ACCURACY IMPROVEMENT IN ODIA ZIP CODE RECOGNITION TECHNIQUE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology
In
Electronics and Instrumentation Engineering

BY

DEBESH KUANR

108EI016

LOKANATH TRIPATHY

108EI030

Department of Electronics & Communication Engineering
National Institute of Technology
Rourkela – 769008.

2012

ACCURACY IMPROVEMENT IN ODIA ZIP CODE RECOGNITION TECHNIQUE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology
In
Electronics and Instrumentation Engineering

BY

DEBESH KUANR

108EI016

LOKANATH TRIPATHY

108EI030

Under the guidance of

Prof. U.C. Pati

Department of Electronics & Communication Engineering
National Institute of Technology
Rourkela – 769008.
2012

NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA

CERTIFICATE

This is to certify that the Thesis Report entitled "**ACCURACY IMPROVEMENT IN ODIA ZIP CODE RECOGNITION TECHNIQUE**" submitted by Mr. Debesh Kuanr and Mr. Lokanath Tripathy in partial fulfillment of the requirements for the award of Bachelor of Technology degree Electronics and Instrumentation Engineering during session 2008-2012 at National Institute of Technology, Rourkela (Deemed University) and is an authentic work by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other university/institute for the award of any Degree or Diploma.

Prof. **U. C. Pati**
Dept. of E.C.E
National Institute of Technology
Rourkela-769008
Email:ucpati@nitrkl.ac

Date:

ACKNOWLEDGEMENT

First of all, we would like to express our deep sense of respect and gratitude towards our advisor and guide **Prof U.C. Pati**, who has been the guiding force behind this work. We are greatly indebted to him for his constant encouragement, invaluable advice and for propelling us further in every aspect of our academic life. His presence and optimism have provided an invaluable influence on our career and outlook for the future. We consider it our good fortune to have got an opportunity to work with such a wonderful person.

Next, we want to express our respects to **Prof. G. Panda** for teaching and also helping how to learn. He has been great sources of inspiration to us and we thank him from the bottom of our heart.

We would like to thank all faculty members and staff of the Department of Electronics and Communication Engineering, N.I.T. Rourkela for their generous help in various ways for the completion of this thesis.

We would like to thank all our friends and especially our classmates for all the thoughtful and mind stimulating discussions we had, which prompted us to think beyond the obvious. We have enjoyed their companionship so much during our stay at NIT, Rourkela.

ABSTRACT

Odia is a very popular language in India which is used by more than 45 million people worldwide, especially in the eastern region of India. The proposed recognition schemes for foreign languages such as Roman, Japanese, Chinese and Arabic can't be applied directly for odia language because of the different structure of odia script. Hence, this report deals with the recognition of odia numerals with taking care of the varying style of handwriting. The main purpose is to apply the recognition scheme for zip code extraction and number plate recognition. Here, two methods "gradient and curvature method" and "box-method approach" are used to calculate the features of the preprocessed scanned image document. Features from both the methods are used to train the artificial neural network by taking a large no of samples from each numeral. Enough testing samples are used and results from both the features are compared. Principal component analysis has been applied to reduce the dimension of the feature vector so as to help further processing. The features from box-method of an unknown numeral are correlated with that of the standard numerals. While using neural networks, the average recognition accuracy using gradient and curvature features and box-method features are found to be 93.2 and 88.1 respectively.

Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

INTRODUCTION

Handwritten character recognition has been a very popular area of research for the last several years. Lots of journals have been published in this area and still the research is going on. Though lots of research work has been done for the foreign languages like ‘Arabic’, ‘Chinese’, ‘Japanese’, Indian scripts have not been able to seek major attention. India is a multi-language country with twenty-two languages and eleven writing scripts. Out of them, only ‘Bengali’, ‘Devnagri’,’Urdu’ are the languages which are used for research purpose. In eastern India, Odia is a popular language which is used by a large population, so in this report we have tried to make an OCR system for Odia language which will be helpful in the areas of zip code extraction and number plate recognition. Optical character recognition (OCR) in which characters is automatically recognized after the document is being scanned and applied to the OCR system. In pattern recognition, OCR has become one of the most fascinating and challenging areas with various practical application potentials. It can contribute immensely to the advancement of an automation process and can improve the interface between man and machine in many applications.

1.1 CHARACTER RECOGNITION SYSTEMS

Character recognition systems are basically into two:

1.1.1 Off-line character recognition:

The system accepts image as input from the scanner, it is more difficult than on-line character recognition system because of unavailability of contextual information and prior knowledge the like text position, size of text, order of strokes, start point and stop point, further there are noises in images, while the noises in On-line character recognition[11] near to be absent. Eg., Machine Printed character recognition.

1.1.2 On-line character recognition:

The system accepts the moment of pen from the hardware such as graphic tablet, light pen and there is a lot of information during input process available such as current position, moment's direction, start points, stop points and stroke orders. E.g. Handwritten character recognition [12].

1.2 APPLICATION OF CHARACTER RECOGNITION

. Some practical application potentials of OCR system are: (1) reading aid for the blind, (2) automatic text entry into the computer for desktop publication, library cataloging, ledgering, etc. (3) automatic reading for sorting of postal mail, bank cheques and other documents, (4) document data compression: from document image to ASCII format, (5) language processing, (6) multi-media system design, etc.[1]

1.3 PREVIOUS WORK

The origin of character recognition was found in 1870 when Carey invented the retina scanner- an image transmission system using a mosaic of photocells. Later, in 1890, Nipkow invented the sequential scanner, which is a major breakthrough both for modern television and reading machines. However, character recognition was initially considered as an aid to the visually handicapped and the early successful attempts were made by the Russian scientist Tyurin in 1900. Depending on versatility, robustness and efficiency, the commercial OCR systems can be divided into four generations. The first generation systems can be characterized by the constrained letter shapes which the OCRs read. Such machines appeared in the beginning of the 1960s. The first widely commercialized OCR of this generation was the IBM 1418, which was designed to read a special IBM font, 407. The recognition method was logical template matching where the positional relationship was fully utilized. The next generation is characterized by the recognition capabilities of a set of regular machine printed characters as well as hand-printed characters. At the early stages, the scope was restricted to numerals only. Such machines appeared in the middle of 1960s to early 1970s. In this generation, the first and famous OCR system was IBM 1287, which was exhibited at the 1965 New York world fair. In terms of hardware configuration, the system was a hybrid one, combining analog and digital technology. The first automatic letter-sorting machine for postal code numbers of Toshiba was also developed during this period. The methods were based on the structural analysis approach. The third generation can be characterized by the OCR of poor print quality characters, and hand –printed characters for a large category character set. Commercial OCR systems with such capabilities appeared roughly during the decade 1975 to 1985. The fourth generation can be characterized by the OCR of complex documents intermixing with text, graphics, table and mathematical symbols, unconstrained hand written characters, color document, low-quality noisy documents like photocopy and fax, etc. Some pieces of work on complex documents provided good results. Although many pieces of work on unconstrained hand written character are available in the literature, the recognition accuracy hardly exceeds 85%. Very few studies on color documents have been published and research on this problem is continuing. Also, research

on noisy document is in progress. Among other commercial products, postal address readers are available in the market. In the United States, about 60% of the hand-printed is sorted automatically. Reading aid for the blind is also available. An integrated OCR with speech output system for the blind has been marketed by Xerox–Kurzweil for English language. At present, more sophisticated optical readers are available for Roman, Chinese, Japanese and Arabic text. These readers can process documents which has been typewritten, typeset, or printed by dot-matrix, line and laser printers. They can recognize characters with di8erent fonts and sizes as well as different formats including intermixed text and graphics. With the introduction of narrow range scanners, measuring 3 to 6 in wide, columnar scanning is now possible. With these scanners an optical reader can recognize multiple columns or sections of a page or mailing lists. Some are equipped with software for spell checking, and for flagging suspicious characters or words [1].

