dept., L.D. college of engineering, Ahmadabad, Gujarat, India*

Abstract
Pattern recognition deals with categorization of input data into one of the given classes based on extraction of features. Handwritten Character Recognition (HCR) is one of the well-known applications of pattern recognition. For any recognition system, an important part is feature extraction. A proper feature extraction method can increase the recognition ratio. In this paper, a Principal Component Analysis (PCA) based feature extraction method is investigated for developing HCR system. PCA is a useful statistical technique that has found application in fields such as face recognition and image compression, and is a common technique for finding patterns in data of high dimension. These method have been used as features of the character image, which have been later on used for training and testing with Neural Network (NN) and Support Vector Machine (SVM) classifiers. HCR is also implemented with PCA and Euclidean distance.

Keywords: *Pattern recognition, handwritten character recognition, feature extraction, principal component analysis, neural network, support vector machine,* e*uclidean distance.*

1 INTRODUCTION

Hhandwritten character hrecognition is an area of pattern recognition that has become the subject of research during the last few decades. Handwriting recognition has always been a challenging task in pattern recognition. Many systems and classification algorithms have been proposed in the past years. Techniques ranging from statistical methods such as PCA and Fisher discriminate analysis [1] to machine learning like neural networks [2] or support vector machines [3] have been applied to solve this problem. The aim of this paper is to recognize the handwritten English character by using PCA with three different methods as mentioned above. The handwritten characters have infinite variety of style varying from person to person. Due to this wide range of variability, it is very difficult for a machine to recognize a handwritten character; the ultimate target is still out of reach. There is a huge scope of development in the field of handwritten character recognition. Any future process in the field of handwritten character recognition will able to increase the communication between machine and men. Generally HCR is divided in four major parts as shown in Fig. 1[4]. These phases include binarization, segmentation, feature extraction and classification. Few major problems faced while dealing with segmented, handwritten character recognition is the ambiguity and illegibility of the characters. The accurate recognition of segmented characters is important for the recognition of word based on segmentation [5]. Feature extraction is most difficult part in HCR system.

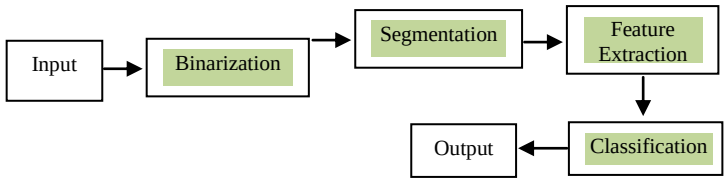

Figure 1: Block diagram of HCR system.

But before recognition, the handwritten characters have to be processed to make them suitable for recognition. Here, we consider the processing of entire document containing multiple lines and many characters in each line. Our aim is to recognize characters from the entire document. The handwritten document has to be free from noise, skewness, etc. The lines and words have to be segmented. The characters of any word have to be free from any slant angle so that the characters can be separated for recognition. By this assumption, we try to avoid a more difficult case of cursive writing. Segmentation of unconstrained handwritten text line is difficult because of inter-line distance variability, base-line skew variability, different font size and age of document [5]. During the next step of this process features are extracted from the segmented character. Feature extraction is a very important part in character recognition process. Extracted feature has been applied to classifiers which recognized character based on trained features. In second section, we have described feature extraction method in brief and described principal component analysis method. In the next session we have discussed neural network and SVM and Euclidean distance methodology.

2 FEATURE EXTRCTION

Any given image can be decomposed into several features. The term 'feature' refers to similar characteristics. Therefore, the main objective of a feature extraction technique is to accurately retrieve these features. The term "feature extraction" can thus be taken to include a very broad range of techniques and processes to the generation, update and maintenance of discrete feature objects or images [6]. Feature extraction is the most difficult part in HCR system. This approach gives the recognizer more control over the properties used in identification. Character classification task recognizes the character which is compared with the standard value that comes out the learning character, and the character should be corresponded to the document image that is matching a setting document style in the document style setting part. Here we have investigated and developed PCA based feature extraction method.

A. Principal component analysis

PCA is a useful statistical technique that has found application in fields such as face recognition and image compression, and is a common technique for finding patterns in data of high dimension [7]. It is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Since patterns in data can be hard to find in data of high dimension, where the graphical representation is not available, PCA is a powerful tool for analyzing data [7].

The other main advantage of PCA is that once you have found these patterns in the data, and you compress the data, i.e. by reducing the number of dimensions, without much loss of information [7].

