



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Handwritten Text Recognition Using ICA with Voice Enhancement Features

Beena K

Lecturer (Electronics), Maharajas Technological Institute, Thrissur, Kerala, India

ABSTRACT: Handwriting is one of the most important means of daily communication. Although the problem of handwriting recognition has been considered for many years there are still many open issues, especially in the task of unconstrained handwritten text recognition. This project focuses on the automatic system that recognizes continuous English text based on Independent component analysis with voice enhancement features. ICA is a method for finding underlying factors or components from multi-dimensional data. ICA unlike other methods looks for components that are both statistically independent and non Gaussian. The aim of ICA is to separate independent source signals from the observed signals, which is assumed to be the linear mixtures of independent source components. The input provided is an image of some text, and the system produces, as output, an ASCII transcription of the input. This procedure involves a number of processing steps, some of which are quite difficult. Typically, preprocessing, segmentation, feature extraction, classification, and post processing operations are required. The ultimate goal of handwritten text recognition is to have machines which can read any text with the same recognition accuracy as humans but at a faster rate. Text recognition is important in a number of future applications, for example, the transcription of personal notes, faxes, and letters, or the electronic conversion of historical handwritten archives in the context of the creation of digital libraries. This project shall be useful to our society at large and particularly to the visually challenged as voice enhancement features are also included

KEYWORDS: Eigen Vector, Handwritten text recognition, ICA, MATLAB, preprocessing, segmentation, text identification, voice enhancement

I. INTRODUCTION

English comprises of 26 basic alphabets which are simple to write and recognize but becomes very complex when they are handwritten. The paper presented makes an effort towards the recognition of offline handwritten English text recognition by using an Independent component analysis (ICA) based feature extraction with additional voice enhancement features.

A typical offline handwritten character recognition system involves activities, namely, digitization, pre-processing, segmentation, feature extraction, classification and post processing.

The recognition of cursive handwriting is very difficult due to a large number of variations found in shapes and overlapping of characters. In the offline handwriting recognition system, the pre-written text document is converted into a digital image through an optical scanner. In a handwritten text, there is a good amount of variation in the writing style and size of a character.

Some important applications of offline handwritten character recognition system are - Handwritten notes recognition - Reading notes that are used to record facts, topics, or thoughts and written to assist memory

Cheque reading - Signature verification and recognition of amount filled by user, Postcode recognition - Reading handwritten postal address, recognition of digits of postcodes and to sort mails/parcels automatically, Conversion of handwritten document to text format.

Independent component analysis (ICA) is a method for finding underlying factors or components from multivariate (multi-dimensional) statistical data. What distinguishes ICA from other methods is that it looks for components that are both statistically independent, and non Gaussian. ICA is important because of its potential applications in signal and image processing. The goal of ICA is to separate independent source signals from the observed signals, which is assumed to be the linear mixtures of independent source components. The mathematical model of ICA is formulated by

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

mixture processing and an explicit decomposition processing. ICA decomposition enables to separate reflections, shadows and specularities from natural scene texts.

This paper will help in better interface between human beings and computers and also facilitate the progress of the expansion of such systems for recognition of handwritten texts of other languages.

Voice enhancement features enables text to speech conversion which will be useful to people with learning disabilities, visual impairment, literacy difficulties, and to those who speak or understand the language but do not read it.

II. METHODOLOGY

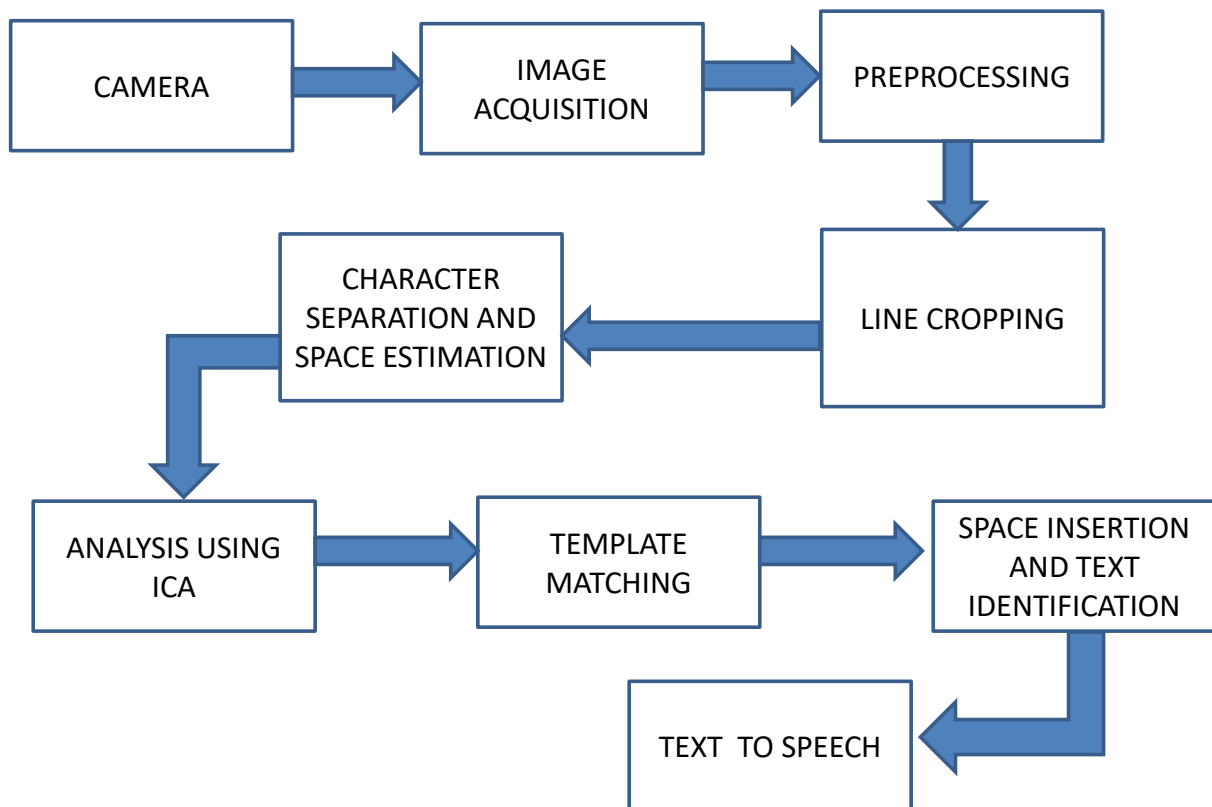


Fig Methodology - Block diagram

IMAGE ACQUISITION

A scanned image of the text to be recognised is acquired through a scanner, digital camera or any other suitable digital device as an input image. Image Acquisition Toolbox provides functions and blocks that let you to connect industrial and scientific cameras to MATLAB. It includes a MATLAB app that lets you interactively detect and configure hardware properties. The toolbox enables acquisition modes such as processing in the loop, hardware triggering , background acquisition and synchronizing acquisition across multiple devices. Here we have to set the device type, resolution, device properties etc. We set the resolution as YUY2_320*240, frames per trigger as 1, that represents the



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

number of frames per trigger we want to acquire and color space as grayscale. The image should have a specific format such as JPEG, bmp, png etc.

PREPROCESSING

Preprocessing is a series of operations performed on the scanned input image. It essentially enhances the image rendering it suitable for segmentation. The various tasks performed on the image in preprocessing stage are binarization, filtering and smoothing, thinning, alignment, normalization, and baseline detection. Binarization process converts a gray scale image into a binary image using thresholding technique (image appears as black letters on white background). $BW=im2bw(I, level)$ converts the grayscale image I to a binary image. The output image BW replaces all pixels in the input image with luminance greater than level with the value 1 (white) and replaces all other pixels with the value 0 (black). To compute the level argument we can use the `graythresh`. If we do not specify level `im2bw` uses the value 0.5.

$IM2= im$ complement (IM) computes the complement of the image IM . IM can be binary, grayscale or RGB image. $IM2$ has the same class and size as IM . In the complement of a binary image, zeros become one and ones become zero i.e. Black and white are reversed. In the complement of an intensity or RGB image each pixel value is subtracted from the maximum pixel value subtracted by the class and the difference is used as the pixel value in the output image where dark areas become lighter and light areas become darker.

SEGMENTATION

Image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels). The goal of segmentation is to simplify or change the representation of an image into something that is more meaningful and easier to analyze. The segmentation stage separates the different logical parts, like text from graphics, lines of a paragraph, and characters of a word. After the preprocessing stage, most Optical Character Recognition (OCR) systems isolate the individual characters before recognizing them. Segmenting a page of text can be broken down into two levels - page decomposition and word segmentation. When working with pages that contain different object types like graphics, headings, mathematical formulas, and text blocks, page decomposition separates the different page elements, producing text blocks, lines, and sub-words. While page decomposition might identify sets of logical components of a page, word segmentation separates the characters of a sub-word. Here line segmentation is done by finding rows corresponding to beginning and end of each line and lines are cropped one by one. Character segmentation is done column wise. Columns corresponding to beginning and end of each letter is identified and cropped one by one. In this process the columns in between two characters are also estimated.

In threshold based segmentation, histogram thresholding and slicing techniques are used to segment the image. In edge based segmentation detected edges in an image are used to represent object boundaries and are used to identify these objects. In region based segmentation edge based technique is used to find the object boundaries and then locate the object itself by filling them in, a region based technique starting in the middle of an object and then growing outward until it meets the object boundaries.

