International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 1, January- 2013

A Hybrid Approach To Optical Numeral Recognition (Composition Of PCA & LVQ)

Nishant Gupta[1], Arun Sharma[2]

Dept. Of Computer Science, MGM CoET, Noida, India[1], Dept. Of Computer Science, KIET, Ghaziabad, India[2]

Abstract

The aim of present work is to recognize handwritten digits written by different persons. All the images for digits from "0" to "9" will be collected. Then these digits will be scanned using a scanner and stored for further processing. After this, various pre-processing techniques will be applied on these scanned digits in order to remove various artefacts like size variations etc. Then the Features will be extracted from these processed images and the PCA & LVQ will be used in order to classify various digits to facilitate the recognition and decision making process. The idea of the proposed technique is to use the PCA to produce a coefficient vector of the digit images, which will be directly used to recognize characters. For the classification of numerals we used Learning Vector Quantization.

1. INTRODUCTION

The system for recognizing handwritten numerals is used for converting manual systems to automated or mechanized systems. It can be used to reduce the human intervention, so as to increase the speed of jobs. Recognition system has many applications in different areas, such as industries, forensic, security based System, forecasting, inventory control and other fields. An effort has been made to explore the basic concepts about how the handwritten digits can be converted into suitable form for recognition purposes. Out of two approaches on-line and off-line, only off-line has been considered. The two main stages in most of the OCR systems are features extraction and classification. The most popular classification methods used for OCR systems are generaly Artificial Neural Network and Hidden Markow Model.. We proposed a method that relies on Principal Component Analysis (PCA) for optical numeral recognition. The idea of the proposed technique is to use the PCA to produce a coefficient vector of the digit images, which will be directly used to recognize the digits. For the classification of numerals we used Learning Vector Quantization.

One should know about pattern and pattern recognition which is the base for optical numeral recognition. Pattern is an object that has to be matched with an existing object or thing. For example, a written digit/word, a fingerprint image, a human face etc. Some of the examples are shown

in Fig.1. Procedure used to recognize a particular pattern is called pattern recognition.

Figure 1 Examples of Pattern

2. METHODOLOGY

The biggest challenge of character recognition is how to understand the concept of a character's shape and the mechanism that identifies any instantiation of this concept. For the handwritten numerals, the nature of the digit which varies from one language to another, the variation in the numeral shape when it is written by different writers (sometimes even by the same writer) and the noise such as stains, dots, and gaps are the main difficulties that the recognition task faces. The proposed system is mainly divided into three modules: Features Extraction, Classification and Recognition. Basically Recognition is the combination of previous two modules.

2.1General Architecture

The general architecture of a hand-printed digit recognition system can be given in the Fig.2.

The very first step while recognition is to pre-process the incoming digits. The pre-processing of the digits is consists of Normalization and Binarization. For the feature extraction and classification of handwritten digits

we used Principal Component Analysis and Learning Vector Quantization.

In the first module of the proposed system we first read the image, store it in matrix form. Then, after using conversion functions we convert the image into spatial -

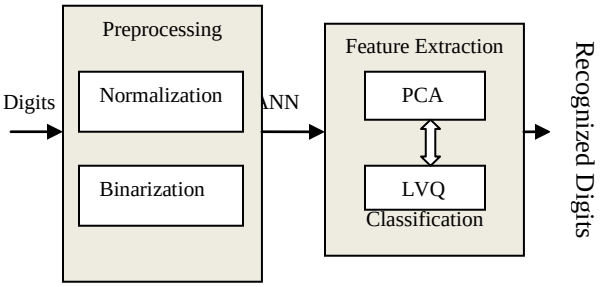

Figure 2 The General Architecture of Handwritten Digit Recognition

form from RGB to GREY conversion and then reconvert it to binary image (Black and White). Since in the module of feature extraction our aim is mainly focusing on foreground and background object, binary images are more suitable for that. After extracting the object area from background image we apply edge detection filter to highlight the edges of handwritten digits. Once we highlight the numerals, then by using Principle Components of the image matrix like Centroid, Area, Eigen Vector, Mean Value etc, we can easily classify the different digits in different classes

The input data that we take always have some amount of noise, and the process of pre-processing is needed to reduce its effect. Pre-processing involves basic image processing steps such as storing the image in the matrix form, converting image into spatial form from RGB to GREY level and reconverting it to binary image (Black and White). After extracting the object area from background image we apply edge detection filter to highlight the edges of handwritten digits. The various steps which are involved in the image pre-processing in the proposed work can be summarized as follows:

1. Reading image and storing it in a matrix form of 64x64.
2. Convert image to Gray level image.
3. Convert Gray level image to Binary Image i.e. in black and white based on Global threshold.
4. Invert image to get the digit in white over a black canvas.
5. Find out edges in intensity image. Edge takes intensity or a binary image as its input, and returns a binary image BW of the same size, with 1's where the function finds edges in the image and 0's elsewhere.
6. Dilate images and fill image regions and holes
7. Resize the image in the bounding box to 64x64.