Many OCR schemes have used KNN and HMM based models, support vector machines, multi-layer perceptron networks, multi-layer cluster neural networks, fuzzy networks, Quadratic classifiers, linear classifiers, minimum distance classifiers to classify the characters. For feature extraction direction histogram method, stroke's directional features, gradient and curvature method etc. have been employed.

1.4 PROPOSED OCR

In our OCR system, the differing styles of handwritten characters have been taken into account and it has been tried to make the OCR independent of those variations. For those various preprocessing steps are done before the actual processing is carried out. These steps make the input scanned document suitable for processing and bring all the character images into the same platform so that same process of recognition scheme can be applied to all. The preprocessing steps include binarization, size normalization, slant correction, middling and thinning. Before preprocessing the scanned image is segmented to individual characters using horizontal and vertical segmentation. While carrying out the segmentation, the spaces between consecutive letters are counted in order to check if a letter is part of a word.

After preprocessing, the features of the characters are calculated. Features represent spatial characteristics by virtue of which we differentiate among the characters. For feature calculation we have used gradient and curvature method and box-method approach. In gradient and curvature method, two different features are found out corresponding to the gradient and curvature of the character. Finally, both the features are combined to form the final feature vector. For gradient features, Roberts filters are used whereas for curvature calculation bi-quadratic interpolation surfaces are used. The image was converted to grayscale image before this method was applied as gradient means the directional change in color or intensity values, for which , binary image can't be used. In the box method the image was divided into some blocks and features are calculated from each of the boxes by taking the vector distance of each black pixel into account. Finally, the features from all the boxes are combined to form the final feature vector.

After the feature vectors are formed, the pattern recognition part was done which involves artificial neural networks and correlation method. While using neural networks, a large no of samples are used to train the network and then tested with enough testing samples. Both the feature calculation methods are used here. In the correlation method, first the image matrices are directly correlated with the standard letters. And finally, box method features are used for correlation scheme. Finally the results from all the methods are compared and error in each case was found out. New approaches have been proposed which can help in reducing these errors.

1.5 FLOW CHART

The flow chart for any OCR system is as shown below.

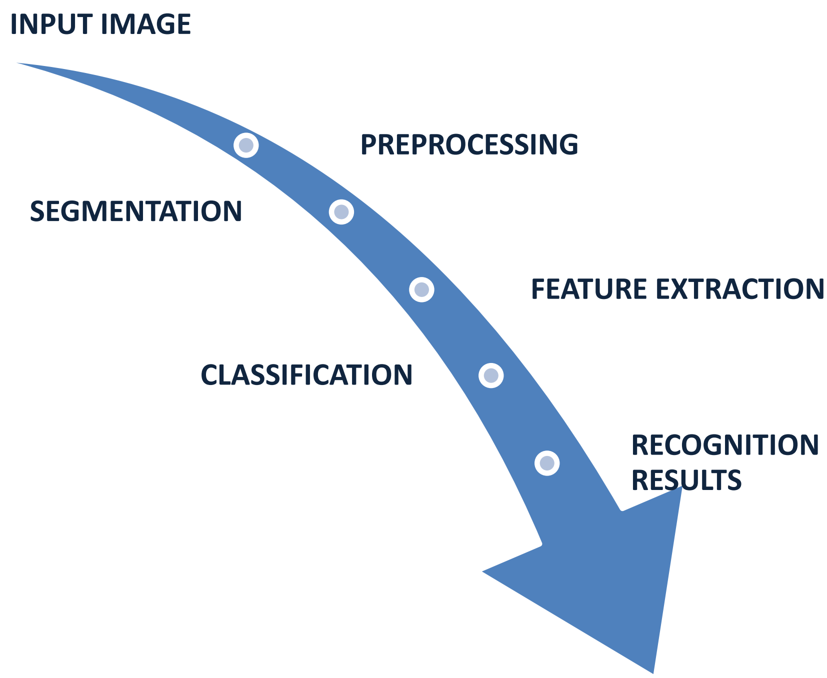

Fig 1.1 Flowchart of an OCR system

CHAPTER 2

SEGMENTATION & PREPROCESSING

SEGMENTATION & PREPROCESSING

2.1 SEGMENTATION

The obtained input image may contain a number of characters which are to be recognized. So, the first step would be to separate the characters to individual images so that the recognition scheme can be applied to each segmented character one by one. And finally, all the recognized characters are combined together so that the result can displayed as a single document in a notepad file. In order to differentiate between words and letters the spacing among the characters have been taken into account.

As here we are focusing on zip code or number-plate recognition, we can assume that the individual letters are not connected i.e. they don't touch or overlap. Also, we can consider the lines to be almost straight though variations are allowed unless they don't overlap.

A function named "clip" has been used each time the document or part of the document is processed, whose objective is to discard the extra regions of the image so that the rightmost and leftmost black pixel will lie on the last and first column. Similarly, the uppermost and lowermost black pixel will lie on the first and last row.

The segmentation of the document into individual letters is done through the following steps.

1. Horizontal segmentation
2. Vertical segmentation
3. Calculation of spaces among the letters

2.1.1 HORIZONTAL SEGMENTATION

Horizontal segmentation is done to separate the lines. The key behind this is to calculate the no of white pixels in each row and compare that with the total no of columns. As, the document is already binarized with white as the background, a row where no part of any letter will be present or the separating row will contain only white pixels.

Hence, the algorithm for separating the lines is as follows.

i. Apply the clip function.

ii. Traverse the rows of the image.

iii. Calculate the sum of all pixels in a row when it is traversed.

iv. Compare the sum with the total no of columns.

v. If the sum equals to the total no of columns, then store the image up to that row in a matrix and store the remaining part in another matrix for further processing. And , then go for cropping the letters in that line.

   Else, go to the next row and repeat the process.

vi. Stop if all the rows are processed.

2.1.2 VERTICAL SEGMENTATION

Vertical segmentation is done to separate the letters or numerals after a line has been separated. Similar methodology as vertical segmentation is followed but here instead of rows, columns are traversed as segmentation of letters will be done vertically.

To differentiate between a separate letter and a letter belonging to a word, the no of spaces between each consecutive letters are calculated. As we know, the spaces between two words will be much greater than the spaces in-between two letters. Hence we can assume that if the space is less than 30-60 % of the maximum space in a line, then it is a separate letter.

