

HAND-WRITTEN CHARACTER RECOGNITION

Using Artificial Neural Network

*A thesis submitted in partial fulfillment of under graduate program of
Bachelor of Technology*

IN

ELECTRONICS & INSTRUMENTATION ENGINEERING

By:

CHANDAN KUMAR – 110EI0240

UNDER GUIDANCE OF: Prof. (Dr.) SUKADEV MEHER

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING



NIT ROURKELA

ROURKELA – 769008 (ODISHA).

2014



**NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA**

CERTIFICATE

This is to certify that the thesis entitled “*Hand Written Character Recognition*” submitted by Chandan Kumar (110EI0240) in partial fulfilment of the requirements for the award of Bachelor of Technology Degree in *Electronics and Instrumentation Engineering* at the National Institute of Technology, Rourkela (Deemed University) has been carried out by them under my supervision.

Date:

Prof. (Dr.) Sukadev Meher

Rourkela

Dept. of E.C.E, NIT Rourkela-769008

Acknowledgment

This work would have not been possible without the guidance and the help of my final year project guide who in one way or another, contributed and extended his valuable assistance in this project.

My utmost gratitude to my project guide **Prof. (Dr.) Sukadev Meher** (HOD, ECE Dept., NIT Rourkela), my dissertation adviser whose sincerity and encouragement I will never forget. **Prof. Meher** has been my inspiration as I hurdle all the obstacles in the completion of this project work and has supported me throughout my project work with patience and knowledge.

Next I would like to thank all the M.Tech. students and the faculty present in the Image Processing Laboratory for their support and timely help.

Above all, I would like to thank all my friends and family members whose direct and indirect support helped me complete my project in time. The thesis would have been impossible without their perpetual moral support.

Date:
Rourkela

Chandan Kumar (110EI0240)
Electronics & Instrumentation

Contents

| | |
|--|-----------|
| 1. Introduction | 6 |
| 1.1 Objective of the Project | 8 |
| 1.2 Approach | 8 |
| 2. Artificial Neural Network | 9 |
| 2.1 Introduction | 10 |
| 2.2 Creating and training of network | 11 |
| 3. Image processing involved in character recognition . | 19 |
| 3.1 Pre-processing of sample image | 20 |
| 3.2 Feature extraction | 22 |
| 4. Simulation and Results | 24 |
| 4.1 Simulation | 25 |
| 4.2 Results | 32 |
| 5. Conclusion and future improvements | 33 |
| 5.1 Conclusion | 34 |
| 5.2 Future scope of this project | 34 |

ABSTRACT

In today's world advancement in sophisticated scientific techniques is pushing further the limits of human outreach in various fields of technology. One such field is the field of character recognition commonly known as OCR (Optical Character Recognition).

In this fast paced world there is an immense urge for the digitalization of printed documents and documentation of information directly in digital form. And there is still some gap in this area even today. OCR techniques and their continuous improvisation from time to time is trying to fill this gap. This project is about devising an algorithm for recognition of hand written characters also known as HCR (Handwritten Character Recognition) leaving aside types of OCR that deals with recognition of computer or typewriter printed characters.

A novel technique is proposed for recognition English language characters using Artificial Neural Network including the schemes of feature extraction of the characters and implemented. The persistency in recognition of characters by the AN network was found to be more than 90% of times.

CHAPTER 1

INTRODUCTION

CHAPTER 1

Introduction

This project, 'Handwritten Character Recognition' is a software algorithm project to recognize any hand written character efficiently on computer with input is either an old optical image or currently provided through touch input, mouse or pen.

Character recognition, usually abbreviated to optical character recognition or shortened OCR, is the mechanical or electronic translation of images of handwritten, typewritten or printed text (usually captured by a scanner) into machine-editable text. It is a field of research in pattern recognition, **artificial intelligence** and machine vision. Though academic research in the field continues, the focus on character recognition has shifted to implementation of proven techniques. Optical character recognition is a scheme which enables a computer to learn, understand, improvise and interpret the written or printed character in their own language, but present correspondingly as specified by the user. Optical Character Recognition uses the image processing technique to identify any character computer/typewriter printed or hand written. A lot of work has been done in this field. But a continuous improvisation of OCR techniques is being done based on the fact that algorithm must have higher accuracy of recognition, higher persistency in number of times of correct prediction and increased execution time.

The idea is to device efficient algorithms which get input in digital image format. After that it processes the image for better comparison. Then after the processed image is compared with already available set of font images. The last step gives a prediction of the character in percentage accuracy.

1.1 OBJECTIVE OF THE PROJECT

The objective of this project is to identify handwritten characters with the use of neural networks. We have to construct suitable neural network and train it properly. The program should be able to extract the characters one by one and map the target output for training purpose. After automatic processing of the image, the training dataset has to be used to train “classification engine” for recognition purpose. The program code has to be written in MATLAB and supported with the usage of Graphical User Interface (GUI).

1.2 APPROACH

To solve the defined handwritten character recognition problem of classification we used MATLAB computation software with Neural Network Toolbox and Image Processing Toolbox add-on. The computation code is divided into the next categories:

- Pre-processing of the image
- Feature extraction
- Creating an Artificial Neural Network
- Training & Testing of the network
- Recognition