Thresholding which is a low level image processing technique used for document analysis step for further processing, for obtaining the binary image from its gray scale one [1].It converts grey-level images to binary images by making all pixels to zero that are below some threshold and the pixels above that threshold to one. If g(x, y) is a thresholded version of f(x, y) at some global threshold T[2],

$$F(x) = \begin{cases} 1, & \text{if } f(\&x, y) \geq T \\ 0, & otherwise \end{cases} \quad ...1$$

The conversion for a digit two from gray level to black and white image is shown in Fig.3

Figure 3 Conversion from Gray level to Black & White

An image to find out edge detection from black and white image is shown in Fig.4.

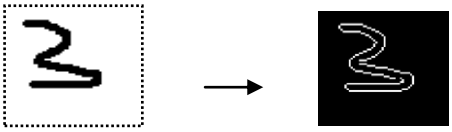

Figure 4 Edge Detection

After finding out the boundary of the digit/image, we will fill the holes in the binary image..Fig.5 shows Image filling of digit three.

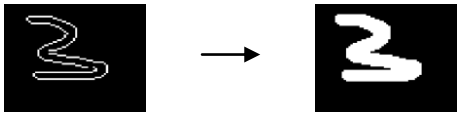

Figure 5 Image Filling

Pre-processing can be broadly seen as a basic image processing steps as shown in the block diagram of Fig. 6

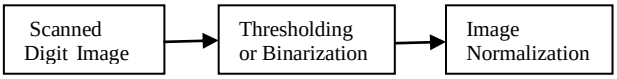

Figure 6 Block Diagram of pre-processing steps

After the pre-processing steps, the two techniques PCA and LVQ can be used for recognition of the handwritten digits.

3. Principal Component Analysis

It is a procedure which is used to convert a set of observations of possibly similar variables into a set of values of linearly uncorrelated variables. This is called principal components. It helps in reducing number of dimensions in the data. Mane and et al[3] proposed a

work for the recognition of alphanumeric characters. They proposed the recognition technique called elastic matching using PCA for off line isolated uppercase letters and Devnagari handwritten characters.The elatic matching gives the displacement vector of 800 x 1 for 20 x 20 images.The following general steps are used in Principal Component Analysis [4].

1. Arrange the data set

Lets say we have a data consist of a set of observations of M variables, and we wish to reduce the data such that each observation can be defined with only P variables, P < M. Suppose further, that the data are arranged as a set of N data vectors $X_1 \ldots X_N$ with each X_N representing a single grouped observation of the M variables.

a. Write $X_1 \ldots X_N$ as column vectors, each of which has M rows.

b. Put the column vectors into a matrix X of dimensions M × N.

2. Calculate the mean

a. Find the mean along each dimension m = 1... M.

b. Put the mean values to an mean vector u of dimensions M × 1.

$$u[m] = \frac{1}{N}\sum_{n=1}^{N} X[m,n] \quad \ldots 2$$

3. Find the covariance matrix

Find the M × M covariance matrix C from the outer product of matrix B with itself.

$$C_{mxm} = (C_{i,j} \,.\, C_{i,j} = \text{cov}\,(\text{Dim}_i, \text{Dim}_j) \quad \ldots 3$$

where Cmxm is a matrix with m rows and n columns, and Dimx is the xth dimension.

4. Find the eigenvectors and eigenvalues of the covariance matrix

a. Calculate the matrix V of eigenvectors

$$V^{-1}CV = D \quad \ldots 4$$

b. Matrix D will now be in the form of an M × M diagonal matrix, where

$$D[p,q] = \lambda m \quad \text{for } p=q=m \quad \ldots 5$$

is the Mth eigenvalue of the covariance matrix C, and

$$D[p,q] = 0 \quad \text{for } p \neq q \quad \ldots 6$$

where D is the diagonal matrix of eigenvalues of C.

c. Now the Matrix V which is of dimension M × M, comprises of M column vectors which is of length M, represent the M eigenvectors of the covariance matrix C.

5. Select components and form a feature vector

The eigenvector that have the highest eigenvalue is the component of the data set. When the eigenvectors were found from the covariance matrix, then the order of highest to lowest of eigenvalue has to be maintain. This gives you the components in order of significance. If initially we have n dimensions in our data, and then we calculate n eigenvectors and eigenvalues, and we will keep only the first p eigenvectors.The final data set has only p dimensions.