Hence, the algorithm for separating the letters is as follows.

i. Apply the clip function.
ii. Traverse the columns of the line.
iii. Calculate the sum of all pixels in a column when it is traversed.
iv. Compare the sum with the total no of rows.
v. If the sum equals to the total no of rows, then store the image up to that row in a matrix (which will represent the current letter being processed) and store the remaining part in another matrix (re1) for further segmentation.
vi. Apply the clip function to the remaining part and store it in another matrix (re2).
vii. Subtract the re2 from re1 to find the no of spaces.
viii. Store the space value in the space vector.
ix. Stop if all the columns are processed.

The current letter being processed is then compared with the standards by different methods as explained later.

2.2 PREPROCESSING

When the document is scanned, there is a great chance that it will contain noise. So, it is of prior importance to convert the image into suitable form and highlight certain features in the image. Generally this step uses techniques like binarization, size normalization, filtering with mean and median filters etc. Following functions are used here for preprocessing.

1. Binarization
2. Size Normalization
3. Slant Correction
4. Middling
5. Thinning

2.2.1 BINARIZATION

The obtained scanned image may contain various gray levels, even sometimes it can be a colored one. Hence, the prime objective would be to binarize the image by using a thresholding function so that, we will be left with an image with only two gray levels (0 or 1). Hence, now it will be easier to highlight the region of interest in a foreground pixel value ('0' in this case) and other undesirable region in a background pixel value ('1' in this case).

The most important step in binarizaing the image is to choose a thresholding function. Lots of methods have been proposed for this, but here we are using "Otsu's method for threshold selection"[3].

The thresholding function used for binarization is shown below.

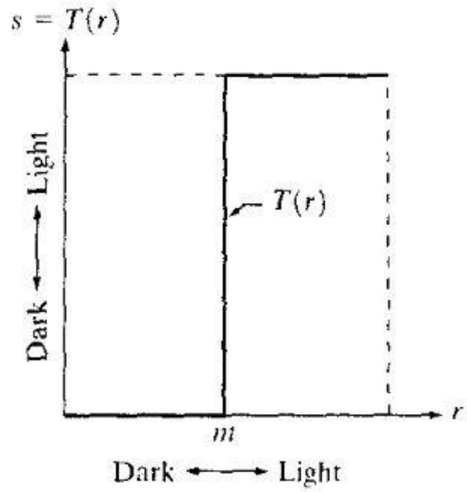

Fig 2.1 Thresholding function

Otsu's algorithm to find threshold:

i. The histogram and probabilities of each histogram level is calculated.

ii. Choose a minimum threshold value 't' .

iii. Calculate the occurrence probability of both classes (W_0 and W_1) into which the image is to be divided by binarization.

iv. Calculate μ_0 and μ_1 from the following formula.

$$\mu_0 = \sum_{i=0}^{t} iP_i$$

$\mu_1 = \sum_{i=t+1}^{L} iP_i$ Where, L = the maximum intensity level and t= threshold

v. Calculate the inter class variance $\sigma_b(t) = W_0(t)W_1(t)[\mu_0(t) - \mu_1(t)]^2$

vi. Repeat the process by increasing 't' .

vii. The maximum value of $\sigma_b(t)$ corresponds to the desired threshold value(t).

2.2.2 SIZE NORMALIZATION

The size of all the letters is normalized which means that the no of rows and columns of each letter is fixed so that, it would help for further processing. In this case, the row and column size is chosen as [64 64]. The size normalization is done while segmenting the letters from the lines.

2.2.3 SLANT CORRECTION

Different people have different style of writings; some prefer straight letters while some prefer to write in a tilted manner. Even sometimes the angle of inclination can be both ways i.e. can be less than 90^0 as well as greater than 90^0. Hence, it becomes difficult to compare. So, the slant i.e. the inclination of the character should be adjusted so that they make a specific angle with the vertical [2].

The algorithm of slant correction is as follows.

i. Divide the image into equal upper and lower half.
ii. Calculate the centroid of both the halves.
iii. The slope of the line joining the centroids defines the slant(β) of the letter.
iv. The slant is corrected by applying the following formulas.

$$x' = x - y * \tan(\beta - dif)$$

$$y' = y$$

Where x' and y' are the slant corrected coordinates and 'dif' is the parameter specifying the default slant. Here the default slant is taken to be $\frac{\pi}{2}$.

For centroid calculation, the x and y value of all the black pixels are summed and divided by total no of black pixels.

2.2.4 MIDDLING

Middling is not a general preprocessing step. But, here it was found that after correcting the slant, the character gets shifted to left or right portion of the image matrix. Hence , to bring it back to the middle position , the column no of the left most and the right most black pixel was found and the average was taken , which is then subtracted or added to get the new column value.

2.2.5 THINNING

Thinning is a morphological operation which is generally used to remove the selected foreground pixels from a binary image. By thinning the calculation of feature will not be affected by the uneven thickness of the characters. As, different people use different type of ink, so there is a great chance that the thickness will differ. Hence, thinning is very much required before we extract the features.

Here Hit and Miss Transform is used for thinning of images. The behavior of the thinning operation is determined by a structuring element.

Thinning operation is done by translating the origin of the structuring element to each and every pixel position of the image; and comparing it with the underlying image pixels in the structuring element. If the structuring element matches exactly with the fore ground and back ground pixels in the image, the image pixels underneath the center of the structuring element is set to zero.It is left unchanged otherwise.

The structures used here are

```
B{1}=[-1 -1 -1; 0  1 0 ; 1 1 1];
B{2}=[0 -1 -1 ; 1  1 -1; 1 1 0];
B{3}=[1  0 -1 ; 1 1 -1; 1 0 -1];
B{4}=[1  1  0 ; 1 1 -1; 0 -1 -1];
B{5}=[1  1  1; 0  1 0; -1 -1 -1];
B{6}=[0  1  1;-1  1  1; -1 -1 0];
B{7}=[-1 0  1;-1  1  1;-1  0  1];
B{8}=[-1 -1 0;-1  1  1; 0  1  1];
```

CHAPTER 3

FEATURE EXTRACTION

FEATURE EXTRACTION

Feature extraction is the crucial phase in numeral identification as each numeral has unique characteristics, which distinguishes it from other numerals. We obtain qualitative information about each character which helps for a better comparison. These features are obtained by computational algorithms, not by human operators. Hence, it is very important to extract the desired features so that the recognition of individual numerals becomes easier on the basis of the features of each numeral.

In this project, we are using two methods for obtaining the features. First the features are calculated by calculating the gradient and curvature matrix of each character image and then the box method approach is used.

The feature extraction part involves two steps, first to calculate the feature and then generating a feature vector by taking the obtained features.

3.1 Feature Extraction Using Curvature and Gradient properties

Gradient represents any directional change in the color, intensity of the image. We can’t use binary image as we are interested to measure the gradient. So, we have to convert it into a grayscale image.

We are applying a 2x2 mean filter several times on the binarized image to convert it into grayscale image. The 2x2 mean filter processes each pixel and replaces its value with the average of the pixel values inside the filter window [4].