CHAPTER 2

ARTIFICIAL NEURAL

NETWORK

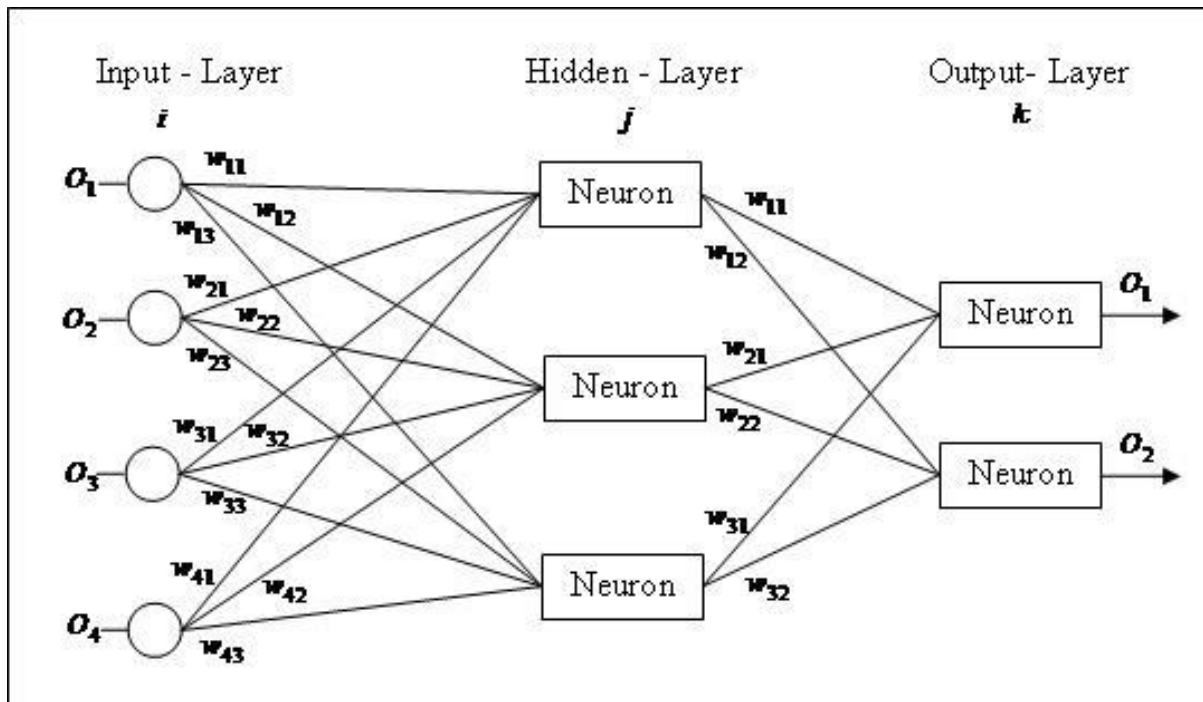
Artificial Neural Network

2.1 INTRODUCTION

An early phase of Neural Network was developed by Warren McCulloch and Walter Pitts in 1943 which was a computational model based on Mathematics and algorithm. This model paved the way for research which was focused on the application of Neural Networks in Artificial Intelligence.

Artificial neural network is basically a mesh of large number of interconnected cells. The arrangement of cells are such that each cell receives an input and drives an output for subsequent cells. Each cell has a pre-defined

The diagram below is a block diagram that depicts the structure and work flow of a created **Artificial Neural Network**. The neurons are interconnected with each other in a serial manner. The network consist of a number of hidden layers depending upon the resolution of comparison of inputs with the dataset.

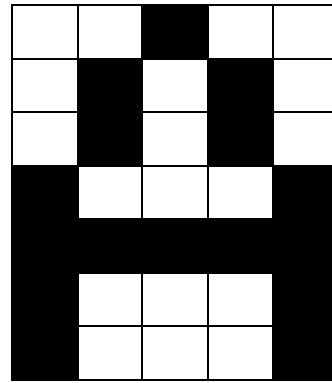


2.2 CREATING AND TRAINING OF NETWORK

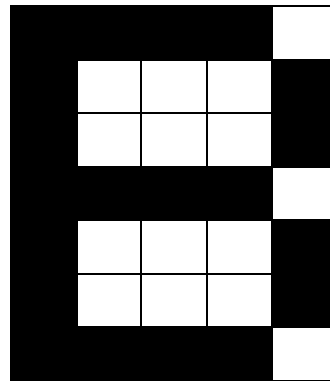
In case of character recognition we have to create a 2D vector of character images which can be fed to the network as ideal set of input variables. In our case there is a total of 26 capital English letters which we are to recognize.

Below is a set of characters written in binary form of 7x5 sized matrix of 26 capital English letters:

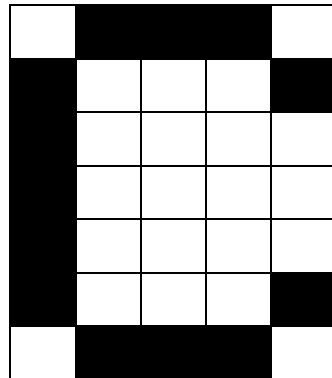
| | | | | |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |



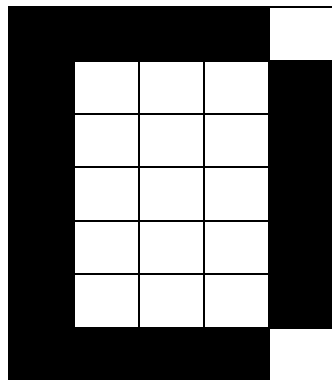
| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |



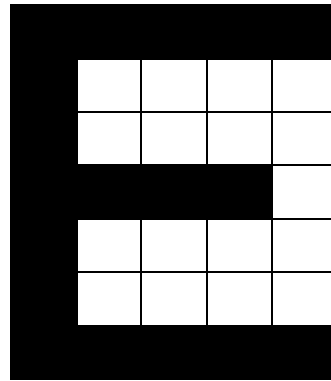
| | | | | |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |



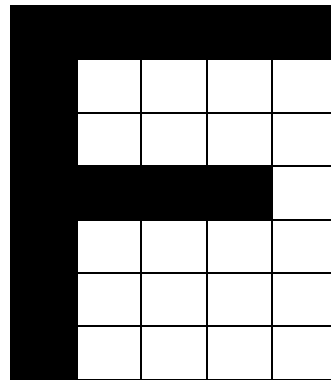
| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |



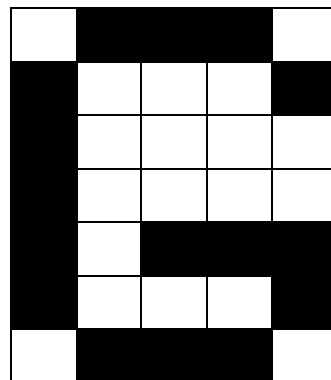
| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |



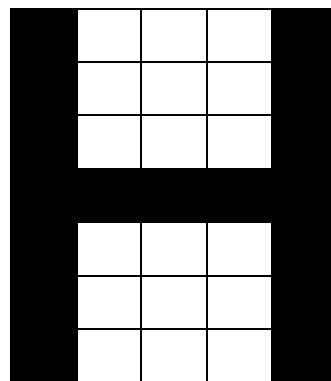
| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |



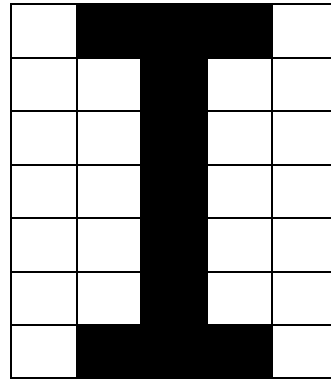
| | | | | |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |



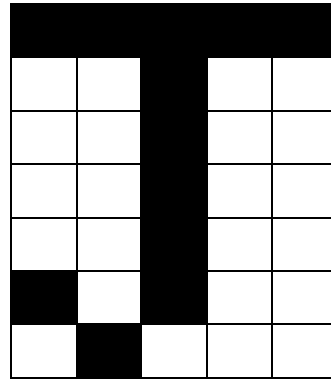
| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |



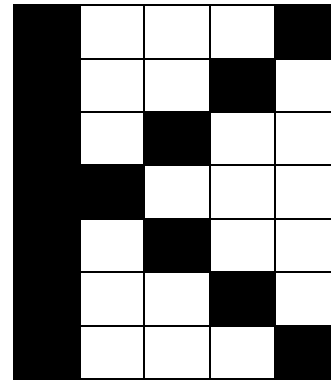
| | | | | |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |



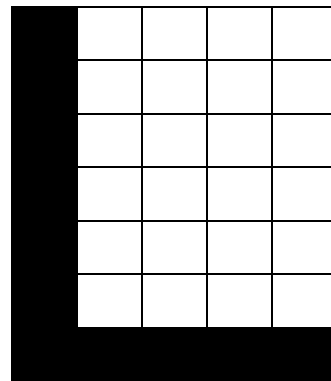
| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |



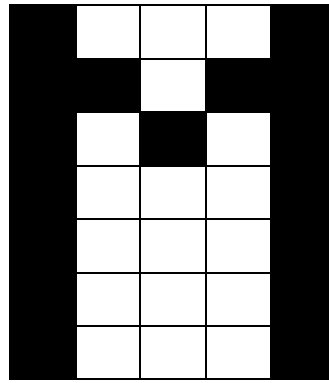
| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |



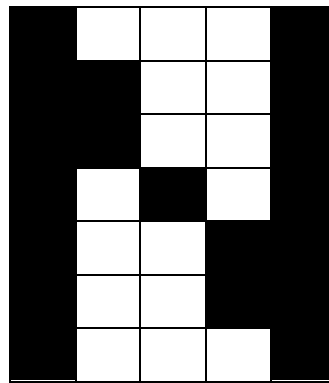
| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |



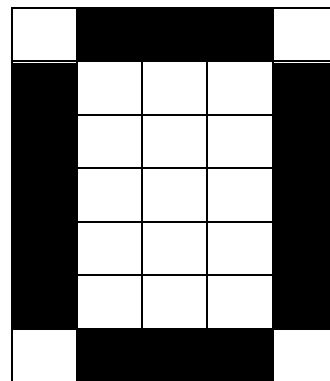
| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |



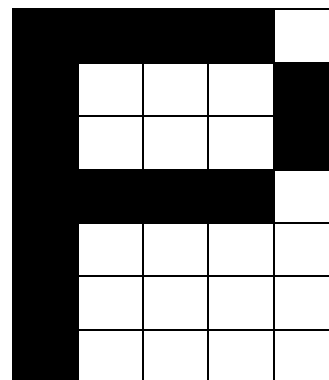
| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |



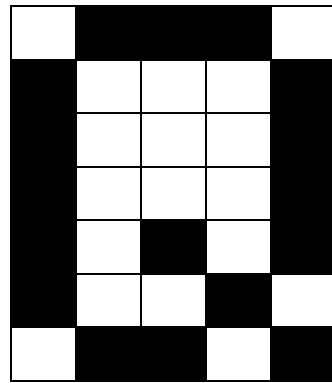
| | | | | |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |



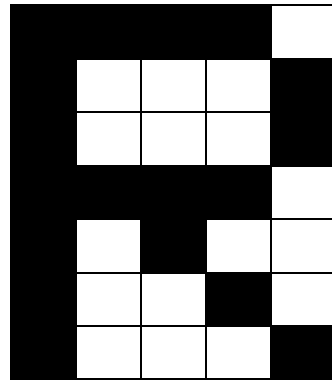
| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |



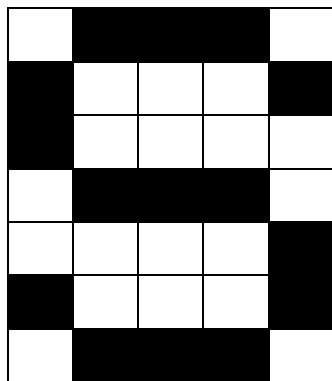
| | | | | |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |



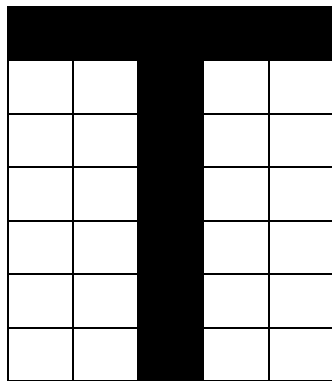
| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |



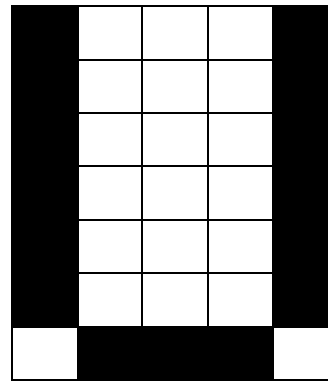
| | | | | |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |



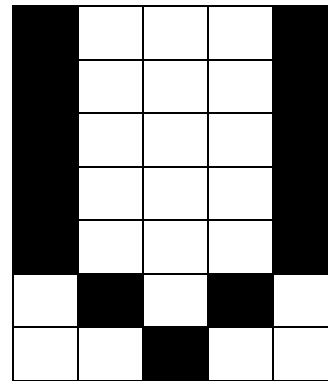
| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |



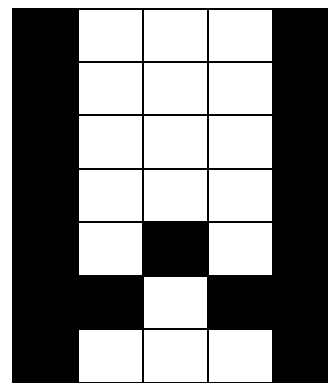
| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |



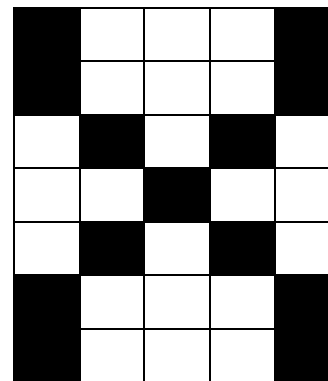
| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |



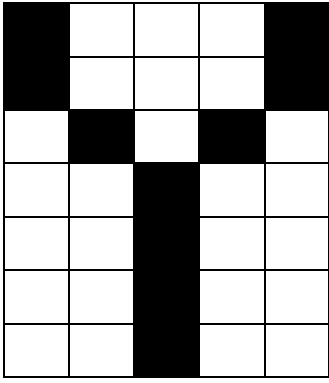
| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |



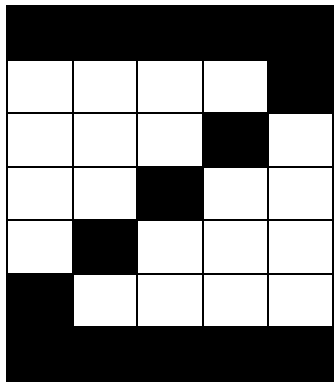
| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |



| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |



| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |



CHAPTER 3

IMAGE PROCESSING

INVOLVED IN

CHARACTER

RECOGNITION

Image Processing involved in Character Recognition

3.1 PRE-PROCESSING OF SAMPLE IMAGE

Pre-processing of the sample image involves few steps that are mentioned as follows:

Grey-scaling of RGB image

Grey-scaling of an image is a process by which an RGB image is converted into a black and white image. This process is important for Binarization as after grey-scaling of the image, only shades of grey remains in the image, binarization of such image is efficient

Binarization

Binarisation of an image converts it into an image which only have pure black and pure white pixel values in it. Basically during binarization of a grey-scale image, pixels with intensity lower than half of the full intensity value gets a zero value converting them into black ones. And the remaining pixels get a full intensity value converting it into white pixels.

Inversion

Inversion is a process in which each pixel of the image gets a colour which is the inverted colour of the previous one. This process is the most important one because any character on a sample image can only be extracted efficiently if it contains only one colour which is distinct from the background colour. Note that it is only required if the objects we have to identify if of darker intensity on a lighter background.

The flow chart shown below illustrates the physical meaning of the processes that are mentioned above:

RGB => Grey-scaling => Binarization => Inversion



3.2 FEATURE EXTRACTION

Features of a character depicts the morphological and spatial characteristics in the image. Feature extraction is a method of extracting of features of characters from the sample image. There are basically two types of feature extraction:

- Statistical feature extraction
- Structural feature extraction

Statistical feature extraction

In this type of extraction the extracted feature vector is the combination of all the features extracted from each character. The associated feature in feature vector of this type of extraction is due to the relative positions of features in character image matrix.

Structural feature extraction

This is a primitive method of feature extraction which extracts morphological features of a character from image matrix. It takes into account the edges, curvature, regions, etc. This method extracts the features of the way character are written on image matrix.

The different methods used for feature extraction are

- Piecewise –linear regression
- Curve-fitting
- Zoning
- Chain code, etc.

The functions that are used in feature extraction are:

Indexing and labelling

This is a process by which distinct characters in an image are indexed and labelled in an image. Thus helps in classification of characters in image and makes feature extraction of characters simple.

Boxing and Cropping

This is a process of creating a boundary around the characters identified in an image. This helps by making cropping of characters easier. After boxing the characters are cropped out for storing them as input variables for recognition.

Reshaping and Resizing

Reshaping is done to change the dimensions of the acquired character in desired shape. Resizing is done to reduce the size of characters to a particular minimum level.

