Bachelor's thesis

Information and Communications Technology

2018

Pramoj Prakash Shrestha

OPTICAL CHARACTER RECOGNITION

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information and Communications Technology

2018 | 50

Pramoj Prakash Shrestha

OPTICAL CHARACTER RECOGNITION

Optical Character Recognition (OCR) is the process of extracting the characters from a digital image. The concept behind OCR is to acquire a document in image or pdf formats and extract the characters from that image and present it to the user in an editable format.

The author of this thesis tested an Artificial Neural Network (ANN), which is a mathematical representation of the functionality of the human brain, using Back-propagation Algorithm with test case files of English alphabets. The purpose of this thesis was to test systems capable of recognizing English alphabets with different fonts, and to be familiar with ANN and digital image processing and apply it for character recognition.

Scientific journals and reports were used to research the relevant information required for the thesis project. The chosen software was then trained and tested with both computer and hand-written alphabets in image files. The tests revealed that the OCR software is able to recognize both computer and hand-written alphabets, and learns to do it better with each iteration.

The study shows that although the system needs more training for hand-written characters than computerized fonts, the use of ANN in OCR is of great benefit and allows for quicker and better character recognition.

KEYWORDS:

Image processing, character recognition, neural network, back-propagation algorithm

CONTENTS

FIGURES

LIST OF ABBREVIATIONS (OR) SYMBOLS

Abbreviation	Explanation of abbreviation (Source)
OCR	Optical Character Recognition
ASCII	American Standard Code for Information Interchange
ANN	Artificial Neural Network
PE	Processing Element
UI	User Interface

1 INTRODUCTION

Optical Character Recognition (OCR) is the process of converting the scanned documents, pdf into computer readable and editable format, i.e., ASCII Characters. OCR is the process of classification of optical patterns contained in a digital image corresponding to alphanumeric or other characters [1]. In spite of the title being the Optical Character Recognition, the work of digital image processing also has to be done. So it is essential to have the background knowledge for both image processing as well as for character recognition.

The word “recognition” plays an important role in our lives. It is a basic property of all human beings; when a person sees an object, he or she first gathers all information about the object and compares its properties and behaviors with the existing knowledge stored in the mind [1]. If we find a proper match, we recognize it.

If we see an object, it has features that describe the object, which is a set of characteristics of that object. These sets, when combined with each other, will create a fixed shape and size of that object, called the Pattern of the object [1]. If there are many objects which share the same pattern then these patterns fall in a class known as the pattern class. So pattern recognition is the process where characters of objects are assigned to pattern class.

Given a pattern, its recognition/ classification may consist of one of the following two tasks: 1) supervised classification (e.g. discriminant analysis) in which the input pattern is identified as a member of predefined class, 2) unsupervised classification (e.g. clustering) in which the pattern is assigned to hitherto unknown class [2]. Pattern recognition can be used to develop applications for fingerprint identification, voice recognition, face recognition, character recognition, and signature recognition. It can also be used in the field of scientific research such as astronomy, engineering, statistics, medical, machine learning and neural networks. The design of pattern recognition system essentially involves the following three aspects: 1) data acquisition and preprocessing 2) data representation and 3) decision making [2].

If we compare the methods of pattern recognition between the human and the machine, the brain of human acts as a memory space where the patterns for different objects are stored. So when a human come across an object, the characters of the object is cross-referenced with the stored patterns and the object is identified. The human brain is made of a large number of neurons that are connected to each other known as synapses. A single piece of information is stored by a number of neurons rather than a single neuron each contributing a small portion of it. So, when a portion of the brain is removed which removes the neurons, the whole of the information is not removed. In case of a machine, the patterns are basically a collection of pixels. They have to be trained using certain patterns in order for the system to be capable of recognizing any patterns.

We can use the machine to model the human pattern recognition process which involves the use of fuzzy logic, neural network and the genetic algorithm.

Fuzzy logic is a logic that is designed to handle the concept of partial truth, i.e., the truth that in between “completely true “and “completely false”. In binary logic is either 0 or 1 but fuzzy logic is a continuous value between 0 and 1. It includes a set of membership functions for each input and a set for each output and a set of rules, which is applied to the membership functions to yield an output value. These logics are well suited to nonlinear systems and systems having multiple inputs and multiple outputs and also to that system which cannot be modeled using conventional means [3].

Neural Network is a collection of nodes which represent the neurons in the human brain, interconnected with each other. In this network, the information is stored in nodes as the weight of the links similar to the information stored in neurons. When there is more information to be stored, the adjustment is made between the links of nodes. The nodes work cooperatively with each other and achieve massive parallel distributed processing. The neural network has adaptive-learning, self-organization, and fault-tolerance.

A genetic algorithm is based on the logic of Darwinian selection, survival of the fittest, only the best are selected for replication. Similar to the biological evolution where genes are transferred from parent to child, during this transformation, there are some changes in the characteristics, replication of the nodes brings changes in a data structure which helps the learning.

The objectives of the thesis are as follows:

- To understand the process and steps involved in the development of OCR.
- To test a system capable of recognizing English alphabets.
- To have better understanding of ANNs and apply it for character recognition.
- To have better understanding of different digital image processing tools.

The study will put emphasis on the testing of the OCR software using computer printed and handwritten English alphabets, as the system is capable of learning and recognizing a single character at a time. The duration of training the system will, therefore, be long because the handwritten characters have more complex factors to be considered such as alignment and different writing styles.

2 DIGITAL IMAGE PROCESSING

Extracting the text from the document image and processing it further for making it capable to be recognized is the task of digital image processing. Reliable character segmentation, removable of various noises, skeletonizing the character and normalizing it all comes under the process of digital image processing. Prior to character recognition, it is necessary to isolate the individual characters from text image. Digital image processing refers processing of two-dimensional picture. Digital Images are the pictures that have been converted into a computer readable binary format consisting of logical 0’s and 1’s [4].

Digital image processing has a broad spectrum of application in various fields such as the remote sensing via satellites and other spacecraft’s, image transmission and storage for a business application, medical processing, radar, sonar, and robotics and automated inspection of industrial parts [3].

The digital image processing operation can be broadly grouped into five fundamental classes, which are as follows:

2.1 Image representation and modeling

Image representation and models deal with the process of finding proper ways to mathematically represent the image. Basically, there are two ways to represent an image. The external representation deals with the boundary of the image and used when the primary focus is on the shape characteristics. The internal representation deals with pixels within the region and used when the primary focus is regional properties such as color and texture. But sometimes it is necessary to use both representation techniques.

2.2 Image Enhancement

Image enhancement refers to the process of making a change to the features of the image such as edges, boundaries, or contrast to make a graphic display more useful for display and analysis [3]. The enhancement process does not increase the inherent information content of data but it does increase the dynamic range of the chosen features so that they can be detected easily. Image enhancement includes gray level and contrast manipulation, noise reduction, edge crispening and sharpening, filtering, interpolation and magnification, pseudocoloring and so on.

2.2.1 Noise removal/reduction

Noise is a random variation of image Intensity and visible as grains in the image. It may cause to arise in the image as effects of basic physics-like photon nature of light or thermal energy of heat inside the image sensors. It may produce at the time of capturing or image transmission [5]. In an image, noise refers to the pixels showing different intensity values instead of true pixel values. It is introduced in the image at the time of image acquisition or transmission.

In a binary image, we refer noise as a black pixel that should be white or vice versa. The noise should be removed in early stages, if not it is propagated to the further steps of processing, then it may affect the operation of the system. There are basically two different types of noise. Salt noise, when the pixel is supposed to be black but it is represented by a white pixel. Pepper noise, when the pixel is supposed to be white but is represented by the black pixel. We can use various types of filters to remove noise from the images such as mean filter, median filter, Laplacian filter, mode filter etc.

a) Median Filter

The median filter is a simple and very effective noise removal filter. It is a non-linear filter and belongs to the class of edge-preserving smoothing filters. When performing median filtering pixel is determined by the median value of all pixels in a selected neighborhood (mask, template, and window) [6]. In the median filter, an odd sized window such as 3 X 3, 5 X 5, 7 X 7 etc. is taken and the pixel values within the window are arranged in ascending or descending order. The median value is taken from the set of values and replace to the working pixel.

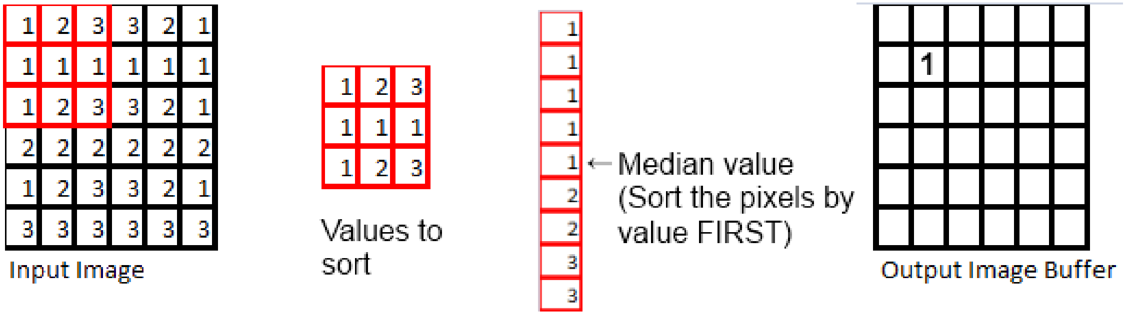

Figure 1. Median Filter ([7]).

b) Mean Filter

Mean filter, a simplest low pass linear filter, is simple and easy to use for smoothing an image. In this filtering, the sum of all the neighboring pixels in the window is divided by the total number of pixels. The value we obtain after the calculation then replaces the value of the pixel being processed. One of the main problems of this filtering is the blurring of the edge. If a sharp edge is required, then this filter is not the best choice.