Principal component analysis (PCA) is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has as high a variance as possible (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to (uncorrelated with) the preceding components. Principal components are guaranteed to

be independent only if the data set is jointly normally distributed. Before starting methodology first of all it's important to discuss following term which are related to PCA [7].

B. Eigenvector and Eigen values

The eigenvectors of a square matrix are the non-zero vectors that, after being multiplied by the matrix, remain proportional to the original vector (i.e., change only in magnitude, not in direction). For each eigenvector, the corresponding eigenvalues is the factor by which the eigenvector changes when multiplied by the matrix. Another property of eigenvectors is that even if we scale the vector by some amount before we multiply it, we still get the same multiple of it as a result. Another important thing to known is that when mathematicians find eigenvectors, they like to find the eigenvectors whose length is exactly one. This is because, as we know, the length of a vector doesn't affect whether it's an eigenvector or not, whereas the direction does. So, in order to keep eigenvectors standard, whenever we find an eigenvector we usually scale it to make it have a length of 1, so that all eigenvectors have the same length [7].

Steps for generating principle components of character and digit images:

Step 1: Get some data and find mean of each data

In this work we have used our own made-up data set. Data set is nothing but handwritten character A-J and 1-5 digits which contains 30 samples of each character or digit. And find the mean using equation 5.

$$M = \frac{1}{N}\sum_{k=1}^{n} X^{k} \qquad (1)$$

Where, M=Mean, N=Total no. of i/p images, X= I/p image

Step 2: Subtract the mean

For PCA to work properly, we have subtracted the mean from each of the data dimensions. The mean subtracted is the average across each dimension (use equation 2), where M is a mean which we have calculated using equation 1.So, all the X values have $\overline{X}$ (the mean of the x values of all the data points) subtracted, and all the Y values have $\overline{Y}$ subtracted from them. This produces a data set whose mean is zero.

$$X^{n} - M \qquad (2)$$

Step 3: Calculate the covariance matrix

Next step is to find out covariance matrix using equation 3.

$$M = \frac{1}{N}\sum_{k=1}^{n}\left(X^{k} - M\right)\left(X^{k} - M\right)^{T} \qquad (3)$$

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

Since the covariance matrix is squared, we have calculated the eigenvectors and eigenvalues for this matrix. By this process of taking the eigenvectors of the covariance matrix, we have been able to extract lines that characterize the data. The rest of the steps involve transforming the data so that it is expressed in terms of them lines.

Step 5: Choosing components and forming a feature vector

Here is where the notion of data compression and reduced dimensionality comes into it. In general, once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalues, highest to lowest. This gave us the components in order of significance. What needs to be done now is you need to form a feature vector, which is just a fancy name for a matrix of vectors. This is constructed by taking the eigenvectors that you want to keep from the list of eigenvectors, and forming a matrix with these eigenvectors in the columns.

Step 6: Deriving the new data set

This final step in PCA, and is also the easiest. Once we have chosen the components (eigenvectors) that we wish to keep in our data and formed a feature vector, we have simply took the transpose of the vector and multiply it on the left of the original data set, transposed.

3 CLASSIFICATION METHODS

A. Neural Network

Artificial neural networks (ANN) provide the powerful simulation of the information processing and widely used in patter recognition application. The most commonly used neural network is a multilayer feed forward network which focus an

input layer of nodes onto output layer through a number of hidden layers. In such networks, a back propagation algorithm is usually used as training algorithm for adjusting weights [9]. The back propagation model or multi-layer perceptron is a neural network that utilizes a supervised learning technique. Typically there are one or more layers of hidden nodes between the input and output nodes. Besides, a single network can be trained to reproduce all the visual parameters as well as many networks can be trained so that each network estimates a single visual parameter. Many parameters, such as training data, transfer function, topology, learning algorithm, weights and others can be controlled in the neural network [9].

B. Support Vector Machine

The main purpose of any machine learning technique is to achieve best generalization performance, given a specific amount of time and finite amount training data, by striking a balance between the goodness of fit attained on a given training dataset and the ability of the machine to achieve error-free recognition on other datasets [10].

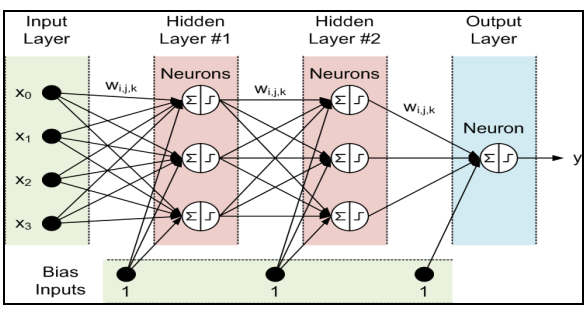

Figure 2: neural network design

With this concept as the basis, support vector machines have proved to achieve good generalization performance with no prior knowledge of the data. The main goal of an SVM [10] is to map the input data onto a higher dimensional feature space nonlinearly related to the input space and determine a separating hyper plane with maximum margin between the two classes in the feature space.