FEATURE EXTRACTION

In this stage, the features of the characters that are crucial for classifying them at recognition stage are extracted. This is an important stage as its effective functioning improves the recognition rate and reduces the misclassification. The feature extraction stage analyzes a text segment and selects a set of features that can be used to uniquely identify the text segment. First features are computed and extracted and then most relevant features are selected to construct feature vector which is used eventually for recognition. The computation of features is based on structural, statistical, directional, moment, transformation like approaches. Selection of a feature extraction method is probably the single most important factor in achieving high recognition performance in character recognition systems. Different feature extraction methods are designed for different representations of the characters, such as solid binary characters, character contours, thinned (skeletons sentence) or gray-level sub images of each individual character. On feature extraction stage each character is represented by a feature vector, which becomes its identity. The major goal of feature



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

extraction is to extract a set of features, which maximizes the recognition rate with the least amount of confusion. The captured picture is normalized and thinned and its distinct features are extracted.

In this paper feature extraction is performed using Independent Component Analysis (ICA).

Independent Component Analysis

ICA finds the independent components (also called factors, latent variables or sources) by maximizing the statistical independence of the estimated components.

The two broadest definitions of independence for ICA are

Minimization of mutual information and Maximization of non-Gaussianity

The Minimization of Mutual information (MMI) family of ICA algorithms uses measures like Kullback-Leibler Divergence and maximum entropy. The non-Gaussianity family of ICA algorithms, motivated by the central limit theorem, uses kurtosis and negentropy.

Typical algorithms for ICA use centering (subtract the mean to create a zero mean signal), whitening (usually with the eigenvalue decomposition), and dimensionality reduction as preprocessing steps in order to simplify and reduce the complexity of the problem for the actual iterative algorithm. Whitening and dimension reduction can be achieved with principal component analysis or singular value decomposition. In general, ICA cannot identify the actual number of source signals, a uniquely correct ordering of the source signals, nor the proper scaling (including sign) of the source signals.

Classification using the values of all pixels gives good results. The extra dimensionality can lead to random noise which will make measurement less accurate. By lowering the number of dimensions of the data it is possible to run the classification algorithm much faster and on larger data sets. A technique that can reduce dimensionality is ICA, where the data is projected on its k directions of largest variance.

Main aim of ICA is dimensionality reduction and feature extraction. By reducing the dimensionality of data, speed of computation is increased without losing important information. ICA is applied on eigen vector approach to reduce the dimensionality of a large data set. It exploits inherently non Gaussian features of the data and employs higher moments. ICA is required when data cannot be ensemble, when data appears to be noisy. It minimises higher order statistics such as kurtosis, a measure of non gaussianity, thus minimising mutual information of the output.

Signal mixtures tend to have Gaussian probability density functions, and source signals tend to have non-Gaussian probability density functions. Each source signal can be extracted from a set of signal mixtures by taking the inner product of a weight vector and those signal mixtures where this inner product provides an orthogonal projection of the signal mixtures. The remaining challenge is finding such a weight vector. One type of method for doing so is projection pursuit.

Projection pursuit seeks one projection at a time such that the extracted signal is as non-Gaussian as possible. This contrasts with ICA, which typically extracts M signals simultaneously from M signal mixtures, which requires estimating a $M \times M$ un mixing matrix. One practical advantage of projection pursuit over ICA is that fewer than M signals can be extracted if required, where each source signal is extracted from M signal mixtures using an M -element weight vector.

ICA identifies independent components for nongaussian signals but one cannot determine the variance of the independent components and cannot rank the order of dominant components.

ICA Estimation

1. The independent components are assumed statistically independent

Random variables y_1, y_2, \dots, y_n are said to be independent if information on the value of y_i does not give any information on the value of y_j for $i \neq j$

Y_i are independent if and only if the joint pdf is factorizable

$$P(y_1, y_2, \dots, y_n) = P_1(y_1) P_2(y_2) \dots P_n(y_n)$$

2. The independent components must have non Gaussian distribution

The higher order cumulants are zero for Gaussian distribution but such higher order information is essential for estimation of the ICA model.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Gaussian distribution has two parameters - mean and variance. When there is zero mean and unit variance the Gaussian distribution becomes normal. For ICA independent components must be non Gaussian, Gaussian variables makes ICA impossible.

ICA Preprocessing Centering

Before applying ICA to rotate the data in order to obtain uncorrelated axes, any existing shift needs to be countered by subtracting the mean of the data from each data point. This simply corresponds to centering the data such that its average becomes zero. Centre X for that subtract its mean vector $m=E\{X\}$ to make X as a zero mean variable.

Normalization

In image processing normalization is a process that changes the range of pixel intensity values. It is referred to as dynamic range expansion, which is nothing but to bring the image into a range that is familiar or normal to the senses, hence the name. For eg. If the intensity range of the image is 50 to 180 and the desired range is 0 to 255, the process entails subtracting 50 from each of pixel intensity, making the range 0 to 130. Then each pixel intensity is multiplied by 255/130, making the range 0 to 255. The eigenvectors of the covariance matrix points in the direction of the largest variance of the data. However, variance is an absolute number, not a relative one. This means that the variance of data, measured in centimeters (or inches) will be much larger than the variance of the same data when measured in meters (or feet). Consider the example where one feature represents the length of an object in meters, while the second feature represents the width of the object in centimeters. The largest variance, and thus the largest eigenvector, will implicitly be defined by the first feature if the data is not normalized. To avoid this scale-dependent nature of ICA, it is useful to normalize the data by dividing each feature by its standard deviation. This is especially important if different features correspond to different metrics.

Whitening

After centering we transform the observed vector X linearly so that we obtain a new vector X' which is white, i.e. its components are uncorrelated and their variances equal to unity. The covariance matrix of X' equals the identity matrix. Since the data will be projected onto the largest eigenvectors to reduce the dimensionality, the eigen decomposition needs to be obtained. One of the most widely used methods to efficiently calculate the eigen decomposition is Singular Value Decomposition (SVD). One popular method for whitening is to use the Eigen value decomposition of the covariance matrix using singular value decomposition algorithm. Whitening reduces the number of parameters to be estimated by removing eigen value that are too small. After preprocessing we will get a rotated version of the original square. The eigen vector represent the direction of the largest variance of the data and the eigen value represent the magnitude of the variance in that direction. Subsequent eigenvectors are always orthogonal to the largest eigen vector due to the orthogonality of rotation matrices.

Projection

To reduce the dimensionality, the data is simply projected onto the largest eigenvectors. Let V be the matrix whose columns contain the largest eigenvectors and let D be the original data whose columns contain the different observations. Then the projected data D' is obtained as $D' = V^T D$. We can either choose the number of remaining dimensions, i.e. the columns of V , directly, or we can define the amount of variance of the original data that needs to be kept while eliminating eigenvectors. If only N eigenvectors are kept, and $e_1 \dots e_N$ represent the corresponding eigen values, then the amount of variance that remains after projecting the original d -dimensional data can be calculated as:

$$s = \frac{\sum_{i=0}^N e_i}{\sum_{j=0}^d e_j}$$



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Eigen Value and Eigen Vector

The eigen vector of a linear operator are nonzero vectors, which when operated by the operator results in a scalar multiple of them. Scalar is then called eigen value(λ) associated with the eigen vector (X). Eigen vector is a vector that is scaled by linear transformation. When a matrix acts on it, only the vector magnitude is changed not the direction.

$AX = \lambda X$, where A is a vector function.

$\det(A - \lambda I) = 0$, when evaluated becomes a polynomial of degree n, called the characteristic polynomial of A. If A is N by N then there are n roots of the characteristic polynomial. Thus there are n eigen values of A satisfying the equation $A X_i = \lambda_i X_i$ where $i=1,2,\dots,n$. If the eigen values are all distinct, then there are n associated linearly independent eigen vectors, whose directions are unique, which span an n dimensional Euclidean space.

Image Representation

Training set of M images of size $N \times N$ are represented by vector of size N^2 . Feature vector of a image is stored in a $N \times N$ matrix. Those two dimensional vector is changed to one dimensional vector. Each image is represented by the vector Γ_i .

Mean Image

It is calculated by $\Psi = (1/M) \sum_{i=1}^M \Gamma_i$, each image differs from the average by $\phi_i = \Gamma_i - \Psi$ called mean centered image.

Covariance Matrix

The covariance matrix defines the shape of the data. Diagonal spread is captured by the covariance while axis aligned spread is captured by the variance. The covariance matrix defines both the spread (variance) and the orientation(covariance)of data. We should try to find the vector which points into the direction of the largest spread of the data and whose magnitude equals the spread in this direction. If the covariance matrix of our data is a diagonal matrix, such that covariances are zero, then variances must be equal to the eigen values. If it is not diagonal such that the covariances are not zero then the situation is more complicated. The eigen value still represent the variance magnitude in the direction of the largest spread of the data. A covariance matrix is constructed as $C = AA^T$ where $A = [\phi_1, \phi_2, \dots, \phi_M]$ of size $N^2 \times N^2$. In statistics C captures the correlation between all possible pairs of measurements for which the diagonal and off diagonal terms are called the variance and covariance respectively. The correlation value reflects the noise and redundancy in the measured data. Eigen vectors corresponding to AA^T can easily be calculated with reduced dimensionality where AX_i is the eigen vector and λ_i is the eigen value.

Eigenspace

The eigen vectors of the covariance matrix resemble the input image but look ghostly called eigenface. Eigen vectors correspond to each eigen face and discard the faces for which eigen values are zero thus reducing the eigen space to an extent. A face image can be projected into the face space by $\omega_i = U_i (\Gamma_i - \Psi)$; $i=1,2,\dots,M$, where U_i is the i^{th} eigenface. The weight is obtained as above form a vector, $\Omega_i = [\omega_1, \omega_2, \dots, \omega_M]$.