CHAPTER 4

SIMULATION AND

RESULTS

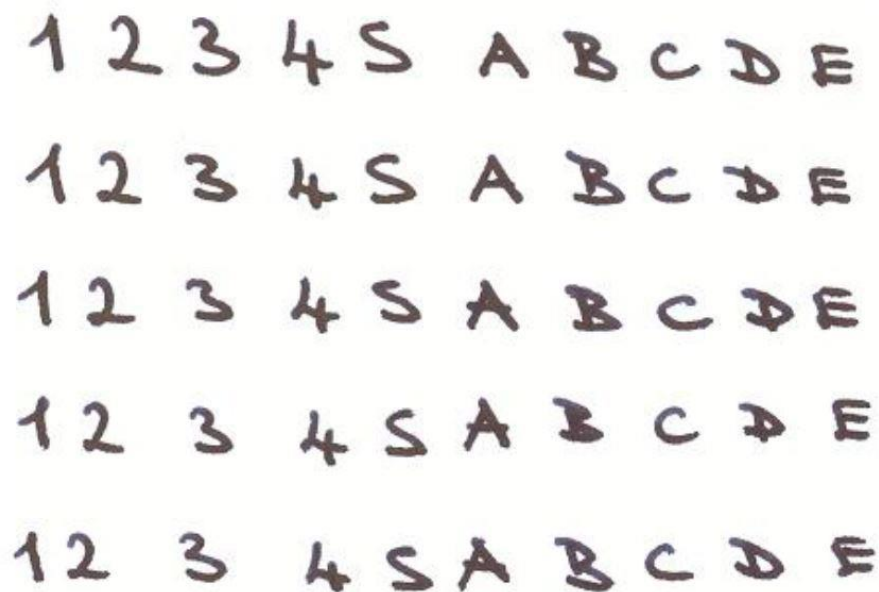
Simulation and Results

4.1 SIMULATION

4.1.1 Pre-processing of the image

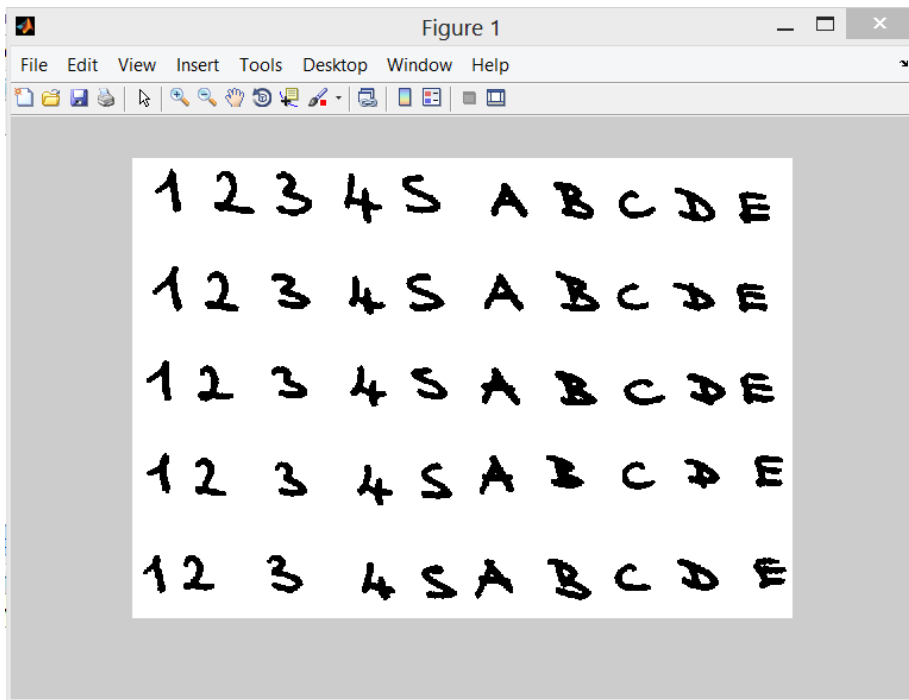
First of all the image on which the characters are written by hand is required.

Below is the example of one case in which an image sample is taken.