3.1.1 Gradient Feature:

Roberts filter are applied to each pixel g(x,y) of the image by using the following equation and the gradient strength and gradient direction is calculated.

$$\Delta u = g(i+1, j+1) - g(i,j)$$

$$\Delta v = g(i+1, j) - g(i, j+1)$$

$$\textit{Direction:} \quad \theta(i,j) = tan^{-1}\frac{\Delta v}{\Delta u}$$

$$\textit{Strength:} \quad f(i,j) = \sqrt{\Delta u^2 + \Delta v^2}$$

−1	0
0	1

0	−1
1	0

Fig 3.1 Roberts cross gradient operators

3.1.2 Curvature Feature:

"Bi-quadratic interpolation method" is used to find the curvature.

In a grayscale image, the curvature *C* at x_0 is defined by $C = \frac{y''}{\sqrt{(1+y'^2)^3}}$

Where, g(x,y) is an equi-grayscale curve passing through x_0 .

y' is the first order derivative of y .

y" is the second order derivative of y.

y' and y" are obtained from the "bi-quadratic interpolating surface" taking the 8-neighbourhood of a pixel in the grayscale image into consideration.

The bi-quadratic surface is represented by

$$Z = \begin{bmatrix} 1 & x & x^2 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} 1 \\ y \\ y^2 \end{bmatrix}$$

Then, the equi-gray scale curve passing through x_0 is given by

$$(a_{22}x^2 + a_{12}x + a_{02})y^2 + (a_{12}x^2 + a_{11}x + a_{01})y + a_{20}x^2 + a_{10}x + a_{00} - f_0 = 0$$

Differentiation of both sides with respect to x, we get

$$y' = \frac{-\{(2a_{22}x + a_{12})y^2 + (2a_{21}x + a_{11})y + 2a_{20}x + a_{10}\}}{2y(a_{22}x^2 + a_{12}x + a_{02}) + a_{21}x^2 + a_{11}x + a_{01}}$$

Substituting the coordinates (0,0) of x_0 , y' at x_0 is given by

$$y' = -a_{10}/a_{01}$$

Similarly, the value of y'' at x_0 is given by

$$y'' = -2\,(a_{10}^2 a_{02} - a_{01}a_{10}a_{11} + a_{01}^2 a_{20})/a_{01}^3 \;;$$

Solving the simultaneous linear equations for 8 neighbor of x_0 , the coefficients of the bi-quadratic surface are given by

$$a_{10} = (f_1 - f_5)/2 \;;$$

$$a_{20} = (f_1 + f_5 - 2f_0)/2 \;;$$

$$a_{01} = (f_3 - f_7)/2 \;;$$

$$a_{02} = (f_5 + f_7 - 2f_0)/2 \;;$$

$$a_{11} = (f_2 - f_8) - (f_4 - f_6)/4 \;;$$

$\boldsymbol{f_2}$	$\boldsymbol{f_1}$	$\boldsymbol{f_8}$
$\boldsymbol{f_3}$	f_0	f_7
$\boldsymbol{f_4}$	f_5	f_6

Fig 3.2 Neighborhood of a pixel

The coefficients a_{10} and a_{01} are the first and the second order partial derivatives of $f(x, y)$ regarding x respectively .Similarly, a_{01} and a_{02} are partial derivatives regarding y, and a_{11} is the partial derivative w.r.t to x and y.

Substituting the value of y' and y'' in the curvature equation, we get

$$\boxed{C = -2\,(a_{10}^2 a_{02} - a_{01}a_{10}a_{11} + a_{01}^2 a_{20})/(a_{10}^2 + a_{01}^2)^{3/2}}$$

3.2 Feature Extraction using "Box-method approach"

"Box-method approach" is a simple method to extract the features which involves dividing the character image into no of boxes and then calculating certain parameters from each of the boxes. Finally, the parameter values are combined to get the final feature [5].

In this method, we are first dividing the image into certain boxes and the no of boxes is decided experimentally. The no of boxes has the greatest impact on the result as this only decides the no of final features and no of comparisons. In our case, as the character image is of size [64 64] , we are taking the box size as [8 8] and [4 4] which yields total no of 64 and 256 no of boxes. From this, two feature vector of different length are obtained which is later used for comparing results.

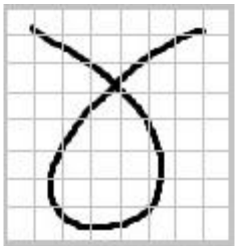

Fig 3.3 Box Method

Algorithm to extract feature

i. Divide the image into several boxes.

ii. Calculate the vector distance of each black pixel by taking the left bottommost pixel as the origin.

$$d(x, y) = \sqrt{x^2 + y^2}$$

iii. In each box, the average of each vector distance corresponding to the black pixels is calculated.

iv. Finally, all the average value is combined to form a single row vector.

3.3 Feature Vector generation

Here, two feature vectors are obtained as features are calculated in two ways. Again, in gradient and curvature method, two different vectors corresponding to gradient and curvature feature are obtained which are then concatenated to form a final feature vector.

3.3.1 Feature vector from gradient and curvature features

As already stated, finally the feature vectors of both gradient and curvature features are combined [4].

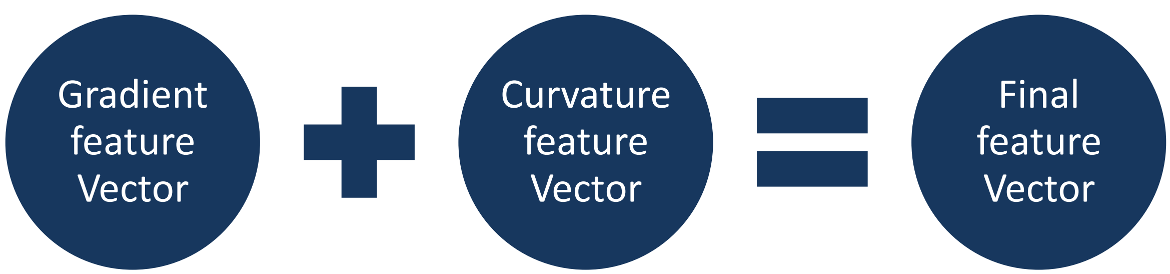

Fig 3.4 Formation of feature vector in gradient and curvature method

Steps to form the final feature vector in gradient and curvature method

i. Quantize the curvature and gradient direction matrix into 32 levels each. In case of gradient direction, the values are quantized with the intervals of $\pi/16$, as the range of direction is $-\pi$ to $+\pi$.

ii. Divide the image matrix into 49 blocks (7 horizontal x 7 vertical).

iii. Accumulate the strength of the gradient in 32 directions and curvatures in each block to produce two separate feature vectors of size 1x32 corresponding to both gradient direction and curvature.

iv. From 49 blocks, two 49x32 feature vectors will be obtained. Concatenate both vectors horizontally to form the final feature vector.

3.3.2 Feature vector from box method

As stated in the feature extraction part, from each box we are getting a value which corresponds to the average value of the vector distance of black pixels. Hence, by combining the values from all the boxes in a row vector, we get the final feature vector.

3.4 DIMENSION REDUCTION

The obtained feature vectors might have a high dimension which makes it difficult for further processing. Even some processors are unable to process and show error. Sometimes, time complexity increases and effective results are not obtained. So, it is desirable to remove the redundant data from the feature vector .Different methods such as Principal Component Analysis(PCA) , Factor Analysis can be employed. Here, we have used only PCA which is an efficient method for dimension reduction. This method not only reduces the dimension but also results in no data loss. It takes the use of correlation to find the data redundancy [6].