2.3 Image Restoration

When an image is acquired using optical, electro-optical or electronic means, there is the possibility of degradation due to sensing environment. The different types of degradation that might occur are as geometrical degradation, illumination and color imperfection, and blur.

Blurring is a form of bandwidth reduction of an ideal image owing to the imperfect image formation process. It can be caused by relative motion between the camera and the original scene, or by an optical system that is out of focus [4].

The field of image restoration (sometimes referred to as image deblurring or image deconvolution) is concerned with the reconstruction or estimation of the uncorrupted image from a blurred and noisy one [4]. The overall effectiveness of the restoration filters depends on knowledge of degradation process and the filter design. Image restoration is not the same as image enhancement. Image enhancement criteria are difficult to represent mathematically whereas the restoration problems can be quantified precisely. Image restoration takes the recorded image and produces the best estimates of the original image.

2.4 Image Analysis

Image analysis is the process of retrieving meaningful information from images, extract statistical data. The feature which is extracted during image analysis can be related to finding shapes, detecting edges, removing noise, counting objects, measuring region and image properties of an object. For this reason, image analysis includes processes, such as segmentation, skeletonization, slant correction, and size normalization.

2.4.1 Skeletonization

Skeletonization is the result of the thinning process, which peels the color of the text until it reaches the most medial one-pixel width [8]. During this process, the pixels of an object are removed without affecting the general shape of the pattern. The output object after skeletonization should be one pixel thick, through the middle of the object and the topology of the object should be preserved.

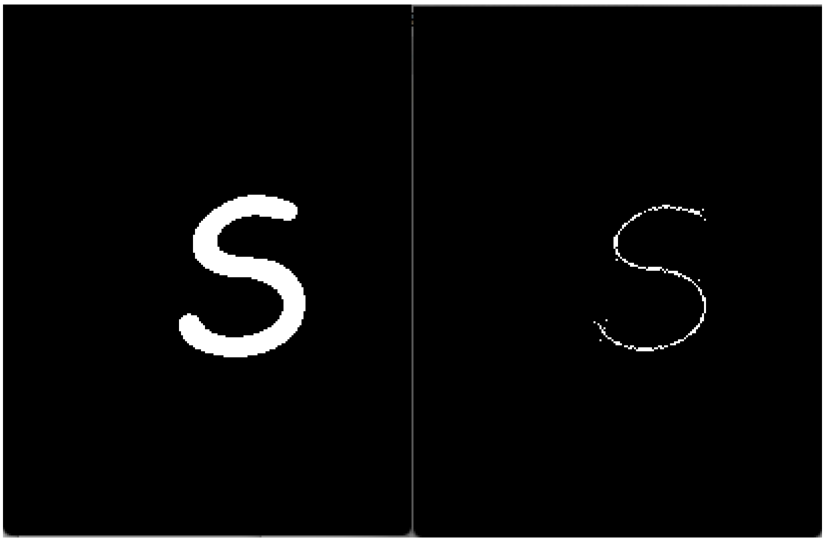

Figure 2. Skeletonization ([9]).

2.4.2 Segmentation

Segmentation is a mid-level processing technique used to analyze the image and can be defined as a processing technique used to classify or cluster an image into several disjoint parts by grouping the pixels to form a region of homogeneity based of the pixel characteristics like gray level, color, texture, intensity and other features[10]. It is the process which finds the boundary and the object in an image. Using the segmentation process we can separate and isolate the object of interest from the background of an image. The different approaches to segment an image are listed below.

a) Intensity-based methods

An intensity-based method is a threshold-based technique where an image is classified into two classes. The pixels of an image are classified according to the intensity value, i.e., those pixels whose intensity value lies in a certain range falls in one class and the rest belongs to the next class.

b) Discontinuity-based methods

In a discontinuity-based method, the abrupt change in the intensity of pixels is used to determine the object. If there are significant abrupt changes in intensity among neighboring pixels in a certain direction, it is considered to be the edges. The possible edges are grouped together to form the boundary of the object.

c) Similarity-based methods

In a similarity-based method, the area of the neighboring pixel is compared with each other for the similarity in the characteristics. According to this method, the pixels inside a region possess similar characteristics and are dissimilar to the pixels in other regions. So a similarity test is done between every neighboring pixel if the characteristics are similar then it is added to a group and the region grows. The growing is stopped when the similarity test fails.

d) Active contour model

An Active contour model can be used to analyze the shape of an object. To obtain an accurate result using this method, one needs the prior knowledge of image intensity distribution, boundary shape, and texture. The main objective is to apply a segmentation process to an image by doing deformation to the initial contour toward the boundary of the object of interest. This is obtained by deforming an initial contour to minimize. An active contour can be categorized as edge-based models or region based models.

2.4.3 Size Normalization

Normalization is the process of transferring the characters into fixed standard size. After the image is segmented into characters, it is important to adjust the size of the character and bring it to a standard size so that the following stages can assume a standard character size and it becomes easy while processing. Size normalization is the process of resizing the image to a standard size. The resizing of characters is needed in order to make the recognition operation process independent of the writing size and the resolution of the scanned. Size normalization mainly involves two processes.

Upscaling is the process of increasing the size of the image. If the input image is smaller than the standard size the scaling up is required. The upscaled image includes duplicates of each selected rows. Duplication may depend on the number of times the image has to be scaled.

Down-scaling is the process of decreasing the size of the image. It is mainly useful if the input image is greater than the standard size. The downscaled image is created by removing selected rows. Numbers of rows to be removed also depend on the times the image has to be scaled down.

2.5 Image Data Compression

The main objective of image data compression is to represent an image signal with the smallest possible number of bits without loss of any information. Compression of data helps in speeding the transmission and minimize the requirements of storage. A visual information occupies a large storage capacity. Although the capacities of several storage media are substantial, their access speeds are usually inversely proportional to their capacity. So the compression techniques focus on reducing the number of bits required to store or transmit images without any loss of information.

The image compression help to decrease the redundancy in image data and also transmit few numbers of samples.

3 ARTIFICIAL NEURAL NETWORKS

Artificial Neural Network (ANNs) are computer programs inspired by the functional processing of information by the human brain. ANNs consist of numbers of nonlinear computational elements called neurons. ANNs gather their knowledge by detecting the patterns and relationships in data and learning through experience, not from programming. The artificial neurons also are known as processing elements (PE), are connected to each other and the connection between two PE's have a coefficients, i.e., weight, which is adjustable. Each PE has weighted inputs, transfer function, and one output. The behavior of a neural network depending on the transfer functions, learning rule and architecture of the neurons. The neuron is activated using the weighted sum of the inputs and the activation signal is passed through a transfer function to produce a single output of the neuron. During the training, the weights are adjusted until the error in predictions is minimized and the network reaches the specified level of accuracy. Once the network is trained and tested it can be given new input information to predict the output.

3.1 Components of ANN

Components, which make up ANN, are discussed here. These components are valid whether the neuron is used for input, output or is in the hidden layer. Figure 3 shows a basic representation of an artificial neuron.

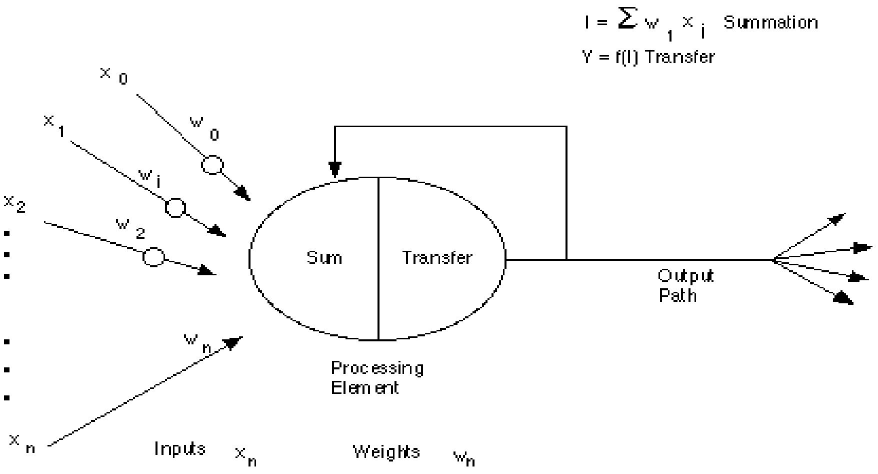

Figure 3. A Basic Artificial Neuron ([11]).

In Figure 3, various inputs to the network are represented by the symbol, x(n). Each of these inputs is multiplied by a connection weight represented by w(n). In a simple case, these products are summed and fed to a transfer function (activation function) to generate a result, and this result is sent as output. There is also the possibility of utilizing the binary properties of ORing or ANDing of inputs along with the summing operation. Such functions can be built into the summation and transfer functions of the network.

Weight Factor:

A neuron usually receives many simultaneous inputs and has its own weight. Weights are adaptive coefficients that determine the intensity of the input signal and are a measure of an input's connection strength.

Summation Function:

Summation function adds up the product of neuron and their corresponding weight of the link e.g. $\sum x_n w_n$. The summation function can be more complex than the simple product. There can be a selection of minimum, maximum, majority, product or several normalizing algorithms depending on the network architecture and paradigm.

Transfer Function:

The final result of the summation function is transferred to the output through an algorithmic process known as a transfer function. This function can be divided into two categories.

a) Threshold

The output of the summation function is compared to some threshold value to determine the neural output. If the sum is greater than the given threshold value, the processing element generates a signal and if less than the threshold value, no signal is generated, or vice versa. In the case of the threshold function, any values above or equal to a given threshold are converted to 1, while anything falling below it is converted to a 0 during activation [12].

b) Sigmoid

The output varies continuously but not linearly as the input changes.