Now we will form a Feature vector which is another name for a matrix of vector.

Feature Vector= (eig1 eig2 eig3 ……eign)

6. Deriving the new data set

This is the final step of PCA. After choosing the components (eigenvectors) that we will keep in our data and formed a feature vector, we take the transpose and multiply it on the left hand side of the original data set, transposed.

Data for further use = Row Feature Vector × Row Adjust Data

where Row Feature Vector is the matrix with the eigenvectors in the columns transposed so that the eigenvectors are in the rows, with the most considerable eigenvector at the top, and Row Adjust Data is the mean-adjusted data transposed, i.e. the data items are in each column, with each row holding a separate dimension.

4. Learning Vector Quantization

The research problem associated with the classification of a large set of data into a predefined set of classes is a long term process. The study shows that the classification system has a trade-off between accuracy and complexity. The purpose of the Learning Vector Quantization (LVQ) network is generally used for categorization of the text documents. In the LVQ method, each class of the data set is represented by a small number of codebook vectors. The decision boundaries are defined by the nearest neighbour rule for the codebook vectors which are positioned in the feature space. It need less training set of examples and is comparable and faster than other classification techniques.

Learning vector Quantization (LVQ) is a neural network that combines competitive learning with supervision[5]. It can be used for pattern classification. A training set consisting of Q training vector - target output pairs are assumed to be given

$$\{s(q) : t(q)\}, \quad q = 1, 2, \ldots, Q$$

where s(q) are N dimensional training vectors, and t(q) are M dimensional target output vectors. M is the number

of classes, and it must be smaller than Q. The target vectors are defined by

$$f(x) = \begin{cases} 1, & if\ s(q) belongs\ to\ class\ i \\ 0, & otherwise \end{cases}$$

The Purpose is to place the hidden units such that it cover the decision regions of the training set. Tin Kam Ho [6] had worked on learning vector quantization and showed its application in recognizing handwritten digits. Zanona and Zaghmouri[7] proposed a work on recognition of Handwritten numbers in the form of hindi numbers.They used the process of LVQ and morophogical approach was used for segmentation. As already discussed LVQ networks are similar to self-organizing map's (SOM) except that the single layer of neurodes uses target output vectors t_q correspond with the input exemplars x_q; that is trains in the supervised mode rather than in the unsupervised mode. Thus the target (identifier) vectors must be available to determine if the winner is correct. The learning vector quantization algorithm was adopted by Kohonen [5] for pattern recognition. When a feature vector x is presented to the network, the values y_m are computed. From 1,..., M of these M neurodal outputs, only a single one puts out a high value to denote class m* as the class to which the input vector x belongs. The winner is reinforced, provided that it is correct upon testing against the targets. Non winners are extinguished. The following Fig.7 displays such a network

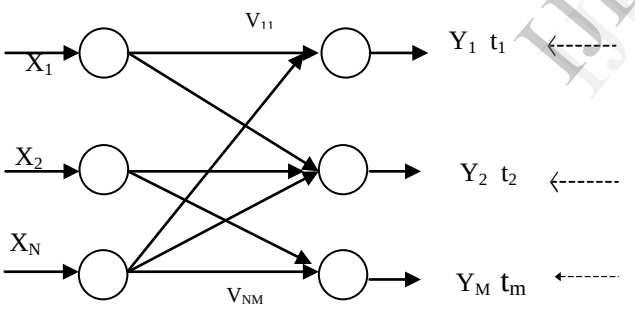

Figure 7 A learning vector quantization network

5. RECOGNITION APPROACH

The main problem which arises during image recognition is determining the distance between images. In perspective of image recognition, there are many image distances which are suffering from some of the following disadvantages:

1. It is complicated to combine the metric with most of the image recognition techniques such as SVM, LDA, PCA, etc.
2. The computation of the measure is difficult.
3. Sometimes two well dissimilar images can be both similar to an unknown object as the distance does not follow the triangle inequality.

Amongst all the image metrics, Euclidean distance is mostly used due to its ease of use. Let *x, y* be two *M* by *N* images, $x = (x^1, x^2 \ldots x^{MN})$, $y = (y^1, y^2, \ldots, y^{MN})$, where x^{kN+l}, y^{kN+l} are the gray levels at location (k,l). The Euclidean distance $d_{E(}x, y)$ is given by

$$d_E^2(x,y) = \sum_{k=1}^{MN}(x^k - y^k)^2 \qquad \ldots 7$$

With the use of Eucledian distance method [8] we can easily calculate minimum distance between the input pattern weight and the training weight. This method is used to finally recognize the digits. The whole process can be summarized in Fig 8.