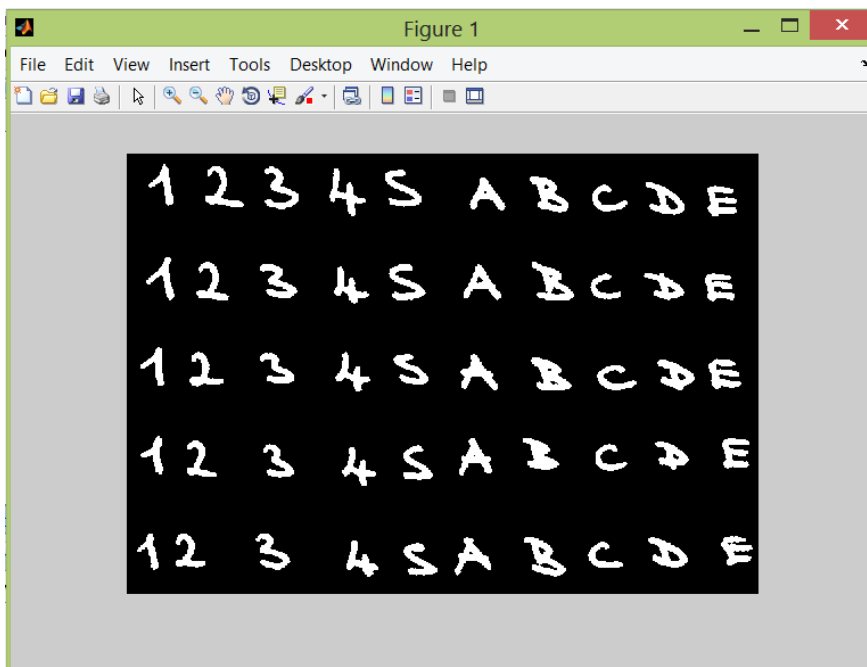
1. original handwritten image sample

Grey-scaling and Binarization of image



2. *Binarisation after gray scaling of image*

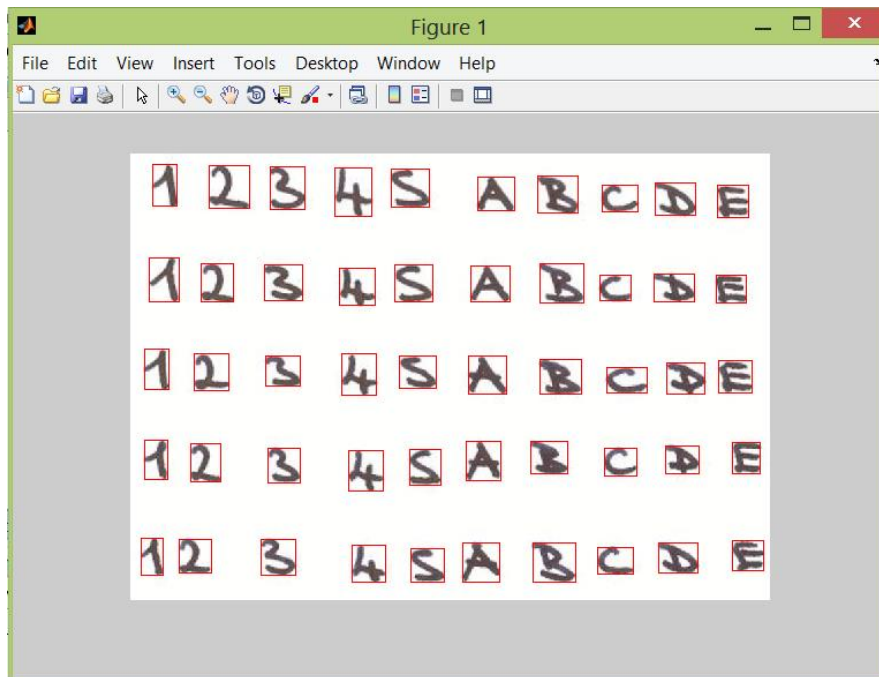
Inversion of binary image



Inverting binary image for blobs

4.1.2 Feature Extraction of characters from image

Indexing and Boxing of characters



3. Labelling and boxing of differentiated characters

Cropping, Reshaping and Resizing

Differentiation of characters is done by cropping the boxed characters of the pre-processed image. At first the sub-images are cropped label by label in the sample image, then the character image array is resized to form a 7X5 matrix pixelated image. This is done because an image array can only be defined with all images of fixed size.

Also the size of the character image should be maintained to a size of 7X5 because the ideal character set is defined as a set of images with 7X5 sized 2D matrix with binary values.

For this to be achieved first the images are reshaped to a 7 by 5 aspect ratio image then resized into a 7 by 5 size image



Acquired character from the sample image

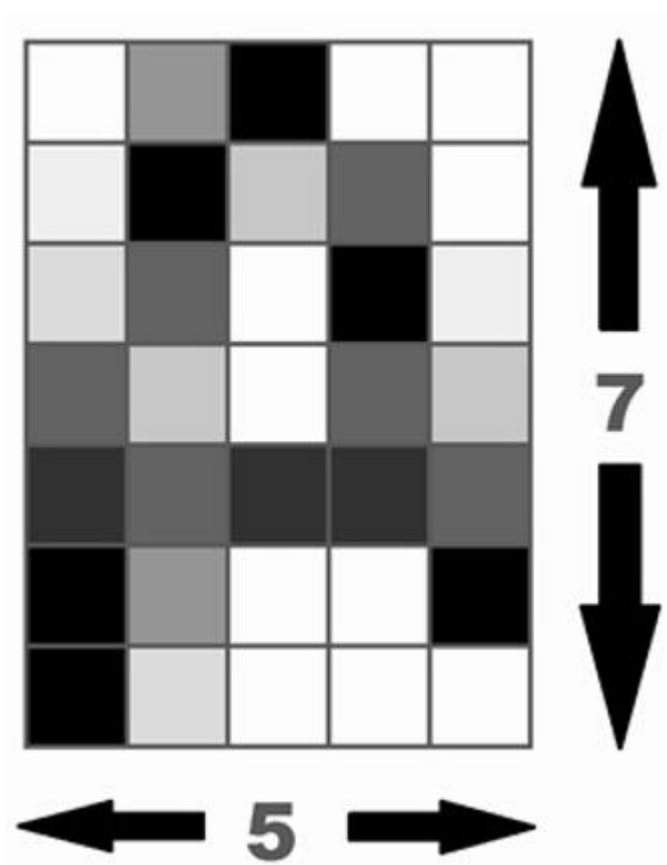


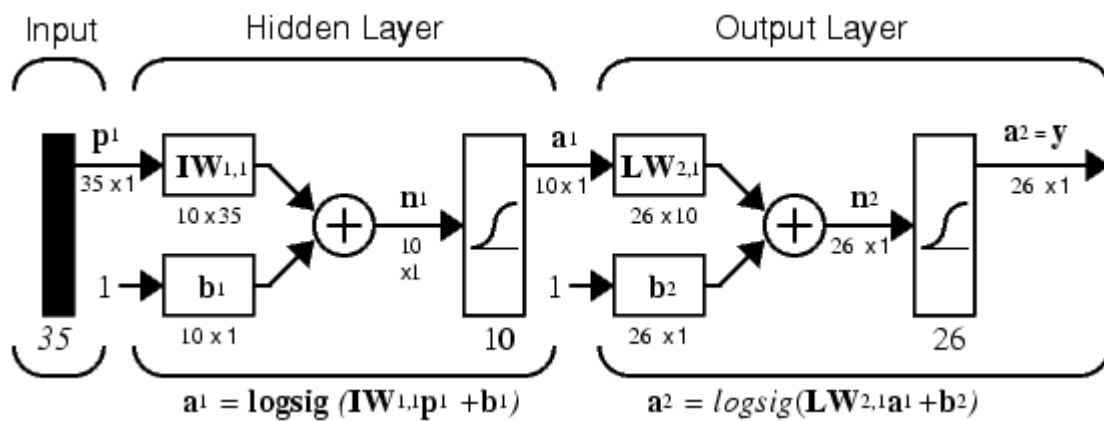
Image after resizing into a 7X5 sized image

4.1.3 Creating an Artificial Neural Network

The input is fed through the network which traverses through each neuron as it compares the input image with each neuron and gives the value in terms of a percentage of similarity between the input image and the neurons.