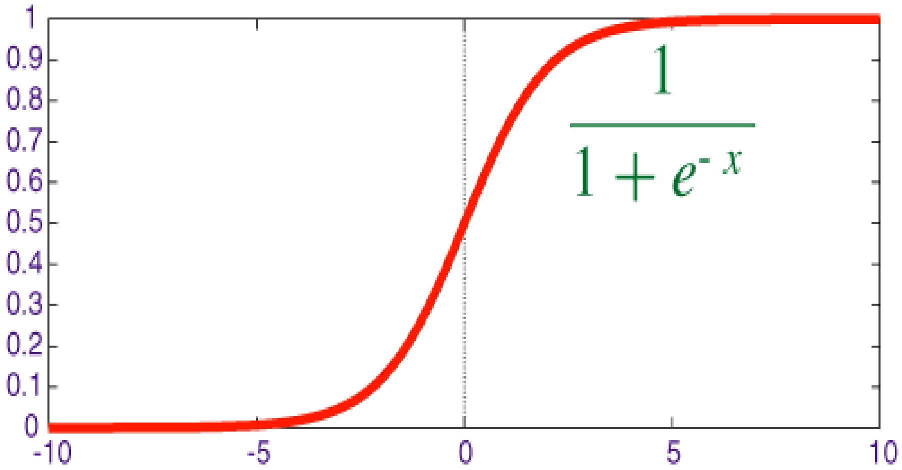

Figure 4. Sigmoid ([13]).

3.2 Topology of ANN

The topology of ANN is defined on the basis of the arrangement of processing units, connections, and pattern of input/output.

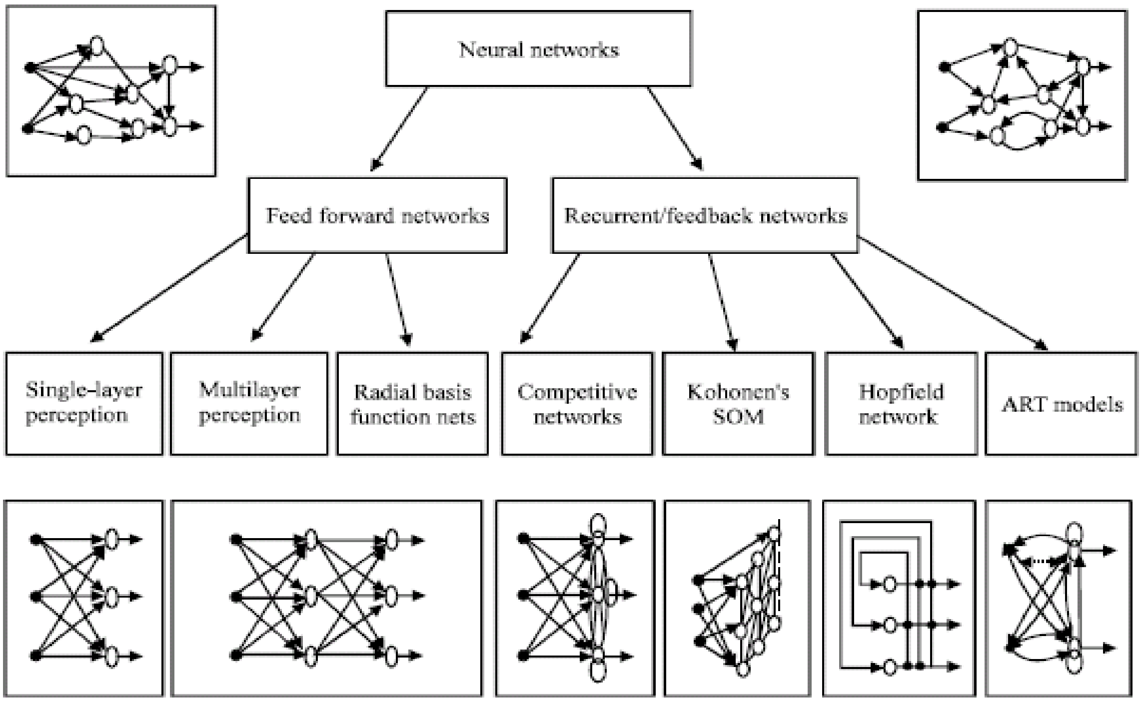

Figure 5. Types of ANN's ([14]).

3.2.1. Single Layered Feed-forward networks

In Single layered Feed-forward networks, all the input notes are directly connected to the output node. The signals are transferred from the input to the output but nor vice-versa. This type of network can only recognize linearly separable functions.

3.2.2 Multilayered Feed-forward networks

In Multilayered Feed Forward networks, there are one or more hidden layers in between the input and output layers. Here the input is not connected directly to the output but instead, it is connected to the output layer via the hidden layer.

3.2.3 Recurrent networks

The Recurrent network differs from the feed-forward network in the sense that there is at least one feedback loop. This type of network has one or more hidden layers with at least one feedback loop. The feedback may be a self-feedback and sometimes use unit delay elements.

3.3 Training of ANN

Once the neural network has been structured, it is ready for training. Training is the process of adjusting the weights of the connection. At the start of training, random weights are assigned to the connections. There are basically two approaches to training, which are discussed below.

3.3.1 Supervised Training

In supervised training, both input and output are provided. So the network has an input value, weight, and desired output value. Here the network processes the inputs and generates an outputs, these outputs are then compared with the desired outputs provided at the start of training. The errors acquired after the comparison is then transferred or propagated back through the system, causing the system to adjust weights, which control the network. This process is repeated continuously until the desired outputs are matched.

3.3.2 Unsupervised, or Adaptive Training

In unsupervised training, the network is provided with the input data but not the output data. The system itself must decide what features it will use to group the input data. The system is provided training set, the objective is to categorize or discover features or regularities in the training data. The network internally monitors their performance and adjust the weights accordingly. During the construction of the network, information about how to organize itself if provided.

3.4 Learning Rates

The purpose of learning rate is to modify the connection weights on the inputs of each processing element. This process of changing the weights of the input connection to achieve some desired result can be also be called adaptation function as well as the learning mode. The higher the learning rate the convergence will be higher, resulting in a higher error in the performance.

4 METHODOLOGY

Before starting with the design and the construction of OCR, the author went through different books, journals and articles related to Artificial Intelligence, Neural Network and digital image processing. After the preliminary study, it was found that back-propagation algorithm was one the best algorithm to be used for the system to learn and recognize the characters. So the search was further focused on finding the in-depth information on back-propagation algorithm.

Similarly a preliminary study of the different algorithms used in digital image processing was done and selection of algorithms were based on the findings from the studies. Median filter was select for removing noise from the image as it preserves the edges during the noise removal. Hilditch's algorithm was one of the best to be used for the process of skeletonization as it was easy to use and preserved the characteristics of a character, and can be used on different patterns effectively.

For the testing of the algorithms, the author researched different software based on the keywords, algorithms and the licensing. The software selected were:

Image Processing Lab in C# was selected as it included different tools required for the thesis, for example; median filter, Hilditch's algorithm, binarization and normalization.

Unicode Optical Character Recognition was selected as it allowed for easy demonstration of the process of segmentation.

Neural Network OCR was able to handle both computer printed and handwritten characters. Also, it uses back-propagation algorithm for the training and recognition processes.

5 DESIGN AND CONSTRUCTION

5.1 Design Phase

Design phase deals with the process of laying out the flow of information for a system. The design mainly consists of how the inputs are provided to the system and how the outputs are displayed. It also consists of the design of the process i.e. how the processing modules take place, how the data moves through the module. The flow of data can be shown with the help of the data flow diagram (DFD).

A DFD maps out the flow of information using different symbols like rectangles, circles, arrows, short text labels etc. It can be simple as well as multi-level that digs progressively deeper into how the data is handled.

5.1.1 General Design

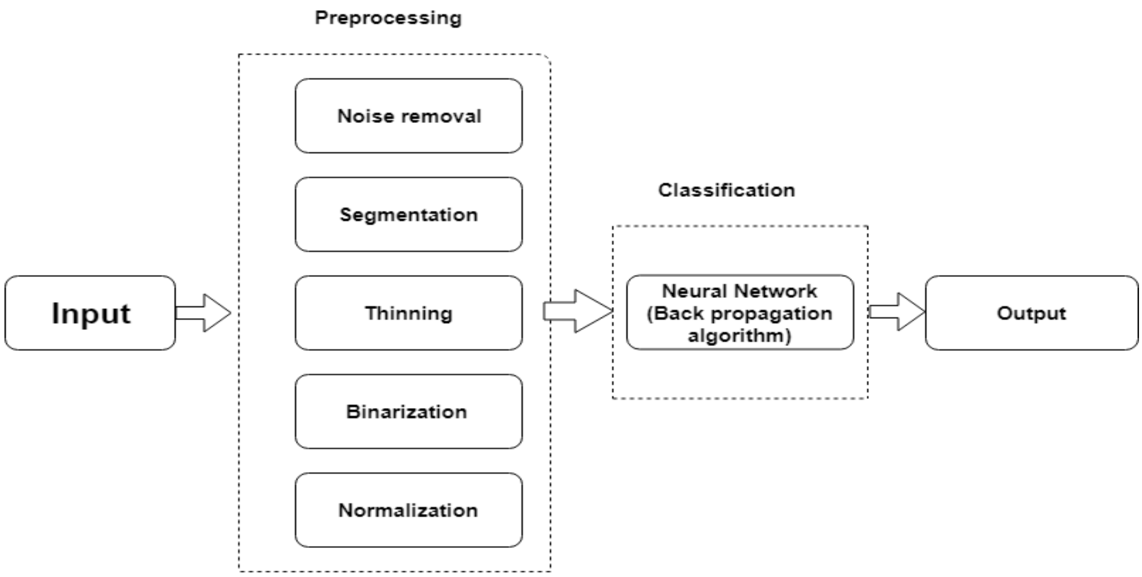

Figure 6. General Design of the System.