CLASSIFICATION

The Classification in an OCR system is the main decision making stage in which the features extracted from a pattern are compared to those of the model set. Based on the features, classification attempts to identify the pattern as a member of a certain class. When classifying a pattern, classification often produces a set of hypothesized solutions instead of generating a unique solution. The (subsequent) post-processing stage uses higher level information to select the correct solution. When input image is presented to HCR system, its features are extracted and given as an input to the trained classifier. Classifiers compare the input feature with stored pattern and find out the best matching class for input. Euclidean distance between two vectors of size N is given by: $ED = \sqrt{\sum_{i=1}^N (A_i - B_i)^2}$, Euclidean distance

between two dimensional images can be given by: $ED = \sqrt{\sum_{i=1}^M \sum_{j=1}^N (A(i,j) - B(i,j))^2}$

Testing Sample Classification

- a) Read the text image and separate the character from it.
- b) Calculate the feature vector of the test face. The test image is transformed into its eigen vector components. First we compare the line of our image with mean image and multiply their difference with each eigen vectors. Each value would represent a weight and save on a vector Ω^T .

$$\Omega_{test} = U_i^T (\Gamma_{test} - \Psi) \quad \Omega^T_{test} = [\omega_1, \omega_2, \dots, \omega_M]$$

- c) Compute the average distance (Euclidean distance) between test feature vector and all the training feature vectors.

Mathematically recognition is finding the minimum Euclidean distance $\epsilon_M = \sqrt{(\Omega_{test} - \Omega_i)^2}$ where $i=1,2, \dots, M$.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

The Euclidean distance between two weight vectors thus provides a measurement of similarity between the corresponding images.

d) The class with minimum Euclidean distance shows similarity to test image.

All methods for handwritten character, word or sentence recognition need to be trained. As a rule of thumb, the larger the training set, the better is the recognition performance of the system. This empirical finding has been confirmed in a number of experiments

Template Matching

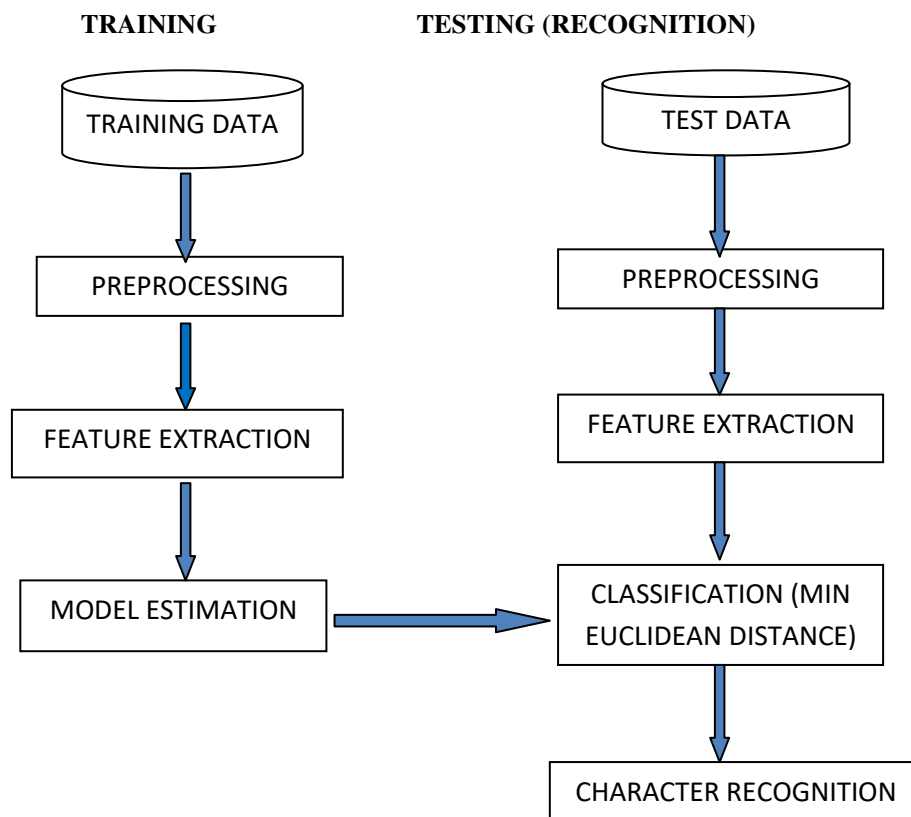


Fig Template matching - Block diagram

Template matching is one of the simplest approaches. In this many templates of each word are maintained for an input image, error or difference with each template is computed. The symbol corresponding to minimum error is output. The technique works effectively for recognition of standard fonts, but gives poor performance with hand written characters. Training database consists of characters of English language with some variation in writing style and shape so that there is no problem in the classification stage. Test database consists of the images of the characters to be identified.

Preprocessing

Preprocessing includes normalization, centering and whitening before ICA is applied. Noise is removed from images of train data base and test data base by the normalisation of image. Data centering makes the signals centred in zero $x_i \leftarrow x_i - E[x_i]$ for each i . In Whitening we transform the x 's linearly so that the \tilde{x} are white.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Feature Extraction

Reduce the amount of data by extracting relevant information results in a vector of scalar values. A matrix ($M \times M$) is calculated for each image of train database and eigen vectors and eigen values for this matrix are found. M eigen vector is chosen such that it has the highest associated eigen values. Similarly feature vector is created for each character to be identified in the test database.

Model Estimation

Estimate a model for each class of the training data from the finite set of feature vectors

Classification

Feature vector of a character to be recognised is compared with the stored feature vector of the train database and the character with minimum Euclidean distance is recognised. In this way each letter of the text is recognised.

Text Recognition

Text is identified by inserting the estimated spacing in between the characters recognized.

TEXT TO VOICE CONVERSION

Speech Application Programming Interface (SAPI)

SAPI is used for text to voice conversion. SAPI which was developed by Microsoft allows the use of speech recognition and speech synthesis within Windows applications. To date, a number of versions of the API have been released, which are shipped either as part of a Speech SDK, or as part of the Windows OS itself. Applications that use SAPI include Microsoft Office, Microsoft Agent and Microsoft Speech Server. There have been two main 'families' of the Microsoft Speech API. SAPI versions 1 through 4 are all similar to each other, with extra features in each newer version. SAPI 5 however was a completely new interface, released in 2000. Since then several sub-versions of this API have been released.

Components of SAPI

- Voice Command – high level objects for command & control of speech recognition
- Voice Dictation - high-level objects for continuous dictation of speech recognition
- Voice Talk - high-level objects for speech synthesis
- Voice Telephony - objects for writing telephone speech applications
- Direct Speech Recognition - objects for direct control of recognition engine
- Direct Text to Speech - objects for direct control of synthesis engine
- Audio objects - for reading to and from an audio device or file

Applications of SAPI

- Microsoft Windows XP Tablet PC Edition includes SAPI 5.1 and speech recognition engines 6.1 for English, Japanese, and Chinese languages (simplified and traditional)
- Windows Speech Recognition in Windows Vista
- Microsoft Narrator in Windows 2000 and later Windows operating systems
- Microsoft Office XP and Office 2003
- Microsoft Excel 2002, Microsoft Excel 2003, and Microsoft Excel 2007 for speaking spreadsheet data
- Microsoft Voice Command for Windows Pocket PC and Windows Mobile
- Microsoft Plus Voice Command for Windows Media Player
- Adobe Reader uses voice output to read document content
- Cool Speech, text-to-speech application that reads text aloud from a variety of sources

SOFTWARE TOOL

MATLAB

MATLAB is a well-known software package that is widely used not only in engineering fields but also in other fields, including economic and biomechanical systems. MATLAB has many advantages for control system design and analysis. Important features include toolboxes for specific applications and a user-friendly programming environment.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

MATLAB is one of the most famous numerical computation software. Graphical User Interface of System Identification Toolbox for MATLAB 107 has a C-like programming environment and has three distinctive features.

1. Automatic storage allocation: Variables in MATLAB need not be declared prior to being assigned. Moreover, MATLAB expands the dimensions of arrays in order for assignments to make sense.
2. Functions with variable arguments lists: MATLAB contains a large collection of functions. They take zero or more input arguments and return zero or more output arguments. MATLAB enforces a clear distinction between input and output. Functions can support a variable number of input and output arguments, so that on a given call not all arguments need be supplied.
3. Complex arrays and arithmetic: The fundamental data type is a multi-dimensional array of complex numbers. Important special cases are matrices, vectors and scalars. All computation in MATLAB is performed in floating-point arithmetic, and complex arithmetic is automatically used when the data is complex.

Graphical User Interface Development Environment (GUIDE)

A graphical user interface provides the user with a familiar environment in which to work. This environment contains push buttons, toggle buttons, lists, menus, text boxes, and so forth, all of which are already familiar to the user, so that he or she can concentrate on using the application rather than on the mechanics involved in doing things. However, GUIs are harder for the programmer because a GUI-based program must be prepared for mouse clicks (or possibly keyboard input) for any GUI element at any time. Such inputs are known as events, and a program that responds to events is said to be event driven.

The three principal elements required to create a MATLAB Graphical User Interface are

1. Components. Each item on a MATLAB GUI (push buttons, labels, edit boxes, etc.) is a graphical component. The types of components include graphical controls (push buttons, edit boxes, lists, sliders, etc.), static elements (frames and text strings), menus, and axes. Graphical controls and static elements are created by the function `uicontrol`, and menus are created by the functions `uimenu` and `uicontextmenu`. Axes, which are used to display graphical data, are created by the function `axes`.
2. Figures. The components of a GUI must be arranged within a figure, which is a window on the computer screen. In the past, figures have been created automatically whenever we have plotted data. However, empty figures can be created with the function `figure` and can be used to hold any combination of components.
3. Call backs. Finally, there must be some way to perform an action if a user clicks a mouse on a button or types information on a keyboard. A mouse click or a key press is an event, and the MATLAB program must respond to each event if the program is to perform its function. For example, if a user clicks on a button, that event must cause the MATLAB code that implements the function of the button to be executed. The code executed in response to an event is known as a call back. There must be a callback to implement the function of each graphical component on the GUI.