Algorithm to apply PCA:

i. Subtract the mean vector from the feature vector.
ii. Find the covariance matrix.
iii. Calculate the Eigen values and Eigen vectors from covariance matrix.
iv. Choose the Eigen values which we want to keep in the final vector. Here, we have chosen Eigen values which are greater than 1.
v. Form a set of Eigen vectors corresponding to the set of Eigen values chosen.
vi. Derive a new feature vector by multiplying the new set of Eigen vector and original data.

CHAPTER 4

PATTERN RECOGNITION

PATTERN RECOGNITION

This step involves assigning each numeral a different class based on its feature. Features are taken from known samples and then unknown samples are compared with those known samples. Different techniques such as Neural Networks, Minimum distance classifier, Bayesian classifier, Quadratic classifier, Correlation are used for this purpose. In this project, we have opted for Artificial Neural Networks and Correlation. Both methods give different result and also are applicable in different cases.

4.1 Artificial Neural Networks

So, neural network is used when we have large no of samples of each numeral with variations among them which are used to train the network and correspondingly weights are updated. Finally, the weights are applied to the testing samples to get the correct output. The main advantage of using Neural networks is that it is unaffected by the differing shape and style of testing samples as the network is already trained with large variations.

Back propagation algorithm is used to update the weights and bias matrix. Here, the learning parameter/step size ‘η’ has a major role as it controls the rate at which the error is reduced which further determines the time complexity.

An artificial neural network can be seen as a computer program that is designed to recognize patterns and learn "like" the human brains. The structure of a neural network is shown below.

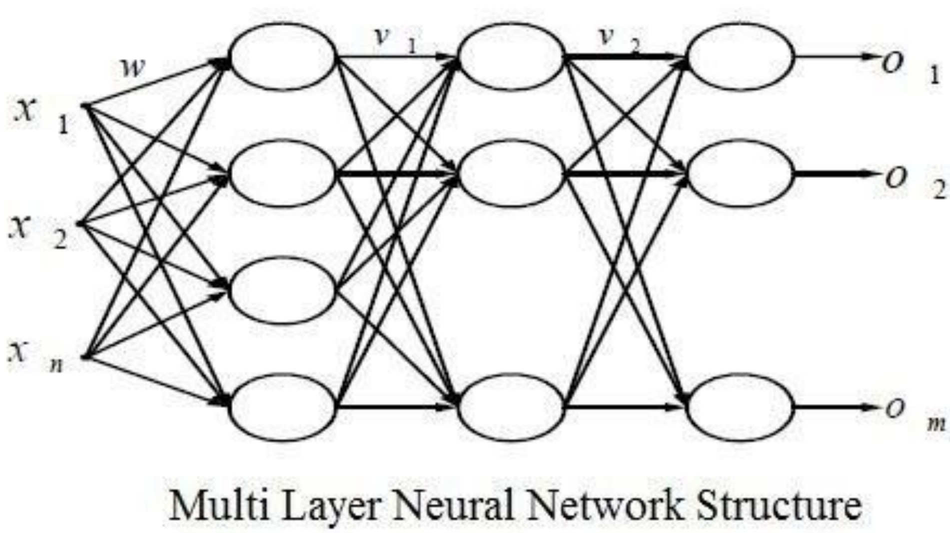

Fig 4.1

An ANN is composed of a large no of highly interconnected processing elements (artificial neurons) working in unison to solve a specific problem. An artificial neuron has (i) inputs X_1, X_2… X_n; (ii) a summing element (iii) a nonlinear element; (iv) connection weighing element, W_1, W_2… W_n, that are adjustable connection weights and (v) output Y. The factor $W_0.X_0 = W_0$ is the bias b, $X_0 = 1$ always.

$$net = \left(\sum_{i=1}^{n} W_i X_i\right) + b = \sum_{i=1}^{n} W_i X_i$$

$$Y = f(net)$$

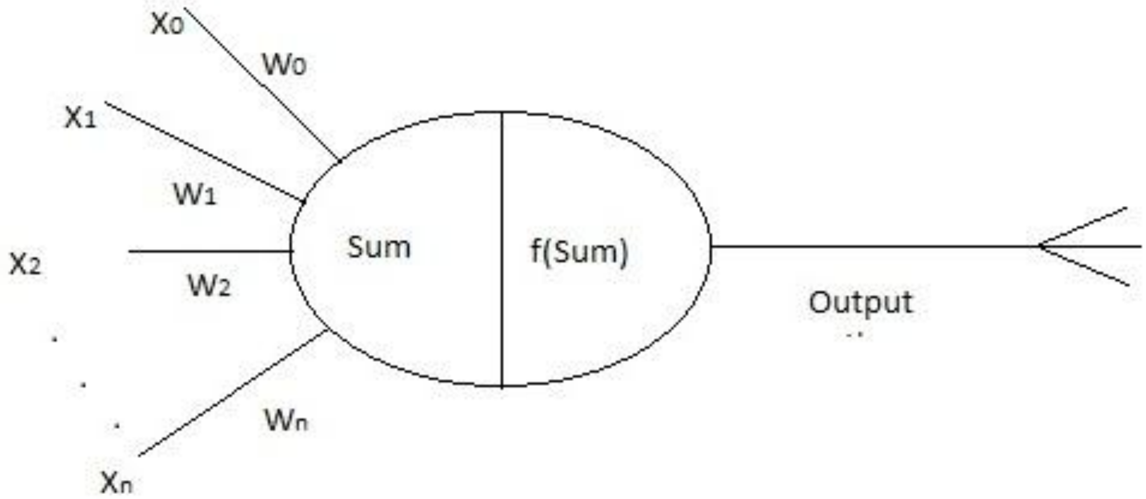

Fig 4.2 function of a neuron

Here, 'logsigmoid' function has been used as the activation function. The no of input layers is equal to the no of features in each the feature vector of each input character. The no of hidden layers have been taken as 10 and the output layers is equal to the no of class, here taken as 10 for 10 numerals (0, 1, 2…9).

4.1.1 Back propagation algorithm

This algorithm is used to update the weights after one output is obtained. The output is compared with the target and error signal is generated. Then the weights are updated using the following formulas till the error becomes less than the goal error. In our case, the no of iteration is taken as 10000 and goal error is chosen as 10 e -5 [9].

Algorithm:

Consider the following diagram.