Figure 6, shows the general design for optical character recognition. It represents the steps involved in the construction of the system and the flow of data. The system is divided into two different main stages. The first one is preprocessing, it includes the algorithms used for digital image processing. Once, the input is supplied by the user to the system. The preprocessing stage extracts the characteristics of the character to be used in the next stage called, classification. For the classification, the neural network has been used. There are many algorithms that can be used for the classification. For this system, back-propagation algorithm has been used. In this stage, the system is trained before it can recognize the characters. Once all the characters have been trained, the system can recognize the characters and an output can be generated for the user.

5.1.2 Process Design

As already stated the process design is done with the help of DFD's.

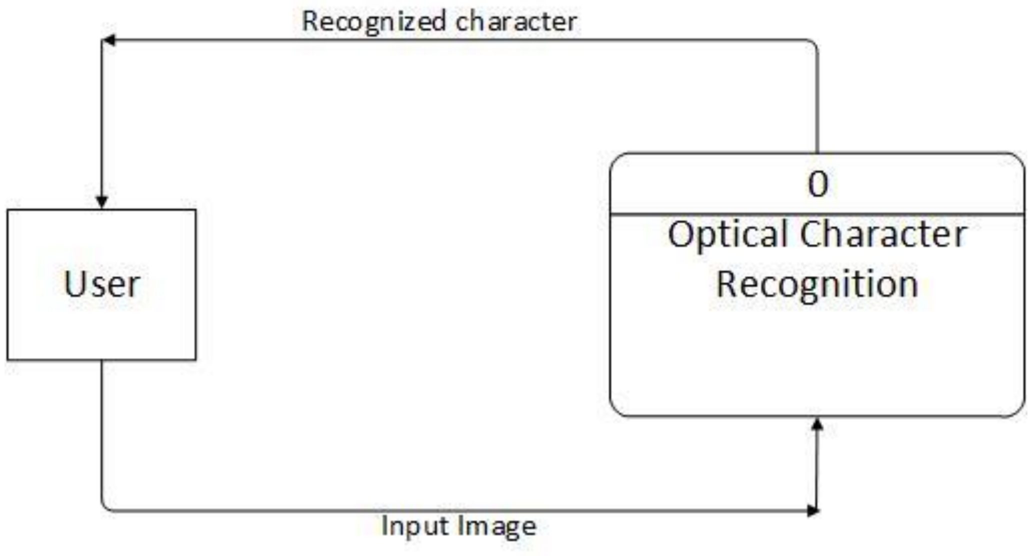

Figure 7. Context Diagram for OCR.

Figure 7, shows the context diagram, which defines the external entities that will interact with the system. For the OCR, the external entity is the user. The user provides the input for the system and waits for the output. This context diagram can be further broken down into smaller diagrams to show the processes involved in details.

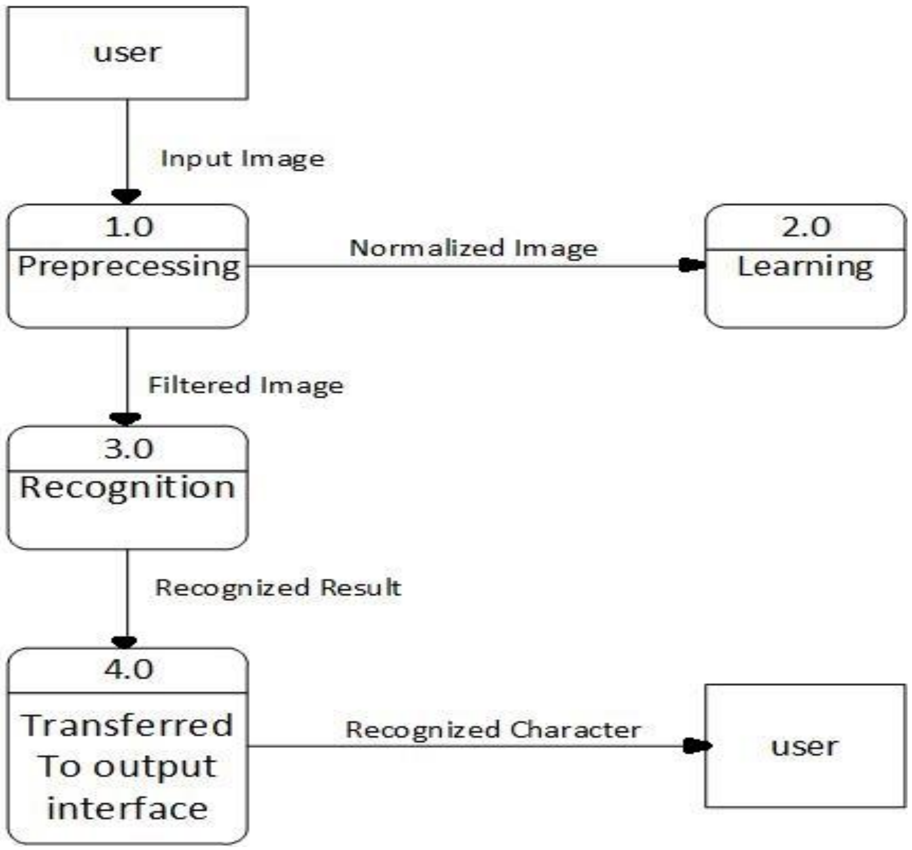

Figure 8. Level 0 DFD for OCR.

Figure 8 shows the Level 0 data flow diagram, which is the breakdown of the context flowchart. OCR consist of four main processes each numbered according to its occurrence and named according to its function. The process numbers are given as 1.0, 2.0 etc. which means the process is the first process of 0 level DFD or the second process of the 0 level DFD.

The input data given by the user first enters the process named preprocessing. The preprocessing consist of different digital image processing tools that will convert the user image to our required criteria for the further processing. During this process, the image goes through filtering of noise, segmentation, thinning, binarization and normalization. The normalized image is either transferred to the learning process or the recognition process according to the requirement. The learning process generates the computed weights as output which is stored in weights database. Similarly, the recognition process requires the weights from the weights database as input. The output from the recognition process is the recognized character that is transferred to the output interface and given to the user.

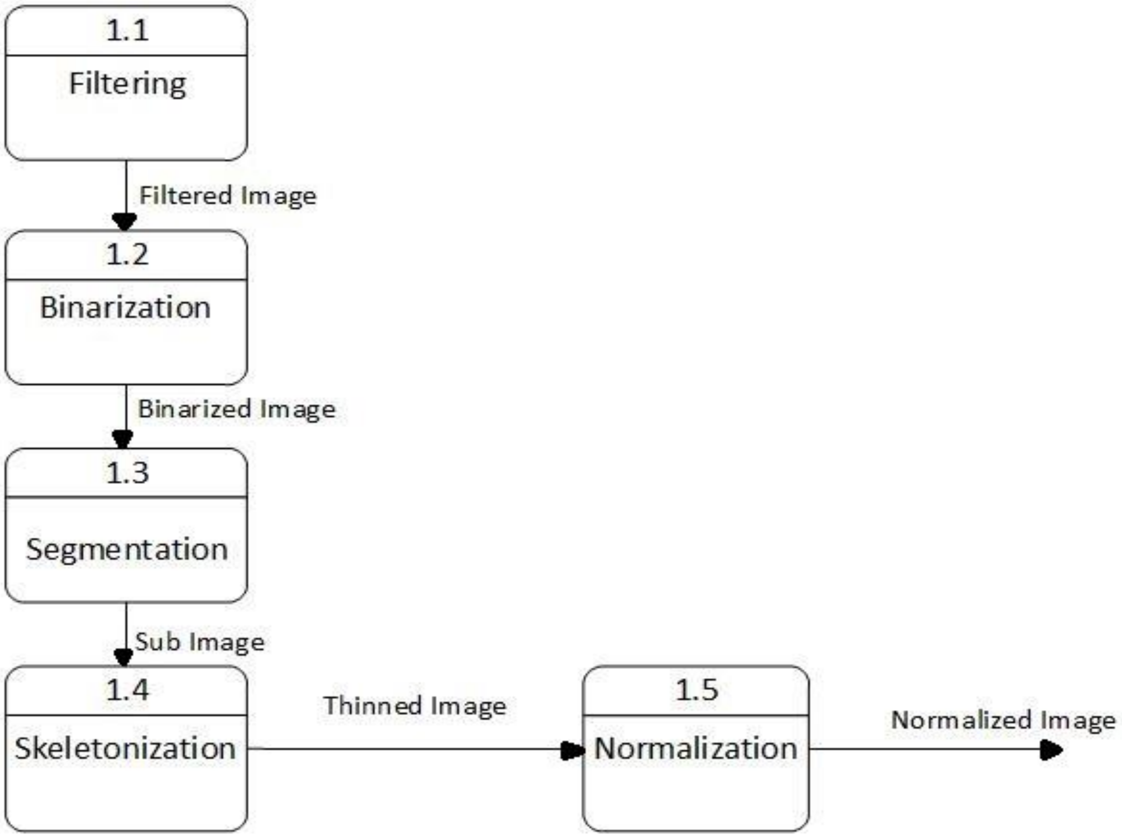

Figure 9. Level 1 DFD for OCR.