MATLAB's GUIDE provides a rich set of tools for incorporating GUIs in M functions. Using GUI, the process of laying out a GUI (ie its button, pop up menus etc) and programming the operation of the GUI are divided into two easily managed and independent tasks. The resulting graphical M function is composed of two identically named files,

1. A file with extension `.fig` called FIG file contains a complete graphical description of all the functions of GUI objects and their spatial arrangements. A FIG file contains binary data that need not be parsed when the GUI based M function is executed.
2. A file with extension `.m` called a GUI M file contains the code that controls the GUI operation. This file include functions that are called when the GUI is launched and excited and call back functions that are executed when a user interacts with GUI objects eg. Pushing a button.

GUI functions

- `uicontrol` - Create user interface control
- `uimenu` - Create user interface menu / `ginput` - Graphical input from mouse
- `dragrect` - Drag XOR rectangles with mouse / `rbbox` - Rubberband box
- `selectmoveresize` - Interactively select, move, resize, or copy objects
- `waitforbuttonpress` - Wait for key/buttonpress over figure
- `waitfor` - Block execution and wait for event / `uiwait` - Block execution
- `uiresume` - Resume execution of blocked M-file



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Hardware Requirements:

Operating Systems	Processors	Disk Space	RAM
Windows 8.1 Windows 8 Windows 7 Service Pack 1 Windows Vista Service Pack 2 Windows XP Service Pack 3 Windows XP x64 Edition Service Pack 2 Windows Server 2012 Windows Server 2008 R2 Service Pack 1 Windows Server 2008 Service Pack 2 Windows Server 2003 R2 Service Pack 2	Any Intel or AMD x86 processor supporting SSE2 instruction set*	1 GB for MATLAB only, 3-4 GB for a typical installation	1024 MB (At least 2048 MB recommended)

Fig Hardware Requirements for 32 Bit MATLAB

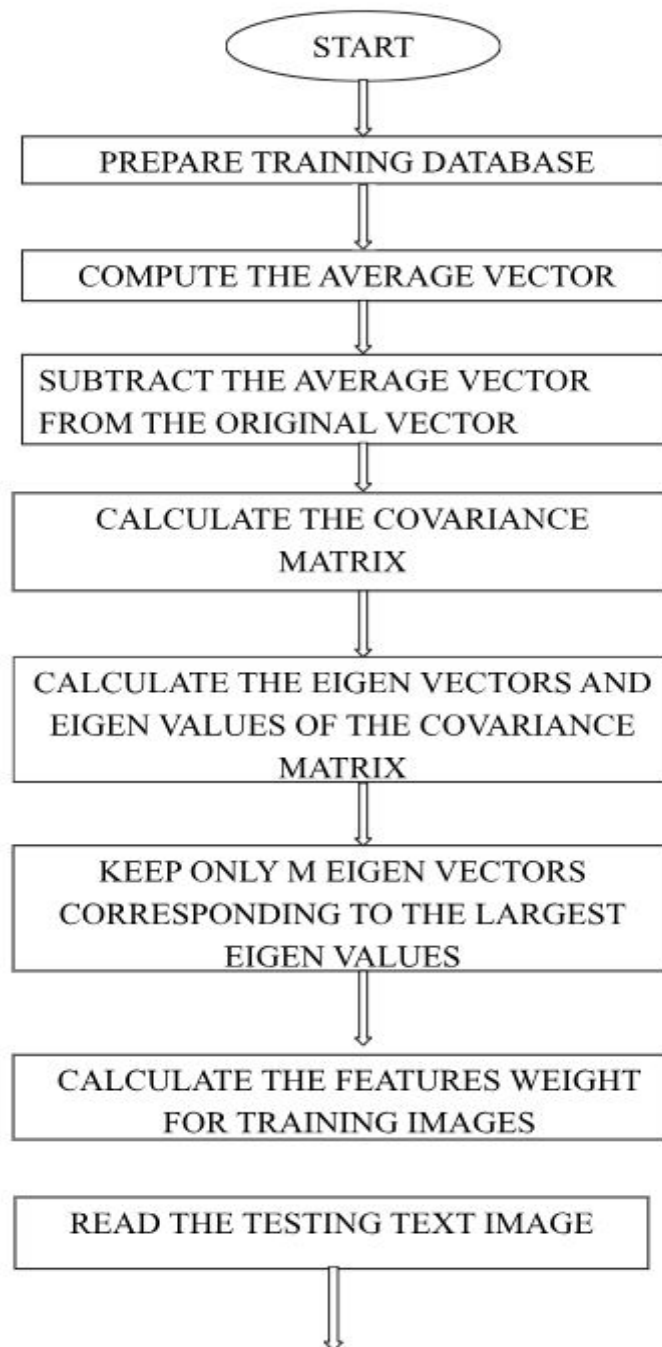
International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

III. IMPLEMENTATION, RESULT AND ANALYSIS

IMPLEMENTATION



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

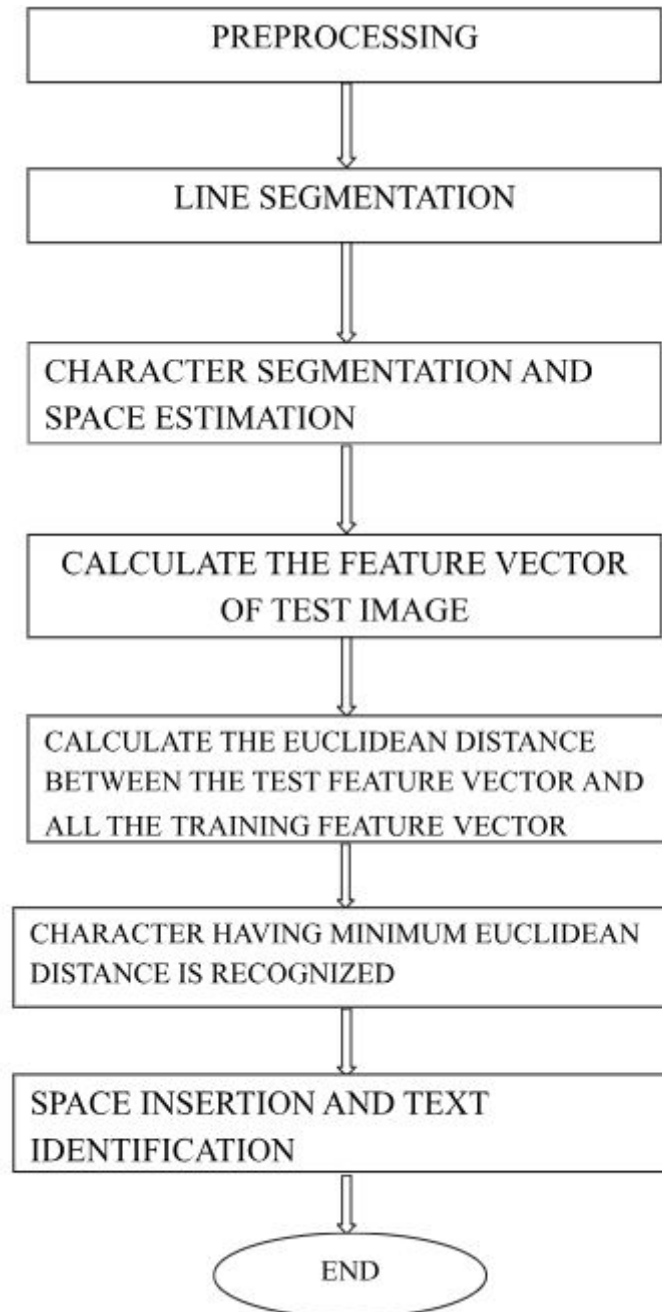


Fig Implementation flow chart

IV. RESULTS AND ANALYSIS

Testing Parameters

Matlab R2013a is used for coding. The camera image is converted into gray scale image as gray scale images are easier for applying computational techniques in image processing. The gray scale image is scaled for a particular pixel size as

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

48 x 84 because input images can be of different matrix size when taken for recognition. The camera image pixel size is 320 x 240. The input image is resized into 48 x 84 pixel size