The basic process which is used to recognize the digits is to make the labels. At every first label the digit one is stored. So after every 10^{th} column, the digit of one will be found. Consecutively at second position the digit two will be placed, the same procedure will be followed for others digits. We will make a matrix of 10 x 10 where the diagonal will shows the value 1.At each column we will be having the value 1.For the first column the value one will be at row 1, which signifies the digit one. Following the same procedure for other digits, the next sequence of the digits from 0 to 9 will start from the position of column 11 to 20.

This data is stored and the same positions of same digits are stored at one place, which is used for the recognition purpose by ANN.

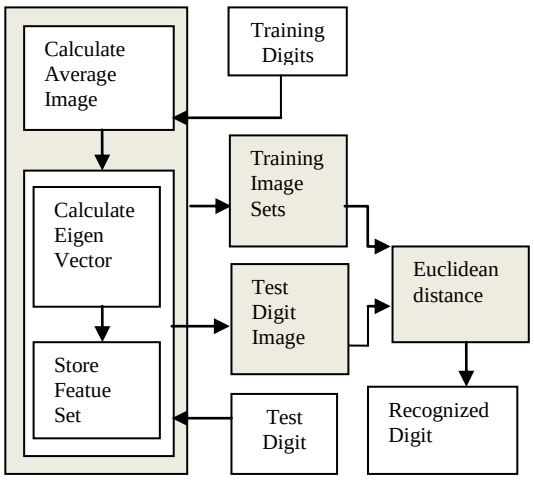

Figure 8 Block diagram showing recognition using Euclidean distance

6. SIMULATION

One of the characteristic of neural networks is that they are not programmed rather they are trained [9]Thus we have trained the neural network with the help of the digits available in the database. Some training inputs are provided to the network using which they generalize and adapt the trends. After which they are able to use their knowledge in giving the output when certain inputs are given. Thus the data after pre-processing is used for simulating the network.

Neural network used: Learning Vector Quantization
No. of hidden neurons: 60
Learning rate: 0.01

The number of neurons in the hidden layer was taken to be as 60 and the network was trained for 100 epochs.

7. NETWORK TRAINING

Network model is trained with the total training digits of 200 from Zero to Nine for which the test data has to be recognized. The size of training data is 200. Twenty images of each digit are trained and fifty images of digits from zero to nine are used for testing. Thus we have total of 200 combinations of digit images to train and 50 images for testing the network.

Therefore, Training set: 200
Testing set: 50

Thus we will be training the network using 250 data value after which 50 values will be used to test the network. NN Tool was used for training of the network. This tool provided with the training and testing data. The following Figures 3.14 shows the performance graphs for various digits during training of neural network via nn tool. The Training confusion matrix for the available data is shown in figure 3.15

8. RESULTS AND DISCUSSIONS

The database is made of 250 sample digits, out of which 50 samples were for cross-validation. Fig. 4.1 and Fig. 4.2 shows some of the handwritten digits form the training set and the test set.

The main concern of character recognition is how to understand the concept of a character's shape and the process that identifies any instantiation of this concept. For the handwritten numerals, the nature of the digit which varies from one language to another, the variation in the numeral shape when it is written by different writers (sometimes even by the same writer) and the noise such as stains, dots, and gaps are the main difficulties that the recognition task faces.

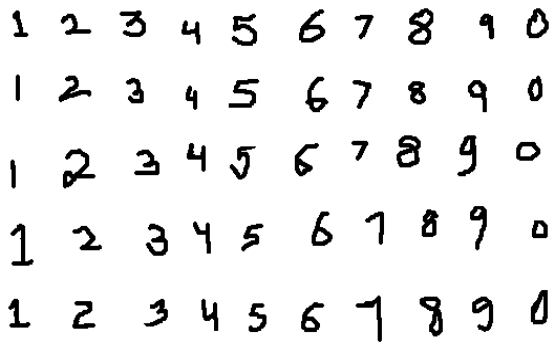

Figure 9 Sample Digits from handwritten training set

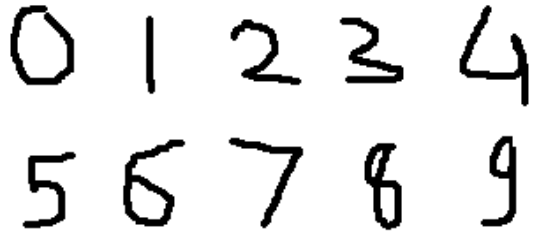

Figure 10 Handwritten digit from Test Set

The original data base contained 250 digits. Then this data base was split into two sets: 200 digits were placed in the training set and the next 50 set samples was placed in the test set. Thus we had two set of total 250 samples. All the digits were then size normalized to fit in 64 x 64 pixel block in order to remove the variations in the height and width of different digits.