Figure 9 is the Level 1 DFD, which is the decomposition of the level 0 DFD. The decomposition is done process wise so, each process has a separate level 1 DFD given that the process can be further decomposed.

The preprocessing process is decomposed into five different processes each numbered according to its occurrence. The process may consist of filtering, binarization, segmentation, thinning and normalization. The number consists of 1.1, 1.2 etc. which means the first or the second process of the decomposed process number 1.

The image acquired from the user is first filtered so as the unwanted errors or noise in the image can be removed. Then the image goes through the process of binarization, where the given image is converted to a binary image. The binarized image is then fed to segmentation process where the individual characters from the image are separated. The single separated character known as the sub-image is then fed to normalization process whose output is the normalized image of standard size.

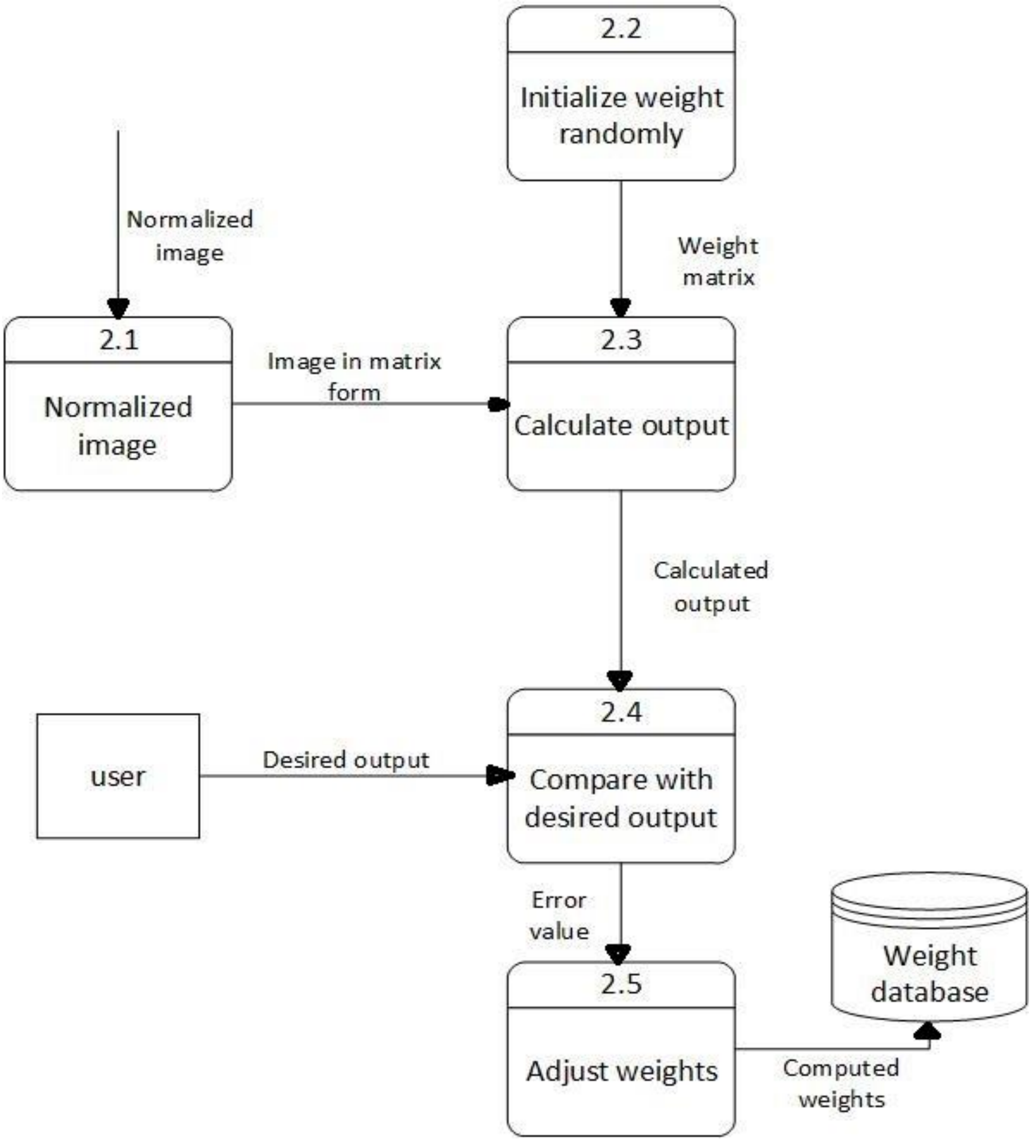

Figure 10. Level 1 DFD for Learning.

Fig 10. is the level 1 data flow diagram of the learning process. First, the output from the normalized image is taken as input. This input along with the output from the ”initialize weights randomly” process, which is the weights matrix, is fed to the next process, which is the ”calculation of the output”. The outputted from this process is the actual output calculated. This output is then compared with the desired output, which is given by the user in the processed named” compare with desired output”. This process gives the error between the desired and the actual output, which is utilized to adjust the weights in the”adjust weights using error value” process. After the weights are adjusted to get the minimum error value, the adjusted weights known as computed weights are then stored in the database known as the weight database.

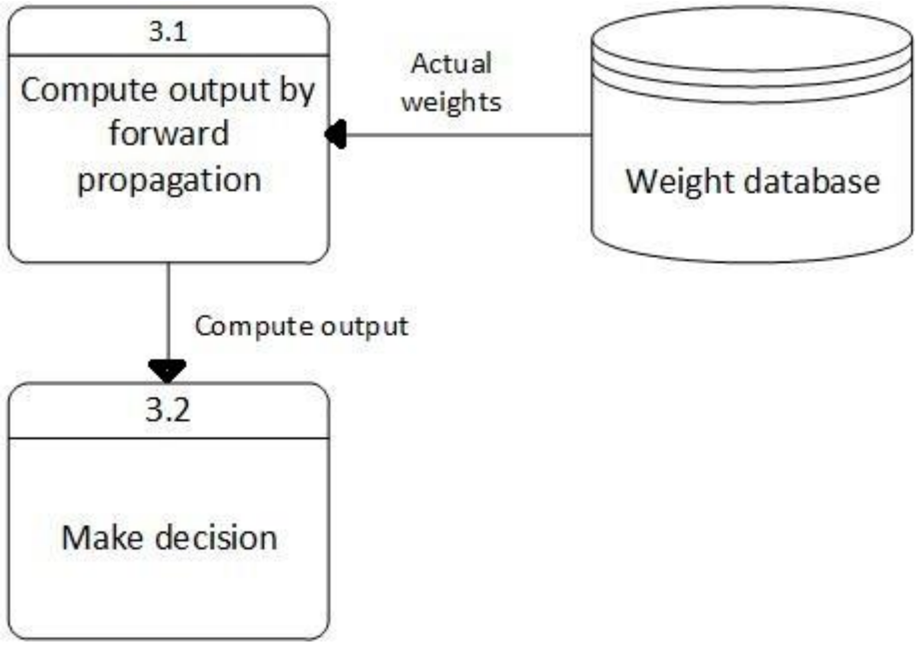

Figure 11. Level 1 DFD for Recognition.

Figure 11 is the level 1 data flow diagram for the recognition process. The further decomposition of the recognition process is shown in fig as the level 1 DFD. It consists of two processes 3.1 and 3.2. The first process is the -"compute output by forward propagation" which gets its input as the normalized image and weights from the weight database. The calculated output is then fed to the -"make decision" process, which analyzes the calculated output and gives the recognized character to the user.

5.2 Construction Phase

After the design phase comes to the construction phase where the algorithms are implemented to develop the system. The system can be divided into two parts, first one is the preprocessing part that consists algorithms for image processing and the second one is the classification part where the backpropagation algorithm from the neural network will be used. The different parts of the system are discussed in detail below.

5.2.1 Preprocessing

Preprocessing involves the image processing algorithms used for the system. There are multiple ways of obtaining the same result. Algorithms that are used in this system are discussed below according to their occurrence.

5.2.1.1 Noise reduction

The input image may contain various types of noise while scanning which may be undesirable while processing. Here noise may be the unwanted pixel of the same intensity as that of the image i.e. a black pixel where ever white is needed and vice versa. So, this type of noise should be removed before further processing. Among the different noise removing algorithms, a median filter with mask size 3 x 3 was selected.

Pseudo code for Median Filter:

```
For every pixel in the image do

          Sort values in the mask

          Pick the middle value in the sorted list

          Replace the pixel value with median one

End
```

5.2.1.2 Binarization

After the noise reduction process, the image needs to be binarized. A simple threshold technique was used. The threshold value of 16777215 / 2 i.e. 8288608 was taken 24 bit image was taken as input. The pixel values that were less than 8288608 were considered as ”0” and those pixels whose value were greater than 8288608 were considered as ”1”. So, the entire white pixels were considered”1” and the entire black pixel as”0”.

Pseudo Code for binarization:

```
Take input image for the noise reduction process
Scan the intensity of a pixel in the image
       IF  intensity of a pixel is less than the threshold
       value
                 Replace the intensity with 0
       Else
                 Replace the intensity with 1
       End IF
Repeat the process for every pixel in the image
```

5.2.1.3 Segmentation

Segmentation is the process of separating the characters from each other. It is very important in OCR because the system can only recognize one character at a time. The basic concept of connectivity was used. In this concept, it is considered that pixels of the single letter are connected to each other at some point. But this type of algorithm does not work for some letters like ”i” and ”j” where the pixels are not connected to each other. Segmentation also consists of the process of separating the image from the background, we will need to convert a color image into the grey scale so that the image is converted to black and white image. To separate a character from a group of the word, clipping of image has to be performed. For clipping, we can use the window of fixed size, which is equal to the size of the single character. The window then cuts a single character according to the requirement.