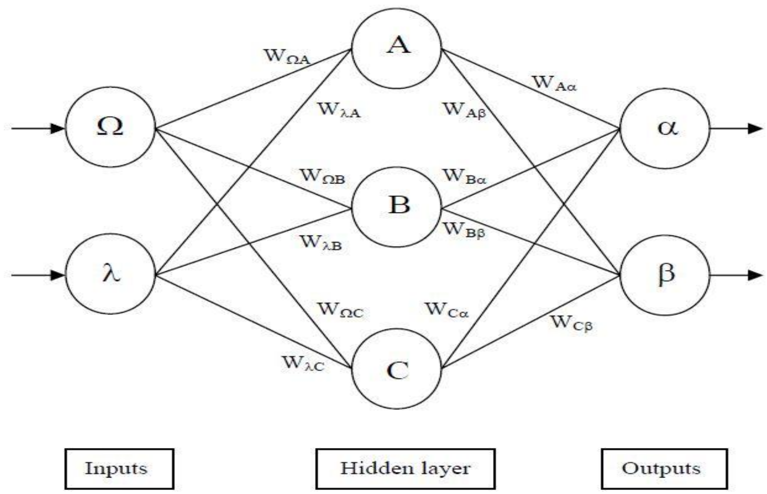

Fig 4.3 Back propagation Algorithm

1. Calculate errors of output neurons

$$\delta_\alpha = out_\alpha (1 - out_\alpha) (Target_\alpha - out_\alpha)$$

$$\delta_\beta = out_\beta (1 - out_\beta) (Target_\beta - out_\beta)$$

2. Change output layer weights

$$W^+_{A\alpha} = W_{A\alpha} + \eta\delta_\alpha \, out_A \qquad W^+_{A\beta} = W_{A\beta} + \eta\delta_\beta \, out_A$$

$$W^+_{B\alpha} = W_{B\alpha} + \eta\delta_\alpha \, out_B \qquad W^+_{B\beta} = W_{B\beta} + \eta\delta_\beta \, out_B$$

$$W^+_{C\alpha} = W_{C\alpha} + \eta\delta_\alpha \, out_C \qquad W^+_{C\beta} = W_{C\beta} + \eta\delta_\beta \, out_C$$

3. Calculate (back-propagate) hidden layer errors

$$\delta_A = out_A (1 – out_A) (\delta_\alpha W_{A\alpha} + \delta_\beta W_{A\beta})$$

$$\delta_B = out_B (1 – out_B) (\delta_\alpha W_{B\alpha} + \delta_\beta W_{B\beta})$$

$$\delta_C = out_C (1 – out_C) (\delta_\alpha W_{C\alpha} + \delta_\beta W_{C\beta})$$

4. Change hidden layer weights

$$W^{+}_{\lambda A} = W_{\lambda A} + \eta\delta_A\, in_\lambda \qquad W^{+}_{\Omega A} = W^{+}_{\Omega A} + \eta\delta_A\, in_\Omega$$

$$W^{+}_{\lambda B} = W_{\lambda B} + \eta\delta_B\, in_\lambda \qquad W^{+}_{\Omega B} = W^{+}_{\Omega B} + \eta\delta_B\, in_\Omega$$

$$W^{+}_{\lambda C} = W_{\lambda C} + \eta\delta_C\, in_\lambda \qquad W^{+}_{\Omega C} = W^{+}_{\Omega C} + \eta\delta_C\, in_\Omega$$

The constant η (called the learning rate, and nominally equal to one) is put in to speed up or slow down the learning if required.

4.2 Correlation

Correlation method is used when we have a standard format of numerals with which we can compare the testing samples. Though, the result is affected a little b the variation in input compared to that of Neural networks, but it gives a better result if we consider the time complexity. The preprocessing steps play a major part in order to neglect the variations in the shape of the numeral and try to make it similar to the standard.

Here first, the input character is correlated with the cell arrays of the standard characters and the correlation coefficient is compared in each case. The maximum correlation coefficient corresponds to the actual numeral. To improve the accuracy, instead of using the cell arrays directly, the features obtained from the box-method are correlated, which indeed provides better result. And, better correlation coefficient was obtained by increasing the no of boxes but up to a certain extent.

Here, two dimensional correlations are used. The formula for finding the correlation is given by

$$\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(A_{ij}-A')(B_{ij}-B')}{\sqrt{\left(\sum_{i=1}^{m}\sum_{j=1}^{n}\left(A_{ij}-A'\right)^2\right)\left(\sum_{i=1}^{m}\sum_{j=1}^{n}\left(B_{ij}-B'\right)^2\right)}}$$

CHAPTER 5

RESULTS &

DISCUSSION

RESULTS & DISCUSSION

5.1 Results

5.1.1 Segmentation Results

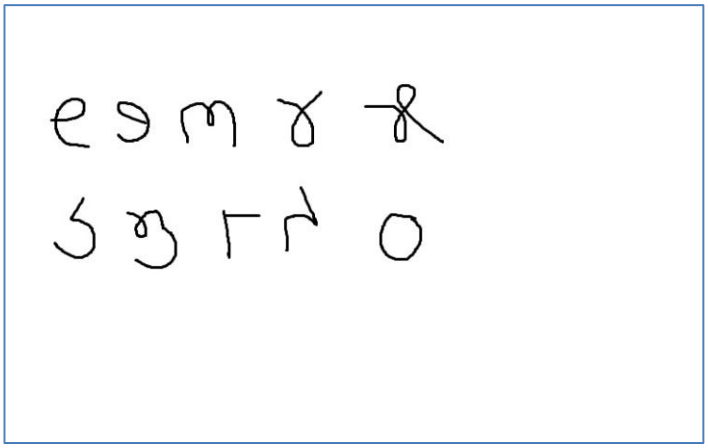

Fig 5.1 Original binarized image

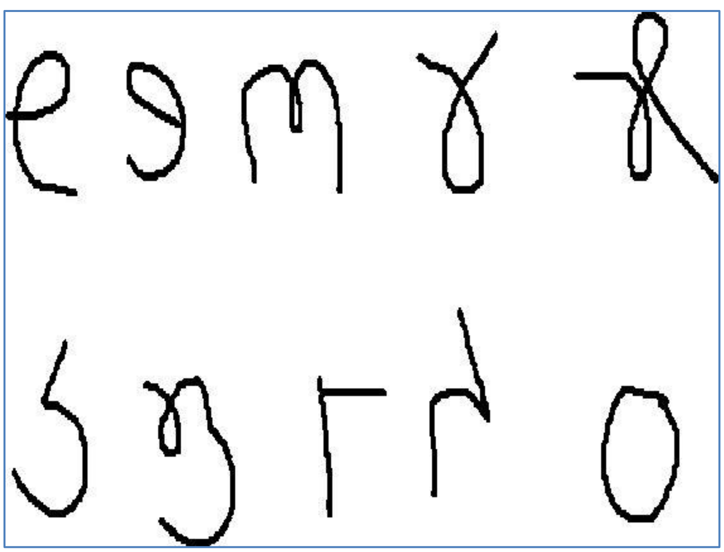

Fig 5.2 Image after using “Clip” function

Fig 5.3 Result of horizontal segmentation

Fig 5.4 Results of vertical segmentation

5.1.2 Space vector

Space vector is found as **spacevector** = [39 44 58 60 0 44 65 34 88 0];