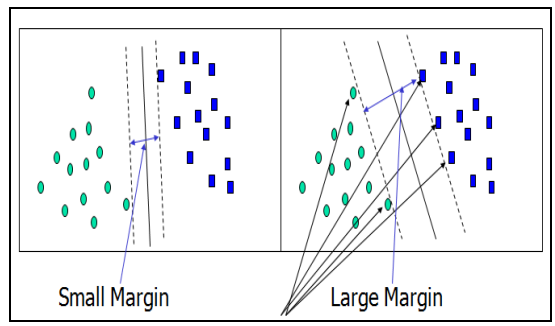

Figure 3: SVM margin and support vectors [10]

Main task of SVM is to finds this hyper plane using support vectors ("essential" training tuples) and margins (defined by the support vectors). Let data D be (Z_1, y_1), …, $(Z_{|D|}, y_{|D|})$, where X_i is the set of training tuples associated with the class labels y_i which has either +1 or -1 value [11]. There are uncountable (infinite) lines (hyper planes) separating the two classes but we want to find the best one (the one that minimizes classification error on unseen data). SVM searches for the hyper plane with the largest margin, i.e., Maximum Marginal Hyper plane (MMH) [11]. The basic concept of SVM can be summarized as,

- A separating hyper plane can be written as [11]
  $$X * Z + C = 0 \qquad (4)$$
  Where X= $\{X_1, X_2 ..... X_n\}$ is a weight vector and c a scalar (bias).
- For 2-D it can be written as [11]
  $X_0 + X_1 Z_1 + X_2 Z_2 = 0$, where $X_0 = c$ is additional weight.
- The hyper plane defining the sides of the margin:

$H1: X_0 + X_1 Z_1 + X_2 Z_2 \geq 1$, for $Y_i = +1$ and

$H1: X_0 + X_1 Z_1 + X_2 Z_2 \leq 1$, for $Y_i = -1$

- Any training tuples that fall on hyper planes H_1 or H_2 (i.e., the sides defining the margin) are support vectors [11].
- If data were 3-D (i.e., with three attributes), then we have to find the best separating plane.

After we got a trained support vector machine, we use it to classify test (new) tuples. Based on Lagrangian [11] formulation, the MMH can be rewritten as the decision boundary.

$$D(ZT) = \sum_{i=1}^{L} Y_i \alpha_i Z_i Z^T + C_0 \qquad (5)$$

Where, Y_i is the class label of support vector Z_i, Z^T is a test tuples, α_i is Lagrangian multiplier, L is the number of support vectors.

C. Euclidean distance.

Euclidean distance is most popular technique for finding the distance between to matrices or images. Let X, Y be two nXm images, $X = (X_1 X_2 .....X_{mn}), Y = (Y_1 Y_2 .....Y_{mn})$.

Euclidian distance between X and Y is given by

$$d^2(X,Y) = \sum_{k=1}^{mn} (X^k - Y^k)^2 \qquad (6)$$

4 EXPERIMENT AND RESULTS

In this work the PCA method as discussed in section II was implemented in Matlab environment. The extracted data is used as features for two classifiers, namely, neural network and support vector machine. We have prepared a real-time dataset comprising of A to J characters and digit 1 to 5. The data set was prepared by taking handwritings of different persons in a specific format. We have taken 30 samples of each character and digit, so finally our dataset contains total 450 samples for characters A to J and digits 1 to 5. We have applied PCA method on this database and prepared feature matrix PC_A. At the other side for testing purpose, we have taken 30 different images. Binarization, segmentation is applied one by one on input image. Same feature matrix PC_B is prepared for all the segmented characters.

A. Implementation Results of ANN & PCA based character recognition

Prepared PC_A matrix is given as an input to the neural network for training purpose. Similarly PC_B matrix is given to this trained network for testing purpose. The overall accuracy of 85% was obtained for the test data using ANN.

B. Implementation Results of SVM & PCA based character recognition

Similarly as we have described above, PC_A matrix is given as an input to the SVM for training purpose. Similarly PC_B matrix is given to this trained network for testing purpose. We have used libsvm package [12] for the classification purpose. The overall accuracy of 92% was obtained for the test data using SVM.