5.2.1.4 Thinning (Skeletonization)

The process of thinning converts the thick characters into one pixel thick characters so that it becomes easy for recognition. It also removes the irregularities in the characters. Besides, making the recognition simpler and removing irregularities, due to the decreased number of pixels for each character, it also reduce memory space necessary for storing the information of the input characters and also reduces the processing time. But the thinning cannot be done haphazardly; there are conditions that should be taken into care. During the process, the connectivity of the character should be maintained. Among the various algorithms for thinning Hilditch's algorithm gave the best result.

Hilditch's Algorithm

Hilditch's algorithm performs a sequential operation and describes the criteria that must be satisfied before a black pixel of a pattern is deleted [15]. This algorithm has mainly two versions, one with 4 x 4 window and the other one is 3 x 3 window.

P9	**P2**	**P3**
P8	**P1**	**P4**
P7	**P6**	**P5**

Figure 12. 8-neighborhood pixel Image [16].

Figure 12 shows the 3 x 3 window of Hilditch's Algorithm with P1 which is a black pixel as the center and its eight neighbors places in a clockwise spiral way. There are mainly four conditions that are required to be satisfied. They are as follows

a) $$2 \leq B(P1) \leq 6 \quad (1)$$

In Equation 1, B(P1) counts the black pixels around P1. This equation has two conditions to be matched before removing P1. The first condition, where the number of a black pixel around P1 should be greater than or equal to 2, ensure that there is no deletion or removal of an end-point pixel or isolated pixel. The second condition where the total count of the black pixel should to less or equal to 6, checks if the pixel is a boundary pixel.

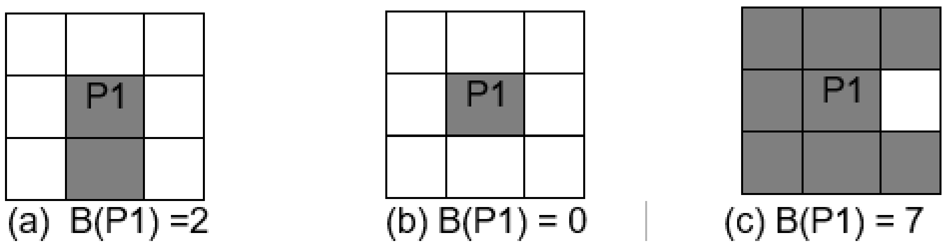

Figure 13. First Condition of Hilditch's Algorithm [17].

Fig 13, shows the different examples to better understand the first condition of the algorithm. In fig 13(a), B(P1) = 1, it shows that P1 is the tip-point of the skeleton of the character and should not be deleted. In fig 13(b), B(P1) = 0, which means P1 is an isolated pixel and should not be deleted. In fig 13(c), B(P1) = 7, which means P1 is not a boundary pixel of the character and should not be deleted.

(b) $$A(P1) = 1 \qquad (2)$$

In Equation 2, A(P1) denote the number of 01 patterns found in the set (P1, P2,........, P9). This condition checks for the connectivity of pixels and ensures that the pixels are connected to each other to ensure no fragmentation in the skeleton.

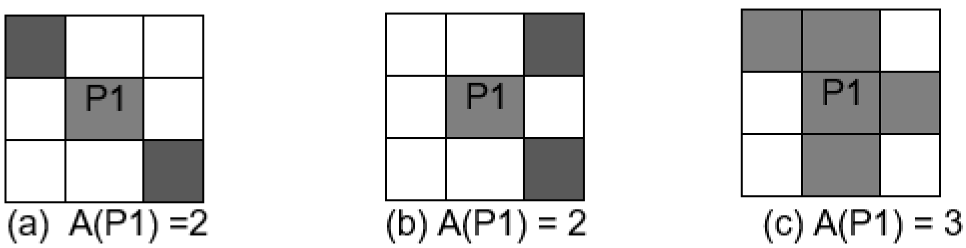

Figure 14. Second Condition of Hilditch's Algorithm [17].

Figure 14, shows the different connectivity test for Hilditch's Algorithm. In all the sub figures, A(P1) greater than 0, which means P1 is connected to other pixels and should be preserved. If A(P1) = 0, then is an isolated pixel and not connected to the character.

(c) $$P2 \cdot P4 \cdot P8 = 0 \ or \ A(P2) \neq 1 \qquad (3)$$

Equation 3, is used to protect the deletion of two pixels thick straight line.

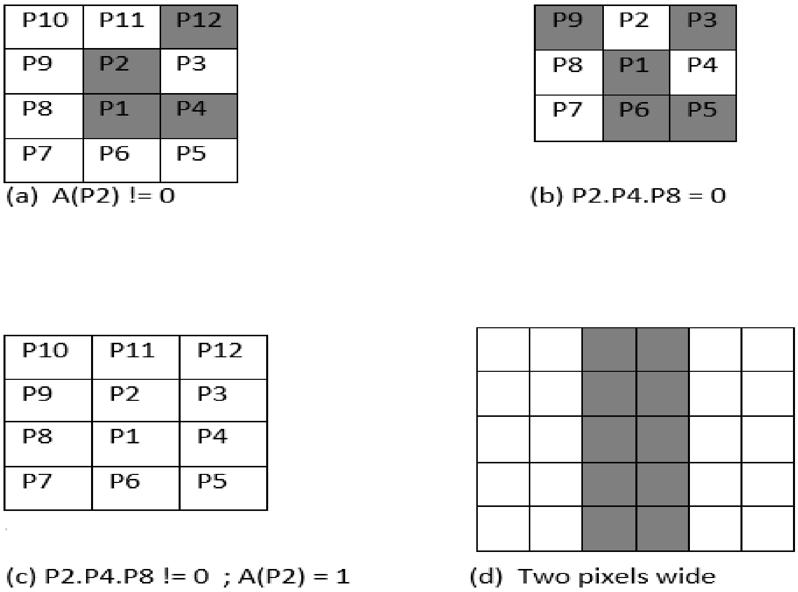

Figure 15. Third Condition of Hilditch's Algorithm [17].

Figure 15 shows examples for the third conditions of Hilditch's Algorithm. This condition prevents the deletion of two pixels thick vertical lines i.e. figure 15 (d), completely.

(d) $$P2 \cdot P4 \cdot P6 = 0 \ or \ A(P4) \neq 1 \qquad (4)$$

Equation 4, is used to protect the deletion of two pixels thick straight line.

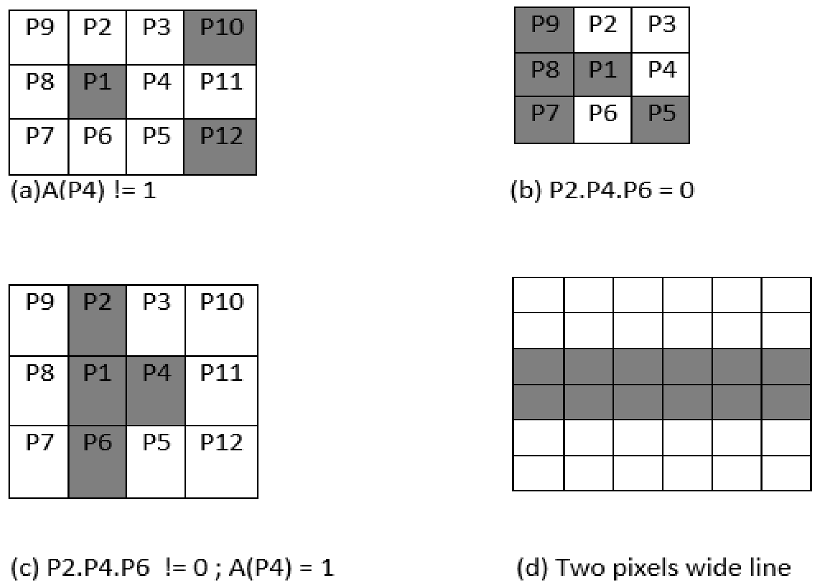

Figure 16. Fourth Condition of Hilditch’s Algorithm [17].

Figure 15 shows examples for the fourth conditions of Hilditch’s Algorithm. This condition prevents the deletion of two pixels thick horizontal lines i.e. figure 16 (d), completely.

In Hilditch’s Algorithm, the above conditions are repeatedly checked until there is no pixel which can be removed.

5.2.1.5 Normalization

The system should have the availability to recognize different font sizes. So, for this, all the input characters must be converted into a standard size before feeding it to the classifier. The process of changing the size of the image to a fixed size ready to be inputted to the classifier is known as normalization. The neural network consists of input nodes, which takes the pixel values of the input, whose number is always fixed. The algorithm for normalization can be described as follows:

1. Define standard size
2. Find the $x_{max}, x_{min}, y_{max}, y_{min}$

   Where x_{max} and x_{min} are the maximum and minimum positions of the character in x directthe ion and y_{max} and y_{min} are the maximum and minimum positions in the y-direction.

3. Find the increment in the x-direction and the y-direction as $x_{inc}$and y_{inc}

$$x_{inc} = (\, x_{max} - x_{min}) \div 2$$

$$y_{inc} = (\, y_{max} - y_{min}) \div 2$$

4. Scan from left to right and from top to bottom
5. For $y = y_{min}$ to y_{max}step y_{inc}

   For $x = x_{min}$ to x_{max} step x_{inc}

   Get the pixel value at the position and store in an array

   End For

   End For
6. Repeat the above procedure from 4 to 5 for all the characters.