Image Acquisition

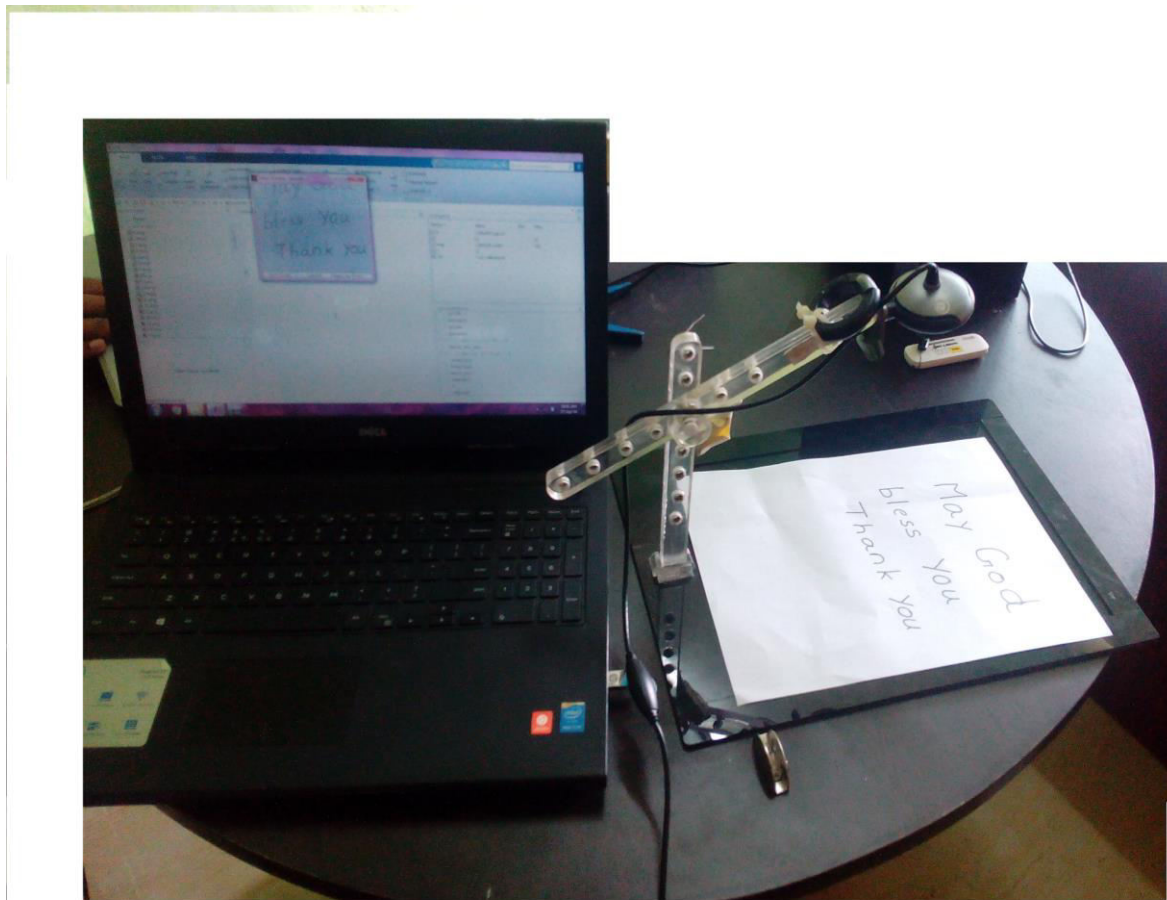


Fig Microsoft HD Camera
Preprocessing
Inverted Binarised Image

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

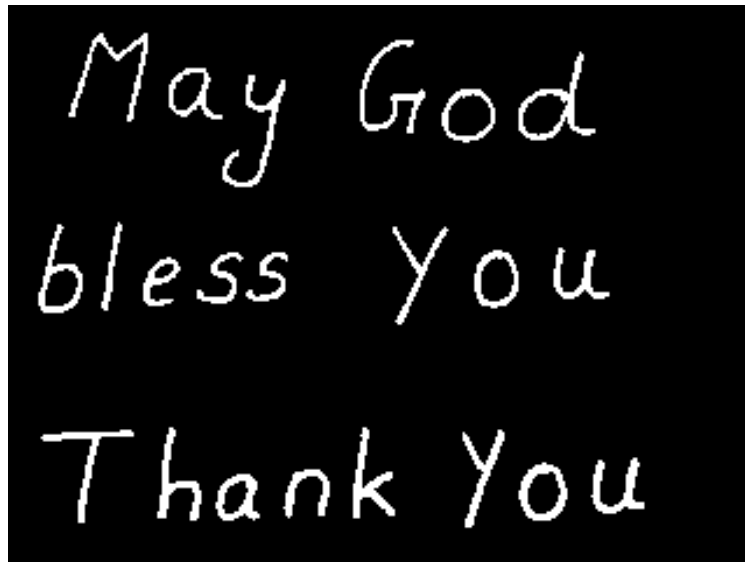


Fig. Inverted Binarised Image

Segmentation

Line Segmentation

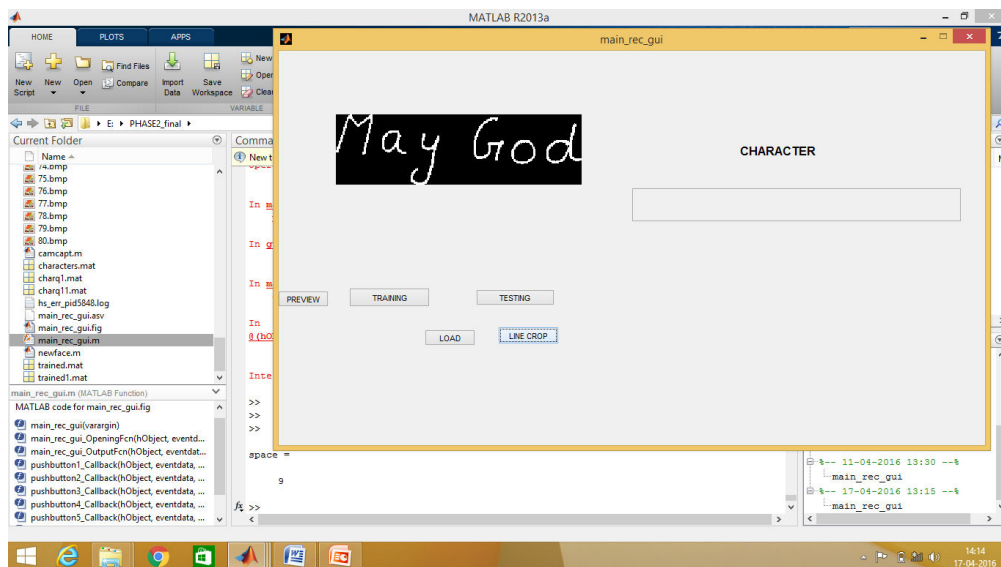


Fig Line Cropped Image

Training Dataset

It consists of cropped inverted binarised images of 17 English alphabets. Variations in input image orientation and size will change the output, therefore input images for different orientation and size are also taken for recognition.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Training set

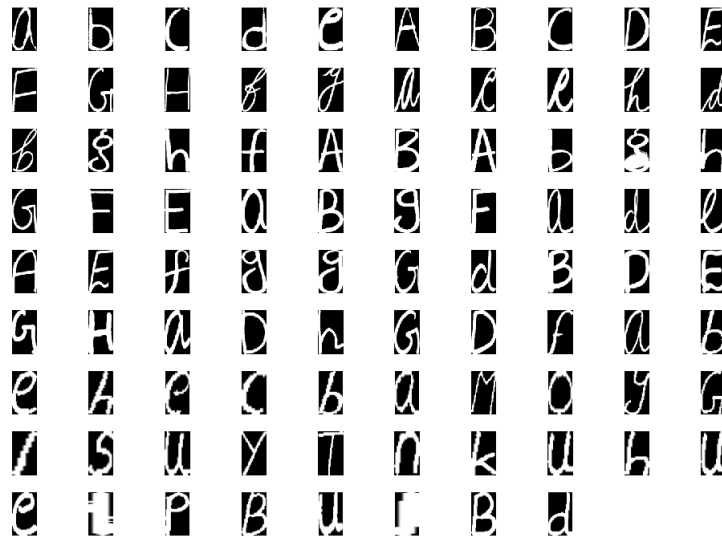


Fig.Database of trained characters

Normalized Training Set

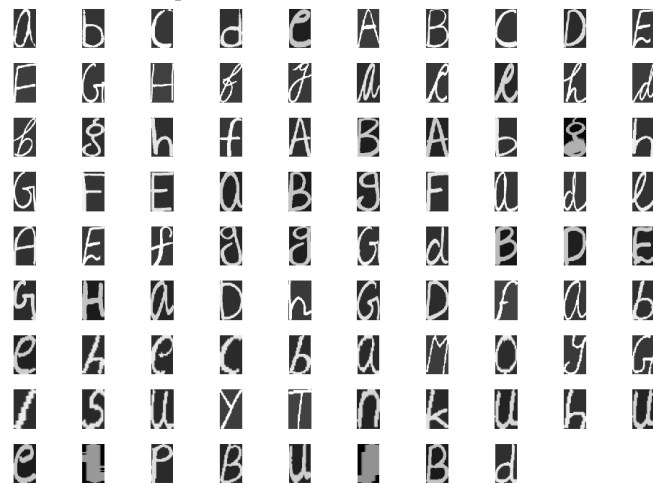


Fig. Database of trained characters after normalisation

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

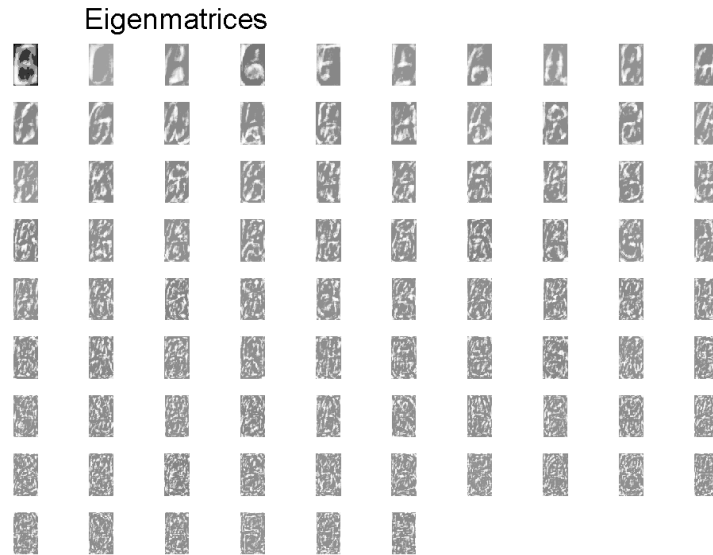


Fig. Eigenmatrices of trained characters

Character Recognition

Character Segmentation, Feature Extraction, Character Recognition and Space Estimation

The following datas represent the Euclidean distance between the calculated weight of 88 trained images of English alphabets and the feature weight of the testing image of alphabet 'M'.