A. Principal Component Analysis

Principal component analysis was tested on the collected database. After preprocessing of the training and test digits, the digits are trained to classify the digits. For recognition the distance between the input pattern weight and the training weight is calculated with the help of Euclidean distance method. The minimum distance in between the training weight and the input pattern weight will recognize the test digit. An accuracy of 95.00% on training set and 94% on test data was achieved by Principal Component Analysis. The overall accuracy of the method on the whole database was found to be 94.50%. The success rate on training and testing set and overall success rate of PCA have been show in Table 1.

B. Learning Vector Quantization network

Learning vector quantization network was tested on the collected database. Several experiments with different number of hidden neurons and with 100 epochs were conducted in order to train the learning vector quantization network. The numbers of hidden neurons

were taken to be 60 and the network was trained for 100 epochs. The disadvantage of this method is that it takes considerable amount of training time as compared to Principal Component Analysis but the most important advantage of this method is that the accuracy may increase with the increase in the size of the training set. An accuracy of 95.50% and 92.00% was achieved by learning vector quantization (LVQ) network on the training and the test set. The overall accuracy of the method on the whole database was found to be 92.25%. The success rate on training and testing set and overall success rate of LVQ have been show in Table 1.

Table 1 Success rate and Overall Success Rate

Technique	Success rate on training set	Success rate on test set	Overall success rate
PCA	95.00%	94.00%	94.50%
LVQ	95.50%	92.00%	93.75%

9. Discussions

The performance of our classifiers is shown in Fig. 4.3-4.10. Fig 4.3-4.10 shows the success and the error rates of different classifiers on the training set and the test set. Although both the classifiers did well on both the data sets but LVQ algorithm was found to give 92.00% accuracy on the test set and an accuracy of 95.50% on the training set. Table 4.3 depicts the classification time taken by the two classifiers. Here we see that there is enormous variation in the speed. PCA classifier takes only 0.0575 seconds and learning vector quantization (LVQ) network takes 1.0295 seconds.

The learning vector quantization network takes longer time during learning stage but after the training is complete its takes very less time to classify the input test vector. One of its advantages is that its performance becomes more striking as the training database continues to increase in size. In the classification of whole database the learning vector quantization network is found to give an accuracy of 93.75 % where as Principal Component Analysis algorithm gave an accuracy of 94.50 %

10. CONCLUSION

The work has allowed us to obtain an understanding of the problem of handwritten digit recognition. Besides, we have seen that there are a lot of various methods to recognize handwritten characters and to transform them into digital data. We also managed to program two recognition methods from the database collected. The first method, Principal Component Analysis is rather simple, but the recognition results are comparable with other classifiers. However, the LVQ network offers better performance during training as far as recognition rate is concerned but takes long during learning. To conclude, our work approaches the problem as it is approached in real applications.

REFERENCES

[1] Mehmet Sezgin, Bulent Sankur (2004), "Survey over image thresholding techniques and quantitative performance evaluation". Journal of electronic imaging 13(1), page(s): 146-165.
[2] Xu Ye, Zhang Wei (2010), "On a clustering method for Handwritten Digit Recognition". Third International Conference on Intelligent Networks and Intelligent Systems (ICINIS) IEEE, page(s):112-115
[3] Vanita Mane, Lena Ragha(2009), "Handwritten Character Recognition using Elastic Matching and PCA". International Conference on Advances in Computing, Communication and Control (ICAC3), page(s):410-415
[4] http://en.wikipedia.org/wiki/principal component analysis
[5] Kohonen, T. (1997), Self-Organizing Maps. 2nd ed, Springer-Verlag, Berlin.
[6] Tin Kam Ho (1993), "Recognition of handwritten digits by combining independent learning vector quantization". Document analysis and recognition, proceedings of the second international conference, page(s): 818-821.
[7] Marwan A. Abu-Zanona, Bassam M. El-Zaghmouri(2012), "Current Arabic (Hindi) Hand Written Numbers Segmentaion and Recognition". Journal of Emerging Trends in Computing and Information Sciences, VOL. 3, NO. 6, page(s): 936-941
[8] http://www.cis.pku.edu.cn/faculty/vision/wangliwei/pdf/IMED .pdf
[9] www.mathworks.in,MATLAB/ Neural Network Toolbox.