The neuron with having highest percentage of similarity to the input image is considered or estimated as the most favorable output which is most likely to that input.

In our case a network with 26 neurons and one hidden layer is enough.



Block diagram of an Artificial Neural Network

4.1.4 Training and testing the network

The screenshot shows the 'Neural Network Training (ntraintool)' window. At the top, the title bar reads 'Neural Network Training (ntraintool)'. Below the title bar, the 'Neural Network' section displays a diagram of a two-layer neural network. The input is processed by the first layer, which consists of a weight matrix W , a bias vector b , an addition node (+), and an activation function. The output of the first layer is then processed by the second layer, which also consists of a weight matrix W , a bias vector b , an addition node (+), and an activation function. The final output is shown on the right.

The 'Algorithms' section lists the following settings:

- Training: Levenberg-Marquardt (trainlm)
- Performance: Mean Squared Error (mse)
- Data Division: Random (dividerand)

The 'Progress' section shows the following metrics:

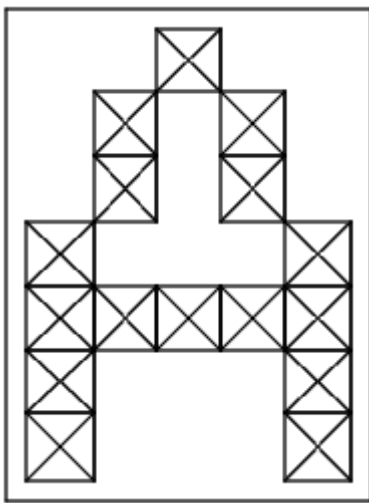
| | | | |
|--------------------|---------|--------------|----------|
| Epoch: | 0 | 2 iterations | 1000 |
| Time: | | 0:00:04 | |
| Performance: | 0.243 | 0.0480 | 0.00 |
| Gradient: | 1.00 | 0.0850 | 1.00e-10 |
| Mu: | 0.00100 | 1.00e-05 | 1.00e+10 |
| Validation Checks: | 0 | 0 | 6 |

The 'Plots' section includes buttons for 'Performance' (plotperform), 'Training State' (plottrainstate), and 'Regression' (plotregression). Below these buttons, the 'Plot Interval' is set to 1 epochs.

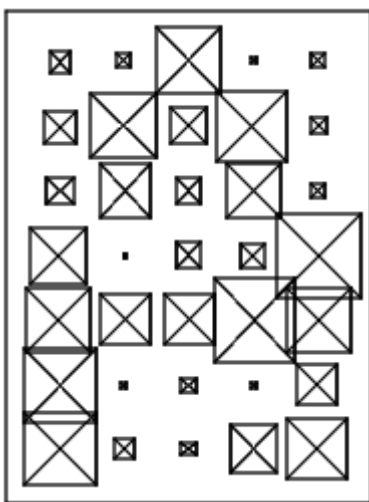
At the bottom of the window, a progress bar shows 'Training neural network...' with a green circular icon. Below the progress bar, there are buttons for 'Stop Training' and 'Cancel'.

The network tool training itself with ideal and noisy dataset

It is important to note that the network would not be immune to noisy hand written input if it is not trained properly. Or in other words, if the network is not trained with noisy set of characters along with the ideal set of characters, the network will not show the correct output every time. In fact all the handwritten characters remain irregular. So to make the network identify irregular shaped characters properly we must have to train the network with a noisy set of characters. In this case a set of noisy characters is obtained by adding some noise programmatically with some non-zero value of mean and variance.