Columns 1 through 9

7.5773 7.6214 7.4978 7.2939 7.3310 6.9345 7.6793 7.7421 7.0602

Columns 10 through 18

6.9787 6.8525 7.8364 6.5822 7.1627 7.2895 8.1546 7.7142 7.9997

Columns 19 through 27

8.1307 7.5927 7.6733 7.5285 7.9008 8.1008 7.3019 7.7407 6.7943

Columns 28 through 36

7.8404 7.9388 7.7787 7.9108 7.5414 7.6606 7.8539 7.6709 7.1348

Columns 37 through 45

7.4768 7.9675 7.4531 7.3215 7.3931 6.9787 8.0736 7.6542 6.5988

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Columns 46 through 54

7.9181 7.7547 7.4547 7.3149 7.9109 7.7792 7.5582 7.8363 7.6315

Columns 55 through 63

7.6478 7.9183 7.3745 7.3269 7.7339 7.2524 7.5541 7.6864 7.2006

Columns 64 through 72

7.2833 7.7922 8.0610 2.5397 7.5285 7.1781 7.2213 7.9368 7.5213

Columns 73 through 81

8.1741 7.2734 6.9796 6.2706 7.2887 8.0504 6.9388 8.0504 7.7370

Columns 82 through 88

7.8231 7.2477 7.0567 8.2365 7.0817 7.4851 7.7103

$p = M$

Finally the word having minimum Euclidean distance is recognized

M space = 4 (spacing between two letters)