5.2.2 Classification

After the preprocessing has been completed, the output from that stage which was an image of standard size was fed to the classifier. The pixel position of the character in the standard sized image is used as input for the classifier. There is a different algorithm in a neural network which can be used for the classification, but the only back-propagation algorithm is used here.

The Back-Propagation Algorithm:

Back-propagation is one of the most popular supervised training methods for the neural network. In the back-propagation algorithm, the output from the network is compared with the desired output provide, which is the part of the training set for the network. The error is calculated using the desired output and actual output, the difference is then fed backward to the system, the error is used to adjust the weights of the connection between neurons [10]. The process of adjusting weight is iterative in nature and the process is stopped when the output is almost equal or equal to the actual output.

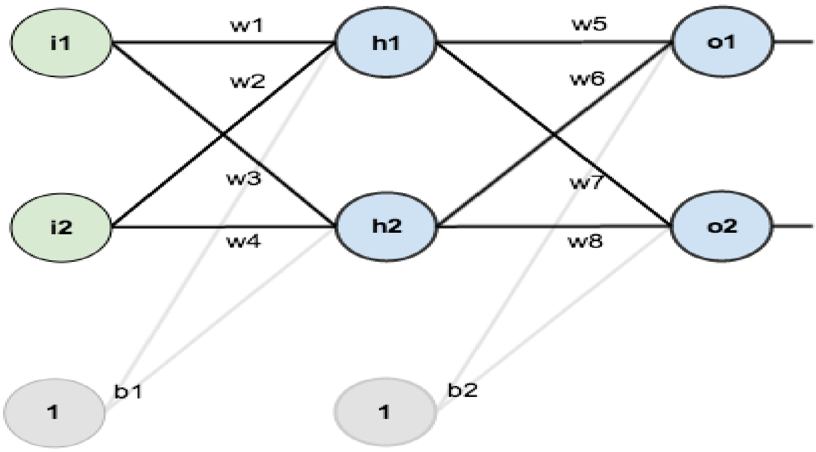

Figure 17. Neural Network ([18]).

Figure 17, is the layout of a simple neural network with two input nodes i.e. i1, i2; two hidden nodes i.e. h1, h2 and two output nodes i.e. o1, o2, w represents the weight between the nodes and b represent the bias.

Let us assume a neural network with hidden nodes where A_j is the activation function, x_i is the input, w_{ji} is the weight and E is the error. The activation function is the linear combination of input and weight, which is Equation 5 [19].

$$A_j(\overline{x}, \overline{w}) = \sum_{i=0}^{n} w_{ji}\, x_i \qquad (5)$$

$$O_j(\overline{x}, \overline{w}) = \frac{1}{1+ e^{A_j(\overline{x}, \overline{w})}} \qquad (6)$$

In Equation 6 [19], $O_j(\overline{x}, \overline{w})$ is output function also known as the sigmoid function which derived using activation function.

The neural network should have a training algorithm before the recognition of a character. As the back-propagation algorithm is a supervised learning system and is supplied with input and the desired output. There are hidden nodes in between the inputs and outputs nodes and the connection are given random weights at the beginning. The aim of the training is to adjust the random weights in order to get the desired output. The errors generated are used in order to adjust the weights. The error is the difference between the desired output and the actual output acquired during the processing. The error function E_j for each output is given in Equation 7 [19].

$$E_j(\overline{x},\overline{w}\ ,d) = (O_j(\overline{x},\overline{w}\ ) - d_j\ )^2 \tag{7}$$

The total error, $\mathrm{E}(\overline{x},\overline{w}\ ,d)$ of the system is the sum of the errors of all the neurons in the output layer, Equation 8 [19].

$$\mathrm{E}(\overline{x},\overline{w}\ ,d) = \sum_j (O_j(\overline{x},\overline{w}\ ) - d_j\ )^2 \tag{8}$$

Adjustments on the weights made using the value generated by the gradient descent method on the error, Equation 9 [19].

$$\triangle\, w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} \tag{9}$$

Equation 9, is repeated until the weights that generate a minimum error is found.

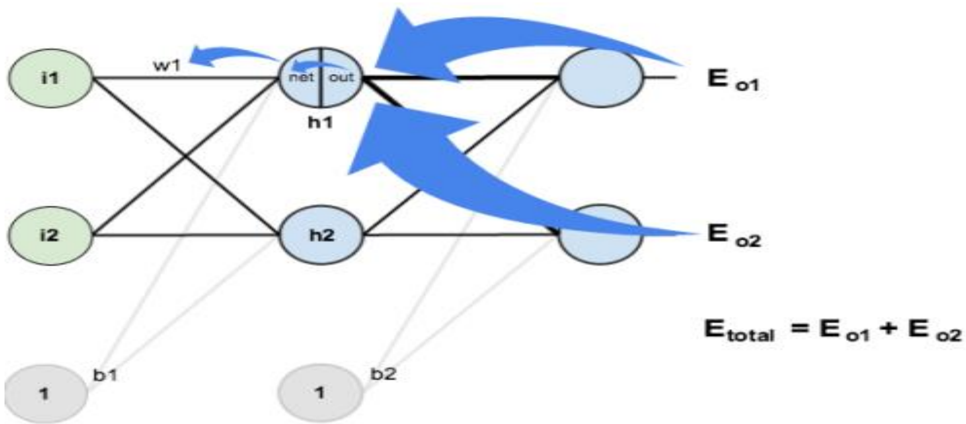

Figure 18. Backward pass [18].

Figure 18 shows the backward pass in the back-propagation algorithm. As seen in the figure, output after performing gradient descent on the error is used to adjust the weights of the connections between output and hidden nodes and will continue backward until the input nodes. Here, the initial random weights that are used for the preliminary calculations need to be replaced by new weights depending upon the gradient descent method. The calculation of derivative of E with respect to w_{ji} is done in order to perform a backward pass. A calculation is done to find how much error depends on the output O_j, which is the derivative of E with respect to O_j [19] from equation (7) is Equation 10 [19].

$$\frac{\partial E}{\partial O_j} = 2(O_j - d_j) \tag{10}$$

Equation 5 and Equation 6 are used to calculate the dependency of output on the activation, Equation 11 [19].

$$\frac{\partial O_j}{\partial w_{ji}} = \frac{\partial O_j}{\partial A_j}\frac{\partial A_j}{\partial w_{ji}} = O_j\left(1 - O_j\right)x_j \tag{11}$$

From Equation 10 and Equation 11, we get Equation 12 [19].

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial O_j}\frac{\partial O_j}{\partial w_{ji}} = 2(O_j - d_j)O_j\left(1 - O_j\right)x_j \tag{12}$$

Using Equation 9 and Equation 12, the amount by which the weight needs to be adjusted is acquired, Equation 13 [19] and Equation 14 [19].

$$\triangle w_{ji} = -2\eta(O_j - d_j)O_j\left(1 - O_j\right)x_j \tag{13}$$

$$\triangle v_{ik} = -\eta\frac{\partial E}{\partial v_{ik}} = -\eta\frac{\partial E}{\partial x_i}\frac{\partial x_i}{\partial v_{ik}} \tag{14}$$

Where, Equation 15 [19],

$$\frac{\partial E}{\partial x_i} = 2(O_j - d_j)O_j\left(1 - O_j\right)w_{ji} \tag{15}$$

Assuming that there are set of inputs x_i into the neuron with v_{ik} [18], from Equation 11, we get Equation 16 [19].

$$\frac{\partial x_i}{\partial v_{ik}} = x_i(1 - x_i)v_{ik} \tag{16}$$

Pseudo Code for Back-Propagation Algorithm

```
1) Create a neural network
2) Choose random weights for initial calculation
3) For training
      a. Supply the input to the network
      b. Calculate the result for the output neuron using the
         input values, weights, and outputs in the hidden layer.
      c. Calculate the error in the outputs.
      d. Calculate the total error generated in the output.
      e. Calculate the weight adjustments using the errors as
         inputs.
      f. Apply the adjusted weights.
      g. Repeat the process from a to f until the desired output
         value is reached.
4) Evaluate the network performance.
```

6 TESTING

Testing is the process of finding out the errors and the bugs in the system. It can also be defined as the process of verifying and validating the system. According to Boehm [20]:

Verification: "Are we building the product right?"

Validation: "Are we building the right product?"

Testing is better if it is done in a smaller module rather than the whole system at once. It will help to keep the better track of the system and the errors in the system. Black box testing is done when the internal structure is unknown. It only tests the fundamental aspect and does not test the internal logical structure. White box testing is used to check the internal structure and inspect the source code of the system.

Image Processing Lab in C#

Image Processing Lab in C# is an application developed by Andrew Kirillov and governed by "The GNU General Public License (GPLv3)". The source code and the demo of the application can be found at "https://www.codeproject.com/Articles/9727/Image-Processing-Lab-in-C". This software deals with the algorithms used for digital image processing.