C. Implementation Results of Euclidean distance & PCA based character recognition

In this method for recognition purpose we have found the Euclidean distance between PC_A and PC_B and found the minimum index and based on this index we have found which character is recognized. PC_A and PC_B prepared using steps that we have discussed in previous section. We have measured over all accuracy of this method is 90%.

D. Comparison of Recognition using ANN, SVM classifiers and Euclidean distance.

In table 1 we have listed different methods and accuracy. As shown in table we can easily say that overall accuracy of PCA (SVM) is good compare to PCA (NN) and PCA (Euclidean distance) method. If we compare these methods on basis of training time then also SVM methods required less time compare to neural network and Euclidean distance. But drawback of SVM methods is we have to generate SVM format training and testing files, while in case of other methods it's not required. Now if we compare individual character accuracy then also PCA (SVM) gives good result compare to other method.

Table 1: Comparison of Overall Accuracy

Sr.no	Method	Structure/Parameter	Accuracy
1	PCA(Neural Network)	[25 30 6 25]	85%
2	PCA (SVM)	Kernel-RBF (Redial Bias Function) Cost-1 Gamma-1	**92%**
3	PCA (Euclidean distance)	-	90%

Table2: Comparison of Individual Character Accuracy

Sr.no	Letter or Digit	Accuracy Of PCA-SVM (%)	Accuracy Of PCA-ANN (%)	Accuracy Of PCA- Euclidean Distance (%)
1	A	96	80	98
2	B	99	80	98
3	C	99	100	96
4	D	95	70	96
5	E	96	80	95
6	F	97	80	95
7	G	96	90	95
8	H	95	80	98
9	I	98	75	96
10	J	97	80	96
11	1	97	80	95
12	2	96	90	95
13	3	95	80	95
14	4	99	80	98
15	5	96	80	95

5 CONCLUSION

A simple and an efficient off-line handwritten character recognition system using a new type of feature extraction, namely, PCA is investigated. Selection of feature extraction method is most important factor for achieving high recognition ratio. In this work, we have implemented PCA based feature extraction method. With the use of this obtained feature, we have trained the neural network as well as SVM to recognition character. We have also implemented character recognition with PCA and euclidean distance. In the investigated work all three method showed the overall recognition of 85% for PCA based neural network, 92% for PCA based SVM and 90% for PCA with Euclidean distance.

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", vol. 23, no. 7, pp. 671- 683, 1990.

[3] H.Fujisawa, Y.Nakano and K.Kurino, "Segmentation methods for character recognition from segmentation to document structure analysis". Proceeding of the IEEE, vol.80, and pp.1079-1092. 1992.

[4] Ravi K Sheth, N.C.Chauhan, Mahesh M Goyani," A Handwritten Character Recognition Systems using Correlation Coefficient", V V P Rajkot, 8-9 April 2011,ISBN NO: 978-81-906377-5-6, pp 395-398..

[5] Pal, U. and B.B. Chaudhuri, "Indian script character recognition: A survey," Pattern Recognition", vol. 37, no.9, pp. 1887-1899, 2004.

[6] Ravi K Sheth, N C Chauhan, M G Goyni, Kinjal A Mehta," Chain code based Handwritten character recognition system using neural network and SVM", ICRTITCS-11, 9-10 December, Mumbai.

[7] Lindsay I Smith," A tutorial on Principal Components Analysis", February 26, 2002

Sophiayati Yuhaniz "The Heuristic Extraction

[8] Dewi Nasien, Habibollah Haron, Siti Algorithms for Freeman Chain Code of Handwritten Character", International Journal of Experimental Algorithms, (IJEA), Volume (1): Issue (1)

[9] S. Arora" Features Combined in a MLP-based System to Recognize Handwritten Devnagari Character", Journal of Information Hiding and Multimedia Signal Processing, Volume 2, Number 1, January 2011

[10] H. Izakian, S. A. Monadjemi, B. Tork Ladani, and K. Zamanifar "Multi-Font Farsi/Arabic Isolated Character Recognition Using Chain Codes", World Academy of Science, Engineering and Technology 43 2008

[11] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery", 1998, pp 121-167.

[12] Jiawei Han and Micheline Kamber "Data Mining Concepts and Techniques", 2nd Edi, MK publication, 2006, pp 337-343

[13] Chih-Jen Lin,"A Library for Support Vector Machines", http://www.csie.ntu.edu.tw/~cjlin/libsvm/