5.1.3 Prepressing Results

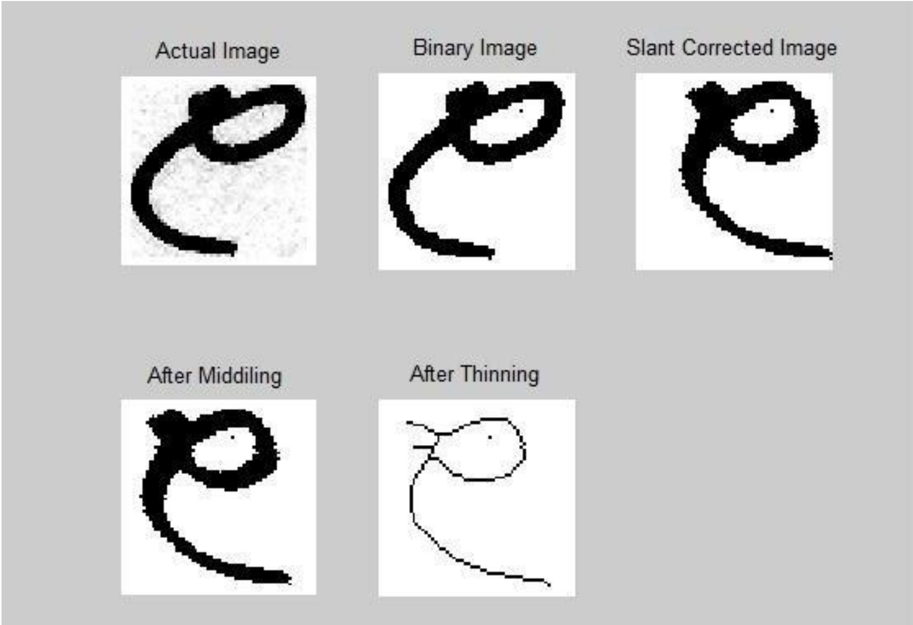

Fig 5.5 Preprocessing Results

5.1.4 Results of gradient and curvature method

ACTUAL IMAGE ,BINARY IMAGE AND GRAY SCALE IMAGE

Fig 5.6 Gray scale image

Direction of Gradient Strength Of Gradient Curvature

Fig 5.7 Gradient direction, gradient strength and curvature of image

5.1.5 Results of PCA

490 samples of each numeral were used for training purpose and the dimension of their feature is reduced by PCA as shown below.

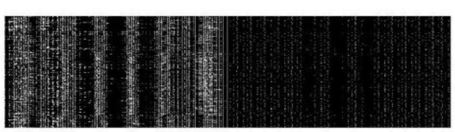

Fig 5.8 size (before PCA) = 490 x 3136

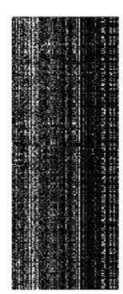

Fig 5.9 size (After PCA) = 490 x 80

5.1.6 Results using neural networks

Table 5.1 Box method approach

Numerals	Maximum Output	Minimum Output	Recognition rate
0	.9828	.9551	78%
1	.9811	.9593	88%
2	.9865	.9714	84%
3	.9830	.9675	90%
4	.9828	.9703	88%
5	.9853	.9628	94%
6	.9846	.9651	91%
7	.9875	.9660	88%
8	.9852	.9556	94%
9	.9846	.9636	86%

Table 5.2 Gradient and curvature method

Numerals	Maximum Output	Minimum Output	Recognition rate
0	.9828	.9551	82%
1	.9811	.9593	95%
2	.9865	.9714	94%
3	.9830	.9675	96%
4	.9828	.9703	93%
5	.9853	.9628	96%
6	.9846	.9651	92%
7	.9875	.9660	92%
8	.9852	.9556	100%
9	.9846	.9636	92%