Estimated Space Vector

May God

9 4 32 7 10 10

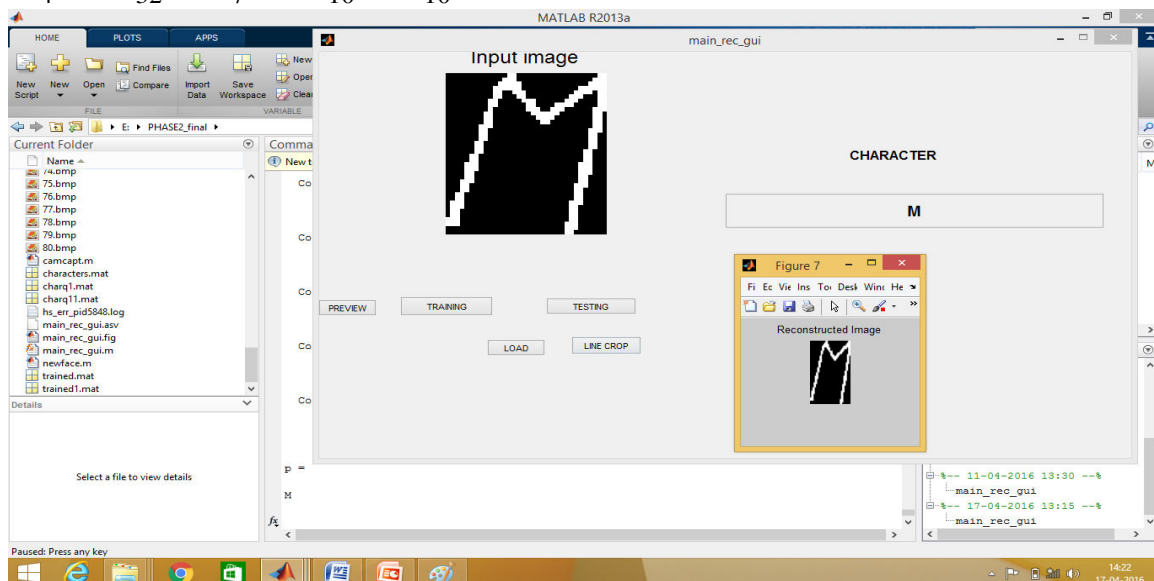


Fig. Character recognised image of M

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

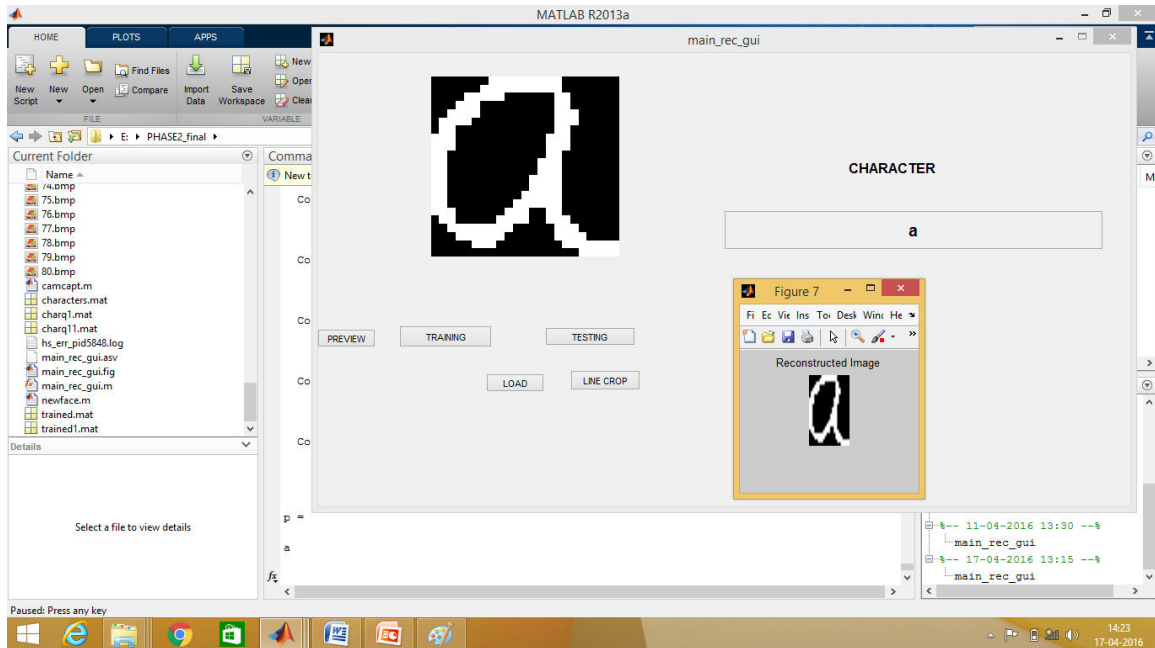


Fig Character recognised image of A

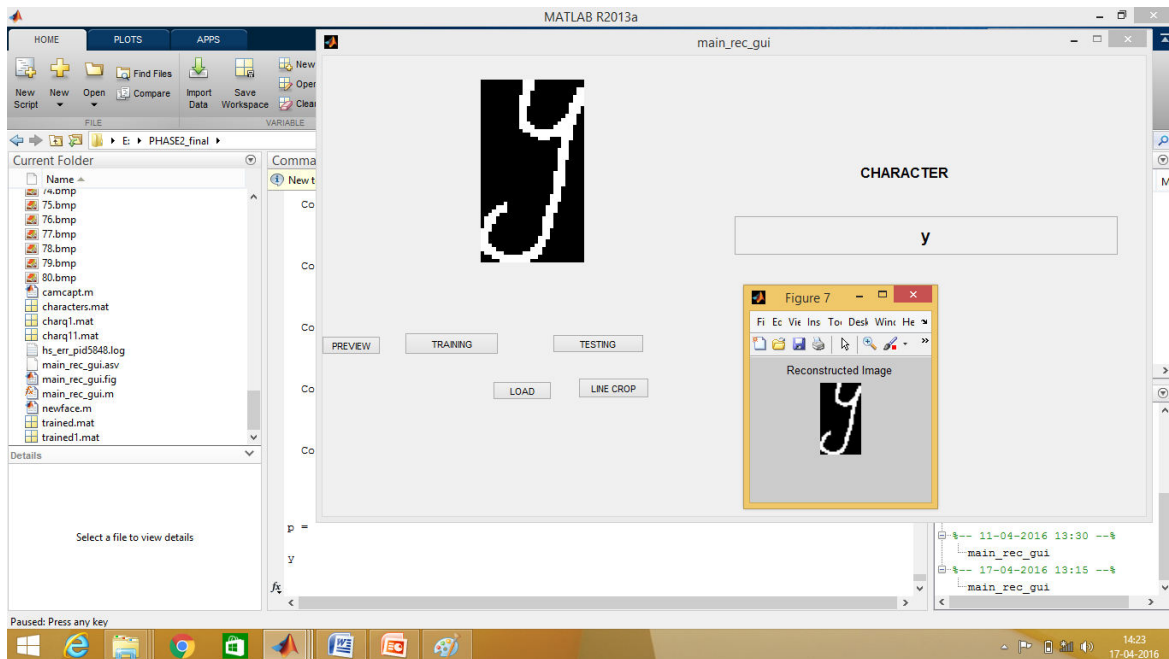


Fig Character recognised image of Y

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

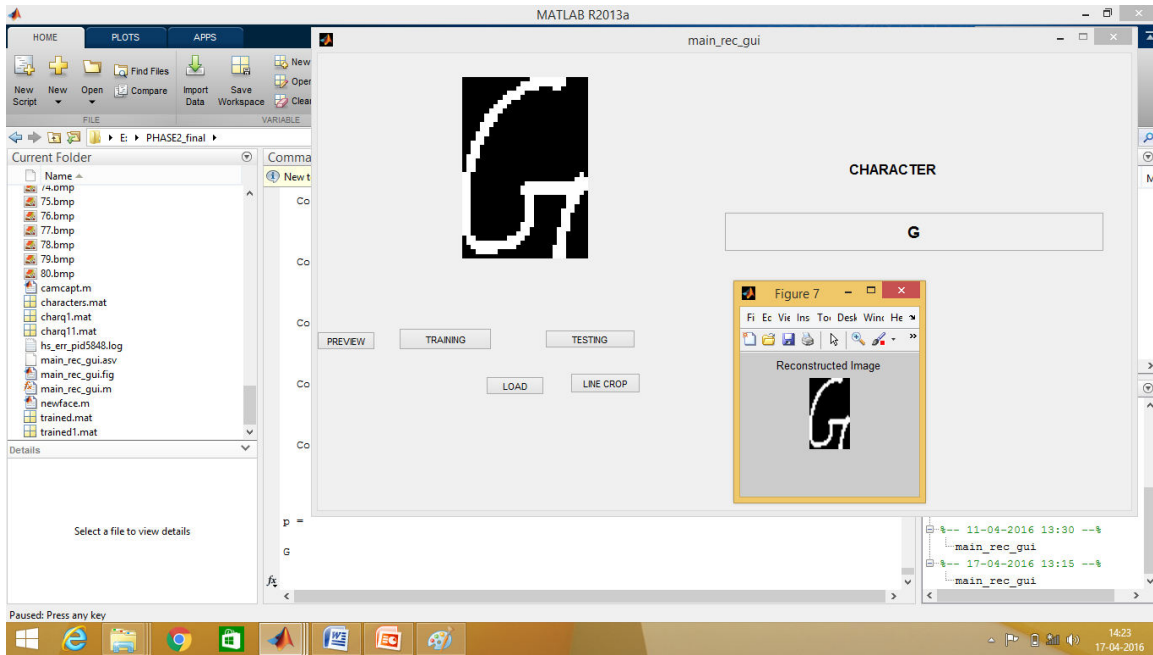


Fig Character recognised image of G

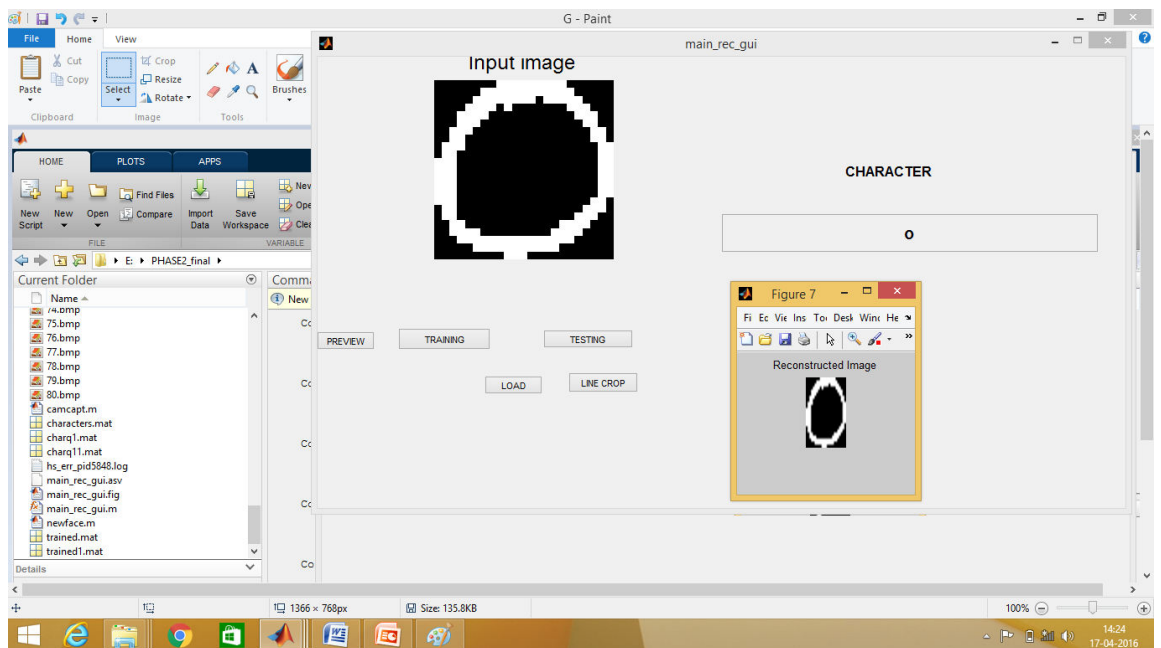


Fig Character recognised image of O

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

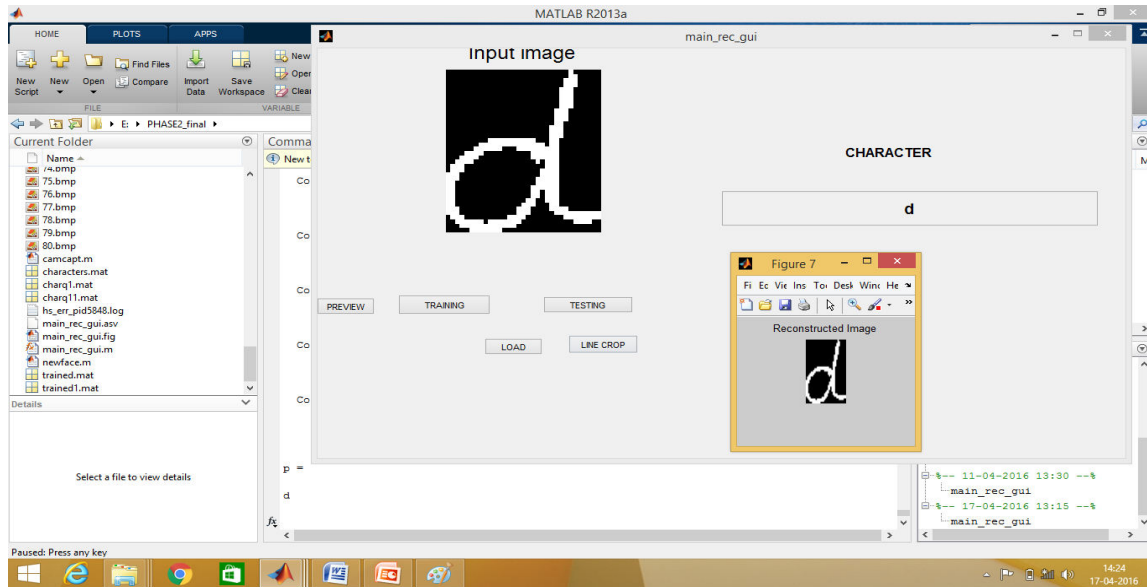
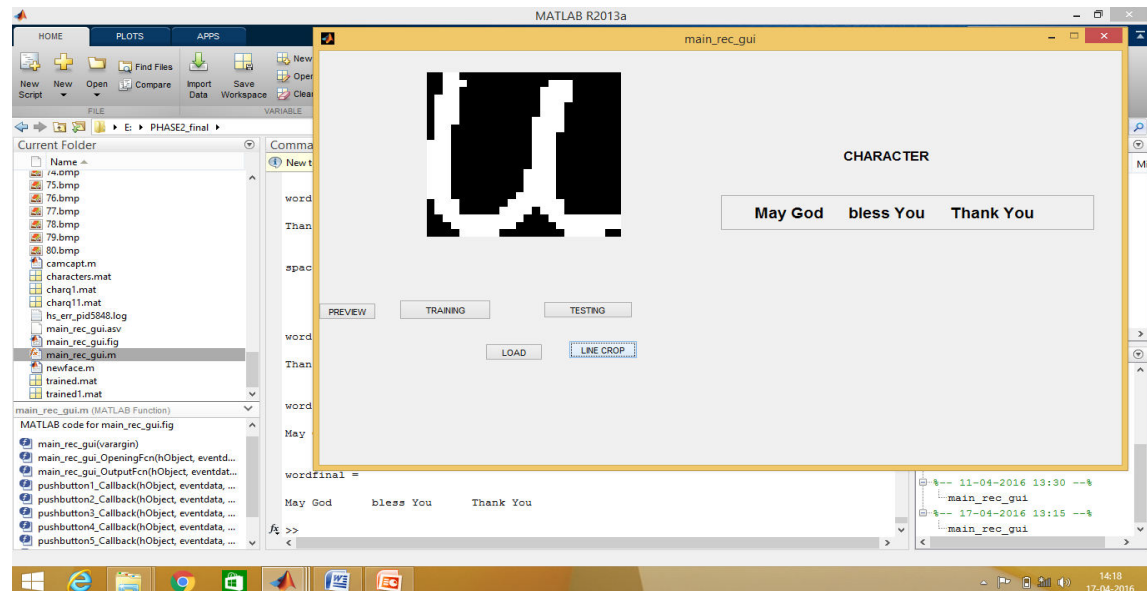


Fig Character recognised image of D

Text Recognition



Output Image

Text to Speech Conversion

Speech Application Programming Interface is used for text to voice conversion.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

V. CONCLUSION

The focus of attention in handwriting recognition has been shifting from isolated character recognition to more complex tasks, such as recognition of words and unconstrained text. It can be expected that a significant percentage of future systems for cursive handwriting recognition will be personal systems serving a single user. Such systems will perform best when they are trained with data provided by the future user. However it can be quite cumbersome for an individual to provide a sufficiently large body of handwritten samples to train a system. Natural language processing techniques, and machine translation are very promising to improve the recognition performance of today's handwriting recognition procedures. In addition they would naturally lead to tools for content based search and retrieval in the context of archives of handwritten texts. The ultimate goal of handwriting recognition is to have machines which can read any text with the same recognition accuracy as humans but at a faster rate. Sentence recognition is important in a number of future applications, for example, the transcription of personal notes, faxes, and letters, or the electronic conversion of historical handwritten archives in the context of the creation of digital libraries. As handwritten text is an important means for day to day communication this paper on text recognition shall be useful to our society at large and particularly to the visually challenged as voice enhancement features are also included.