A character from ideal set

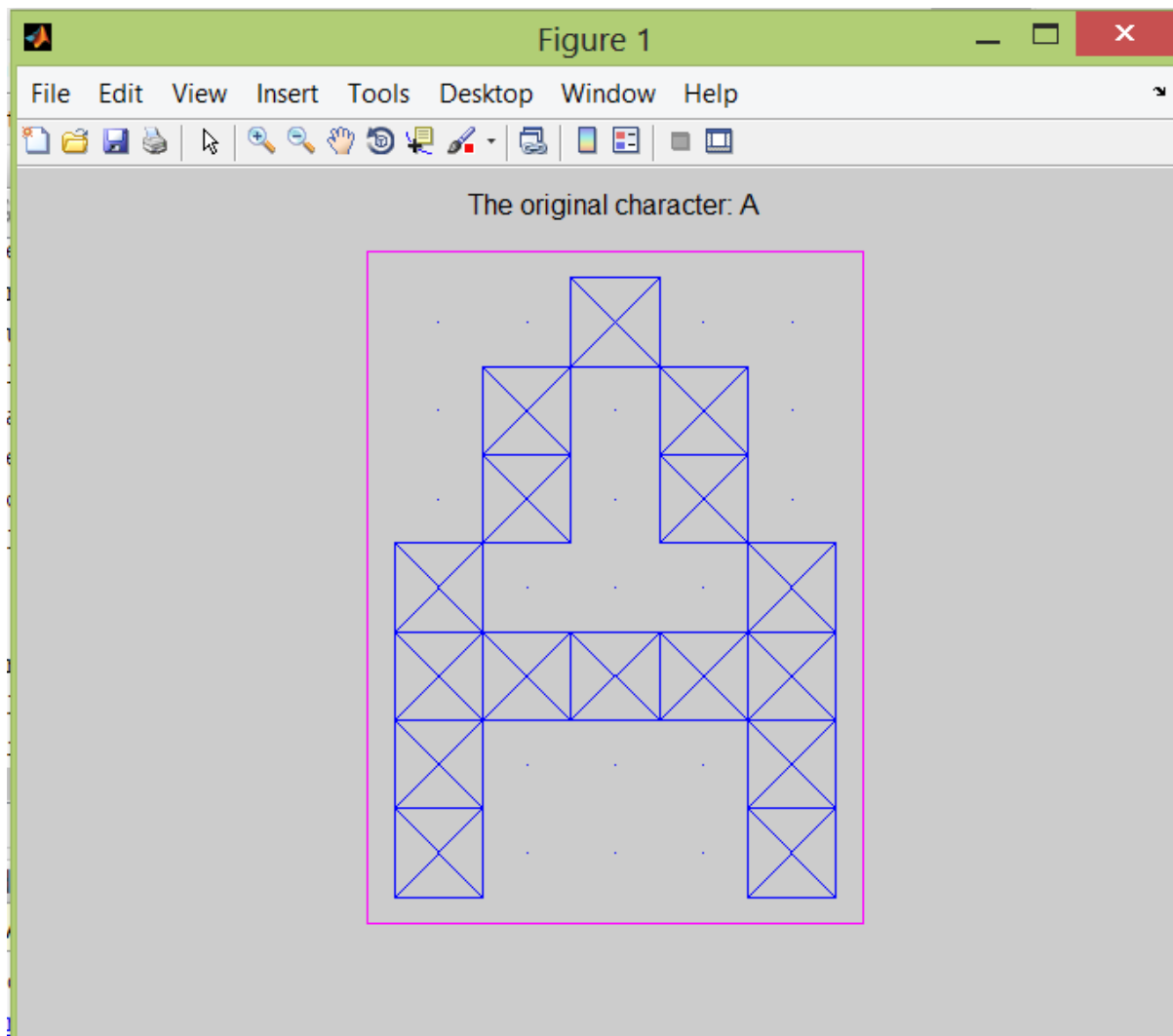


Programmatically adding some noise to the character

4.2 RESULTS

4.2.1 Identification of characters

After proper training and testing of the network, the pixelated 7 by 5 sized image of 'A' is fed to the network as input. Then the out we get is the resultant 2D matrix plot same as the character 'A' from the ideal dataset which was fed to the network as training dataset.



The character 'A' identified by network

CHAPTER 5

CONCLUSION

AND

FUTURE

IMPROVEMENTS

Conclusion and Future Improvements

5.1 CONCLUSION

Classification of characters and learning of image processing techniques is done in this project. Also the scheme through which project is achieved is **Artificial Neural Network** scheme. The result which was got was correct up to more than 90% of the cases, but it would be improved at the end. This work was basically focused on envisaging methods that can efficiently extract feature vectors from each individual character. The method I came up with gave efficient and effective result both for feature extraction as well as recognition. There are also different methods through which ‘handwritten character recognition’ is achieved.

5.2 FUTURE SCOPE OF THIS PROJECT

The application of this HCR algorithm is extensive. Now-a-days recent advancement in technologies has pushed the limits further for man to get rid of older equipment which posed inconvenience in using. In our case that equipment is a keyboard. There are many situations when using a keyboard is cumbersome like,

- We don't get fluency with keyboard as real word writing
- When any key on keyboard is damaged
- Keyboard have scripts on its keys in only one language
- We have to find each character on keyboard which takes time
- In touch-enabled portable devices it is difficult to add a keyboard with much ease

On the other hand if we use an HCR software in any device, we can get benefits like,

- Multiple language support
- No keyboard required
- Real world writing style support
- Convenient for touch enabled devices
- Previously hand written record can be documented easily
- Extensive features can also be added to the software like,
 1. Translation
 2. Voice reading

REFERENCES

1. G.Vamvakas, B.Gatos, N. Stamatopoulos, and S.J.Perantonis, “A *Complete Optical Character Recognition Methodology for Historical Documents*”, the Eighth IAPR Workshop on Document Analysis Systems
2. Debasish Basa ,Sukadev Meher , “*Handwritten Odia Character Recognition*”, Presented in the National Conference on Recent Advances in Microwave Tubes, Devices and Communication Systems, Jagannath Gupta Institute of Engineering and Technology, Jaipur, March 4-5,2011
3. Rajean Plamondon, Fellow IEEE and Sargur N. Srihari, Fellow IEEE , “*On-Line and Off-Line Handwriting character Recognition: A Comprehensive Survey*”, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. VOL. 22, NO. 1. JANUARY 2000
4. Sanghamitra Mohanty, Hemanta Kumar Behera , “A *Complete OCR Development System For Oriya Script*”
5. B.B. Chaudhuri U. Pal M. Mitra, “*Automatic Recognition of Printed Oriya Script*”
6. Fumitaka KIMURA, Tetsushi WAKABAYASHI and Yasuji MIYAKE, “*On Feature Extraction for Limited Class Problem*”
7. Supriya Deshmukh, Leena Ragha, “*Analysis of Directional Features - Stroke and Contour for Handwritten Character Recognition*”, 2009 IEEE International Advance Computing Conference (IACC 2009) Patiala, India, 6-7 March 2009
8. M. Blumenstein, B. K. Verma and H. Basli, “A *Novel Feature Extraction Technique for the Recognition of Segmented Handwritten*

Characters”, 7th International Conference on Document Analysis and Recognition (ICDAR '03) Eddinburgh, Scotland: pp.137-141, 2003.

9. Tapan K Bhowmik Swapan K Parui Ujjwal Bhattacharya Bikash Shaw, “ *An HMM Based Recognition Scheme for Handwritten Oriya Numerals*”.
10. Sung-Bae Cho, “*Neural-Network Classifiers for Recognizing Totally Unconstrained Handwritten Numerals*”, IEEE Transactions on Neural Networks, Vol. 8, No. 1, January 1997
11. N. Tripathy and U. Pal, “*Handwriting Segmentation of Unconstrained Oriya Text*”