5.1.7 Results of Correlation method

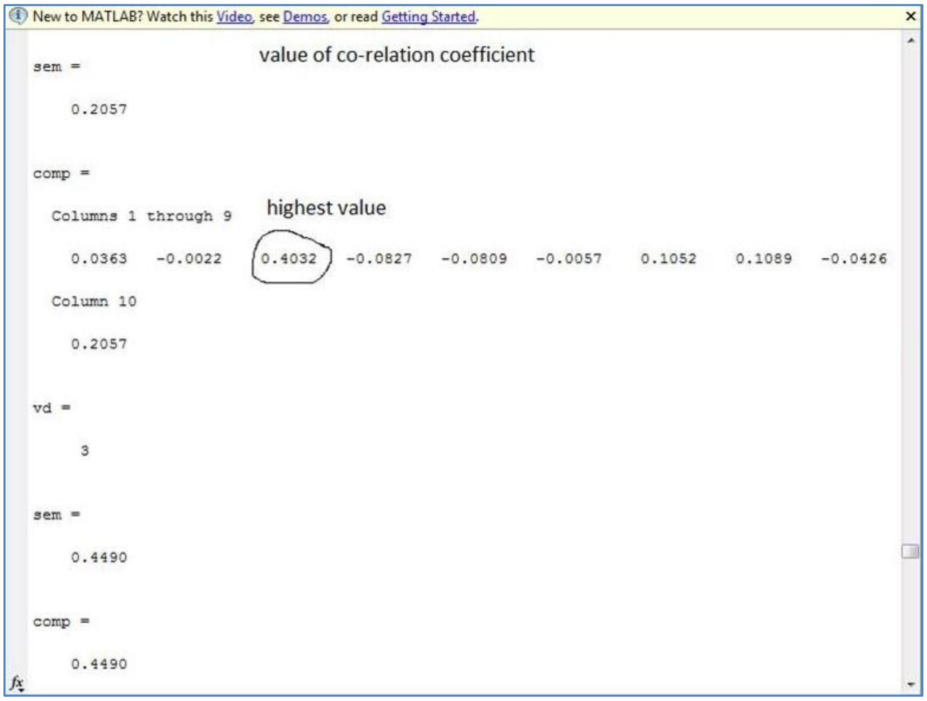

Fig 5.10 Correlation coefficient

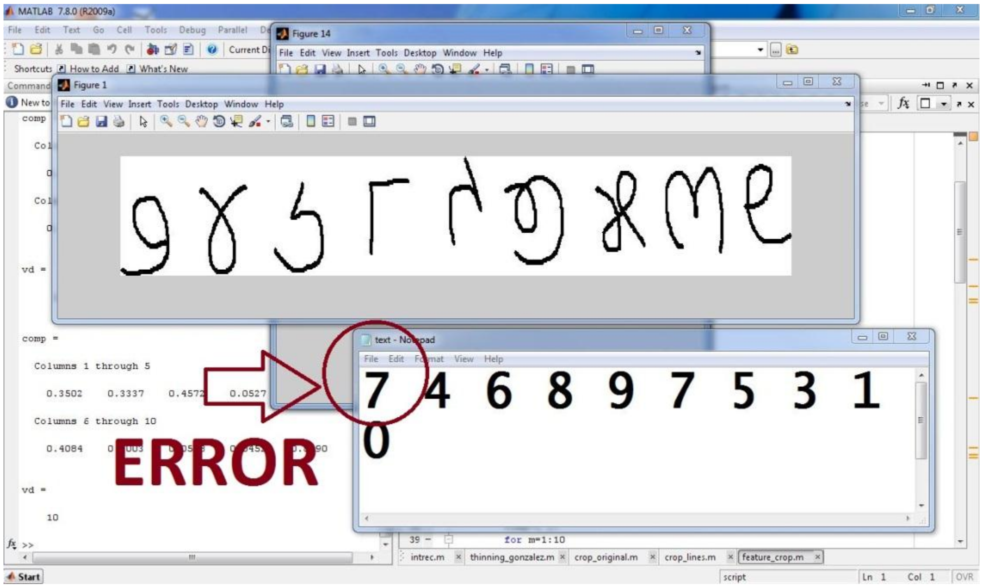

Fig 5.11 Result of correlating the images directly

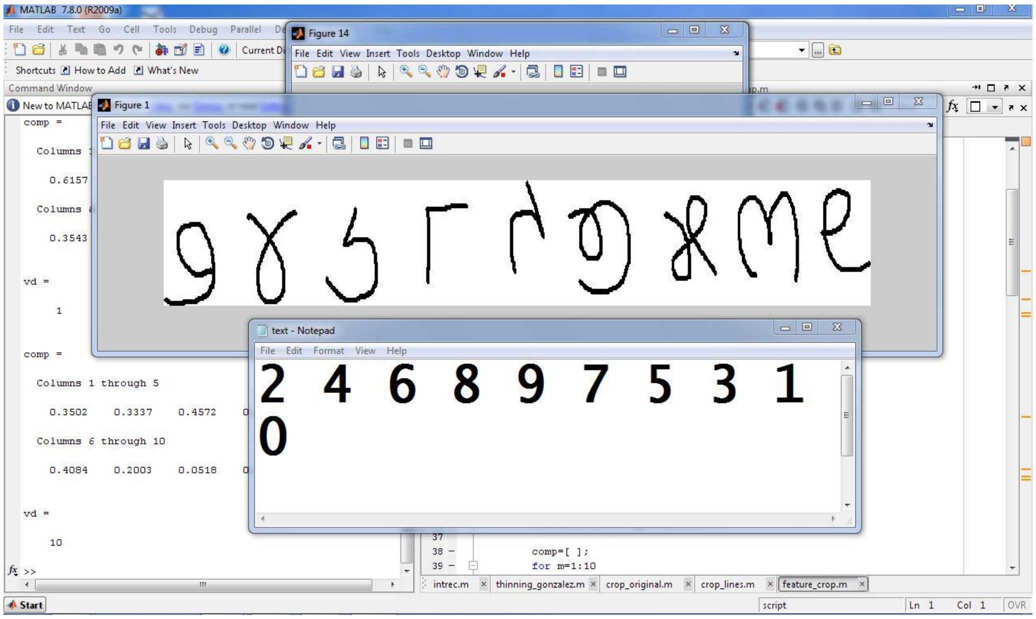

Fig 5.12 Results after correlating the features from box-method approach

5.2 DISCUSSION

The performance analysis of neural networks and correlation method says that, neural networks perform better for varying styles of handwritten characters. As, we are dealing with handwritten characters, hence the neural network should be trained by taking enough no of samples so that it remains unaffected by the deviations from standard. But it was experienced that while using the gradient and curvature method, the dimension of the feature vector becomes so large which creates problem in processing. For which, PCA is applied to reduce the dimension. More the no of samples, more the compression is achieved. But while testing, it is possible that we are left with very small no of samples to test, where PCA won't yield good result. In case of multi-layer networks, the Learning Coefficient η determines the size of the weight changes. A small value of η results in a very slow learning process. The large weight changes may cause the desired minimum to be missed if the learning coefficient is too large. Depending on the problem, η should lie between 0.05 to 1. The multilayer feed forward networks trained with the Back propagation method are probably the most practically used networks for real world applications.

Some problem occurred in case of box method is the feature vectors in case of each numeral don't vary much from each other, which creates confusion while neural network recognition scheme is applied. This problem is solved by the correlation method. When the correlation method is first applied without taking the box method into consideration, we found out that the system sometimes confuses between the numerals "2" and "7" as the writing style is nearly same in both cases. Other pairs where such confusion arose are "1, 2 & 0"; 6 & 9"; "0 & 4". To get rid of this problem, the features of box-method are correlated and it was found out that such confusions are removed up to some extent. The no. of boxes was increased and better performance was achieved. But, after a certain increment the performance it was found out that the performance degrades. This is because the box method finds out the similarity between two numerals in certain parts of the image , such as the center of "5" will almost lie in the middle box , but if we keep on increasing the boxes, then this probability also decreases. The correlation method was found to have very small time complexity compared to that of neural networks. So, this scheme can be used for both off-line and online recognition.

It was also seen that every time the recognition rate for '5', '3', '8' was high as they possess unique characteristics in terms of their geometrical shape. The confusion between '2' and '7' was removed but not completely as some people have a very similar writing style while writing these two numerals. Again, while testing '1' or '2' or '4' one has to be careful as if these numerals are written with a greater distortion then they may look like '0' . To avoid these confusions, we can apply different features to the system and consider the output in each case. Or, one feature can be used to classify some similar numerals to one category whereas the other feature will take care of the final result.

BIBLIOGRAPHY

REFERENCES

[1] U. Pal, B.B. Chaudhuri, "**Indian script character recognition: a survey**", Pattern Recognition 37 (2004) 1887 – 1899.

[2] M. Hanmandlu, O.V. Ramana Murthy, "**Fuzzy model based recognition of handwritten numerals**", Pattern Recognition 40 (2007) 1840 – 1854

[3] Otsu , N., "**A threshold selection method from gray level histograms**", "IEEE Transaction on systems, man and cybernetics, Vol. 9, No. 1, 1979, pp. 62-66

[4] Meng Shi, Yoshiharu Fujisawa, Tetsushi Wakabayashi, Fumitaka Kimura , "**Handwritten numeral recognition using gradient and curvature of gray scale image**" , IEEE Trans , Pattern Recognition 35 (2002) 2051 – 2059

[5] M. Hanmandlu, O.V. Ramana Murthy, "**fuzzy model based recognition of handwritten numerals**", Pattern Recognition 40 (2007) 1840 – 1854.

[6] Jonathon Shlens , "**A Tutorial on Principal Component Analysis**" ,in : Institute for Nonlinear Science, University of California, San Diego La Jolla, CA 92093-0402(2005).

[7] Alexander J. Faaborg, "***Using Neural Networks to Create an Adaptive Character Recognition System***", March 2002, available at: http://web.media.mit.edu/~faaborg/research/cornell/hci_neuralnetwork_finalPaper.pdf

[8] Michael D. Garris, Charles L. Wilson, **Neural Network-Based Systems for Handprint OCR Applications**, IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 7, NO. 8, AUGUST 1998.

[9] **Back Propagation Network**: Soft Computing Course Lecture, notes, slides, www.myreaders.info/html/soft_computing.html , R C Chakraborty.

[10] T. Wakabayashi, S. Tsuruoka, F. Kimura, Y. Miyake, **Increasing the feature size in handwritten numeral recognition to improve accuracy**, Systems and Computers in Japan (English Edition), Scripta Technical, Vol. 26, No. 8, 1995,pp. 35–44.

[11] Nafiz Arica and Fatos T. Yarman-Vural, "**Correspondence An Overview of Character Recognition focused on Off-Line Handwriting**", IEEE Transactions on Systems, Man, and Cybernetics – part C: Applications and Reviews, VOl.31, No.2, May 2001.

[12] Richard G. Casey and Eric Lecolinet, "**A Survey methods and Strategies in Character Segmentation**", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 18, No.7, July 1996.

[13] **Image Processing Toolbox** in Matlab , www.mathworks.com .

[14] **Digital Image Processing**, R C Gonzalez, R E Woods (2002)