REFERENCES

1. Aapo Hyvärinen and Erkki Oja (2000), 'Independent Component Analysis: Algorithms and Applications', Neural Networks, Vol. 13(4-5), pp. 411-430.
2. Aapo Hyvärinen, Juha Karhunen and Erkki Oja (2001), 'Independent Component Analysis', A Wiley-Interscience Publication Final version 7, John Wiley & Sons, <https://www.cs.helsinki.fi/u/ahyvarin>.
3. Algeo and John (2010), 'The Origins and Development of the English Language', pp. 182-187.
4. Amit Choudhary, Rahul Rishi and Savita Ahlawat (2013), 'Off-Line Handwritten Character Recognition using Features Extracted from Binarization Technique', 2212-6716 American Applied Science Research Institute.
5. Andreas Weingessel, Martin Natter and Kurt Hornik (1998), 'Using Independent Component Analysis for Feature Extraction and Multivariate Data Projection', University of Vienna, Vienna University of Technology, Working Paper No. 16.
6. Atishay Jain, Shashikala Tapaswi and Utpal Garain (2008), 'An Independent Component Analysis Approach for Enhancing Camera Based Low Quality Text Images', CITWORKSHOPS, IEEE 8th International Conference on Computer and Information Technology, pp. 368-373, ISBN: 978-0-7695-3242-4, <http://doi.ieeecomputersociety.org>.
7. Bellegarda E. J., Bellegarda J. R. and Kim J. H. (1994), 'On-line handwritten character recognition using parallel neural networks', IEEE International Conference on, Acoustics, Speech, and Signal Processing, pp. 605-608, ISBN: 0-7803-1775-0, <http://doi.ieeecomputersociety.org>.
8. Blumenstein M., Verma B. and Basli H (2003), 'A Novel Feature Extraction Technique for the Recognition of Segmented Handwritten Characters', Proceedings of the Seventh International Conference on Document Analysis and Recognition, 0-7695-1960-1/03 2003 IEEE.
9. Chapman and Stephen J (2001), 'MATLAB Programming for Engineers'.
10. Chi Fang and Xiaoqing Ding (2004), 'Improvement of ICA Based Probability Density Estimation for Pattern Recognition', ICPR Proceedings of the 17th International Conference on Pattern Recognition, pp. 466-469, ISBN: 0-7695-2128-2, <http://doi.ieeecomputersociety.org>.
11. Clifford C. D. (2015), 'Singular Value Decomposition & Independent Component Analysis for Blind Source Separation', J Biomedical Signal and Image Processing.
12. Gregg C. Vanderheiden, Andrew M. York, and Daniel C. Geisler, 'An alternate interface to computers for the physically handicapped -the auto-monitoring communication board', University of Wisconsin-Madison.
13. He Zhiguo, Zhong Yuquan and Cao Yudong (2013), 'Handwritten Chinese Character Recognition based on Support Vector Machine and Kernel Independent Component Analysis' School of Computer Science, Panzhihua University, China.
14. Hongfei Wang (2014), 'ATM - Awesome Translation Machine', ICDH, 5th International Conference on Digital Home, pp. 226-231, <http://doi.ieeecomputersociety.org>.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

15. Horst Bunke (2003), 'Recognition of Cursive Roman Handwriting - Past, Present and Future', ICDAR Proceedings Seventh International Conference on Document Analysis and Recognition, pp. 448, ISBN: 0-7695-1960-1, <http://doi.ieeecomputersociety.org>.
16. Impedovo S., Facchini G. and Mangini F. M. (2012), 'A New Cursive Basic Word Database for Bank-Check Processing Systems', DAS, IAPR International Workshop on, Document Analysis Systems, pp. 450-454, ISBN: 978-0-7695-4661-2, <http://doi.ieeecomputersociety.org>.
17. Jyostna Grace M. and Subhashini K. (2014), 'Image Binarization Based on ICA Approach for Optical Character Recognition', International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 3, No. 8.
18. Kasthuri M. and Britto Ramesh Kumar S. (2014), 'Rule Based Machine Translation System from English to Tamil', WCCCT World Congress on Computing and Communication Technologies, pp. 158-163, <http://doi.ieeecomputersociety.org>.
19. Khan, Rehan Ullah, Nasir Ahmad Khan and Khwaja Naveed (2012), 'Urdu Character Recognition using Principal Component Analysis', International Journal of Computer Applications Vol. 60, No.11.
20. Mansi Shah, Gordhan B. Jethava (2013), 'A Literature Review on Hand Written Character Recognition', Indian Streams Research Journal, Vol. 3, No. 2.
21. Marti U. V. and Bunke H. (1999), 'A Full English Sentence Database for Off-Line Handwriting Recognition', ICDAR 12th International Conference on Document Analysis and Recognition, pp. 705, ISBN: 0-7695-0318-7, <http://doi.ieeecomputersociety.org>.
22. Marti U. V. and Bunke H. (2000), 'Handwritten Sentence Recognition', ICPR, International Conference on Pattern Recognition, pp. 3467, ISBN: 0-7695-0750-6, <http://doi.ieeecomputersociety.org>.
23. Marti U. V. and Bunke H. (2001), 'Text Line Segmentation and Word Recognition in a System for General Writer Independent Handwriting Recognition', ICDAR, 12th International Conference on Document Analysis and Recognition, pp. 0159, ISBN: 0-7695-1263-1, <http://doi.ieeecomputersociety.org>.
24. Matthias Zimmermann, Jean-Cdric Chappelier and Horst Bunke (2006), 'Offline Grammar-Based Recognition of Handwritten Sentences', IEEE Transactions on Pattern Analysis & Machine Intelligence, Vol.28, No. 5, pp. 818-821, <http://doi.ieeecomputersociety.org>.
25. Michael Dunn (2011), 'Speech synthesis and recognition in .NET - Give applications a voice', Redmond Developer News.
26. Morteza Amir Rahimi and John B. Eulenberg 'A computing environment for the blind' Michigan State University, East Lansing, Michigan.
27. Nikolaos Bourbakis (2008), 'Automatic Image-to-Text-to-Voice Conversion for Interactively Locating Objects in Home Environments', ICTAI 24th International Conference on Tools with Artificial Intelligence, pp. 49-55, ISBN: 978-0-7695-3440-4, <http://doi.ieeecomputersociety.org>.
28. Rajashekararadhya S. V, and VanajaRanjan P. (2005), 'Efficient zone based feature extraction algorithm for handwritten numeral recognition of four popular south-Indian scripts', Journal of Theoretical and Applied Information Technology, pp. 1171- 1181.
29. Roman W. Swiniarski and Andrzej Skowron (2004), 'Independent Component Analysis, Principal Component Analysis and Rough Sets In Face Recognition' Department of Mathematical and Computer Sciences, San Diego State University, CA 92182, USA and Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland.
30. Rowly H., Goyal M. and J. Bennet (2002), 'The effect of large training set sizes on online Japanese Kanji and English cursive recognizers', 8th Int. Workshop on Frontiers in Handwriting Recognition, pp. 36-40.
31. Seichi Ozawa, Yoshinori Sakaguchi and Manabu Kotani (2014), 'Feature Extraction of Handwritten Characters using Supervised and Unsupervised Independent Component Analysis' Faculty of Engineering, Kobe University, Japan.
32. Smith S. (1994), 'Handwritten character classification using nearest neighbour in large databases', IEEE Trans. on Pattern Analysis and Machine Intelligence, 16:915- 919.
33. Srinivasan S. H., Ramakrishnan K. R. and Bhagavathy S. (1999), 'The Independent Components of Characters are 'Strokes'', ICDAR, 12th International Conference on Document Analysis and Recognition, pp. 414, ISBN: 0-7695-0318-7, <http://doi.ieeecomputersociety.org>.



ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

34. Tomai C., Zhang B. and Govindaraju V. (2002), 'Transcript mapping for historic handwritten document images', 8th Int. Workshop on Frontiers in Handwriting Recognition, pp. 413–418.
35. Vinciarelli A. and Bengio S. (2002), 'Offline Cursive Word Recognition using Continuous Density Hidden Markov Models Trained with PCA or ICA Features', Proceedings of 16th International Conference on Pattern Recognition, pp. 30081, ISBN: 0-7695-1695-X, <http://doi.ieeecomputersociety.org>
36. Vinciarelli A. and Bengio S. (2002), 'Writer adaptation techniques in HMM based off-line cursive script recognition', Pattern Recognition Letters, 23:905–916.
37. Vinciarelli A., Bengio and Bunke S. (2003), 'Offline Recognition of Large Vocabulary Cursive Handwritten Text', ICDAR, Proceedings of Seventh International Conference on Document Analysis and Recognition, pp. 1101, ISBN: 0-7695-1960-1, <http://doi.ieeecomputersociety.org>.
38. Yan Chen and Graham Leedham (2005), 'Independent Component Analysis Segmentation Algorithm', ICDAR, Proceedings of Eighth International Conference on Document Analysis and Recognition, pp. 680-684, ISBN: 0-7695-2420-6, <http://doi.ieeecomputersociety.org>.
39. Yousif A. El-Imam and Karima Banat (1990), 'Text-to-Speech Conversion on a Personal Computer', IEEE Micro, Vol.10, No. 4, pp. 62-74, <http://doi.ieeecomputersociety.org>.