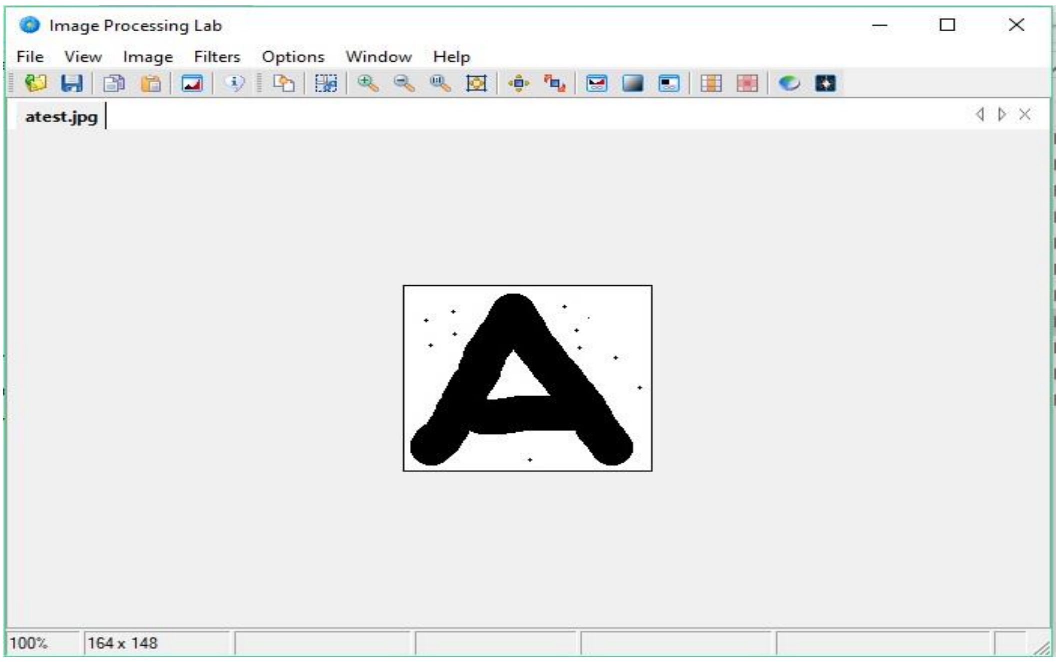

Figure 19. User Interface (UI) - Image Processing Lab.

<u>Unicode Optical Character Recognition</u>

Unicode Optical Character Recognition is developed by Daniel Admassu and the source code for this project can be found at "https://www.codeproject.com/Articles/15304/Unicode-Optical-Character-Recognition". This project is implemented using the digital image processing tools for feature extraction and neural network for classification. It uses Multi-layer Perception (MLP) neural network which is different classification approach than the one explained in this thesis.

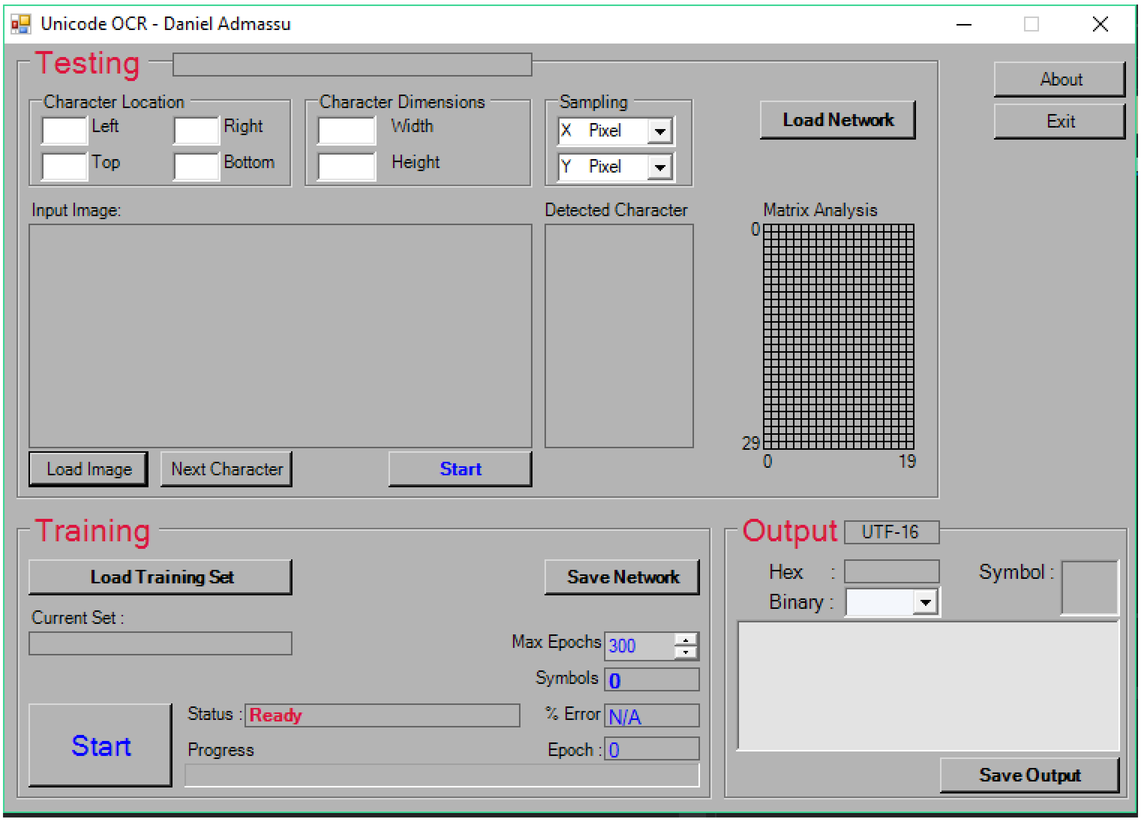

Figure 20. User Interface (UI) – Unicode OCR.

Neural Network OCR

Neural Network OCR was developed by Andrew Kirillova and the application is governed by "The GNU General Public License (GPLv3)."The source code and the demo for the project can be found at" https://www.codeproject.com/Articles/11285/Neural-Network-OCR ".

The application implements back-propagation algorithm for the recognition of digital characters. This application is capable of recognizing handwritten characters and built-in training sets for different fonts such as Arial, Courier, Tahoma, Times New Roman, Verdana also the characters can be a regular font or italic fonts.

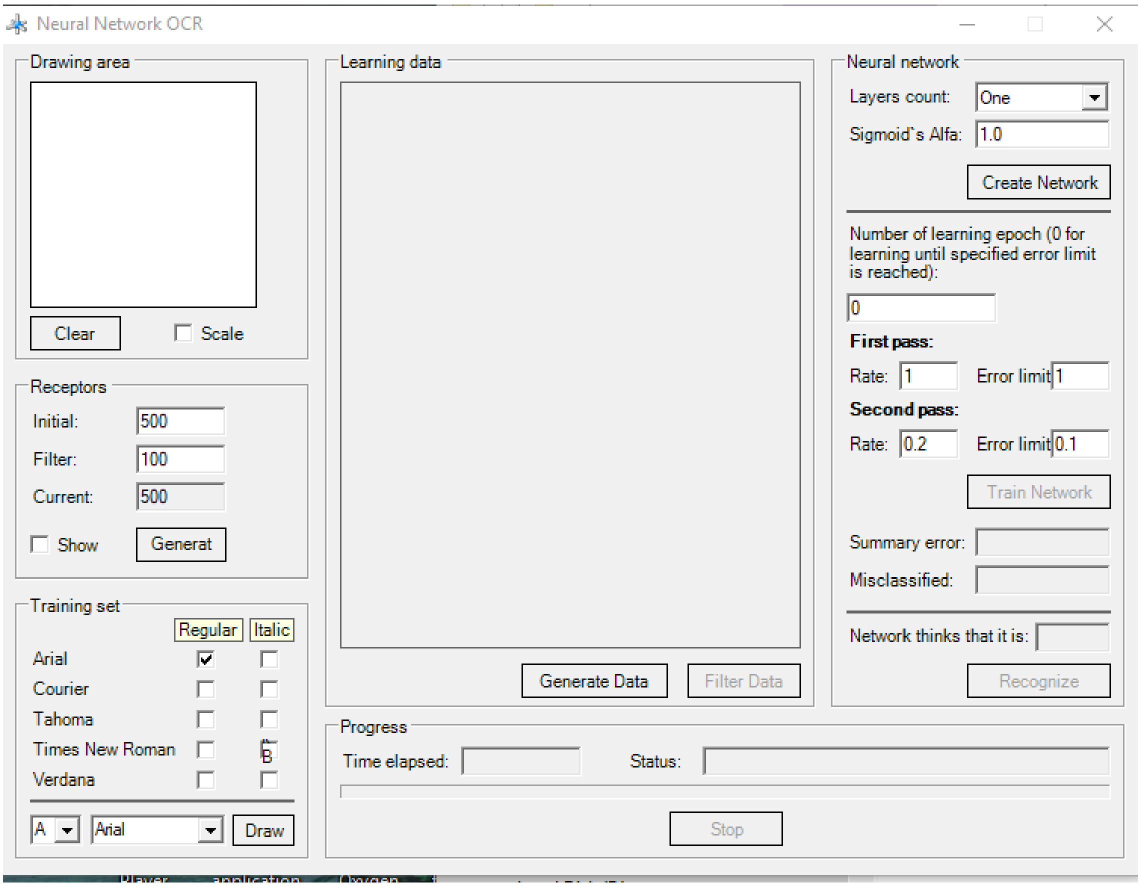

Figure 21. User Interface of Neural Network OCR.

6.1 Unit Testing

Unit testing focus on a single component of an application, software. White box testing module is used to conduct testing on modules such as noise removal, segmentation, binarization, normalization and thinning (Skeletonization).

a) Noise Removal

It is the process of removing unwanted noise from an image. The noise in the image should be removed before any further processing so that the error generated due to the presence of imperfection or noise does not propagate to further steps. The median filter was used for removing the noise. For the testing of the noise removal filter" Image Processing Lab in C#" was used.

Figure 22. Image with salt and pepper noise.

Figure 23. Image after the median filter.

b) Segmentation

Segmentation is the process of separating a single character from a group of characters or a word. It deals with the process of separating the object of interest from the background and separating the character for the object of interest. As a recognition system can only recognize a single character at a time, segmentation is required. For the testing of segmentation process ”Unicode Optical Character Recognition” was used.

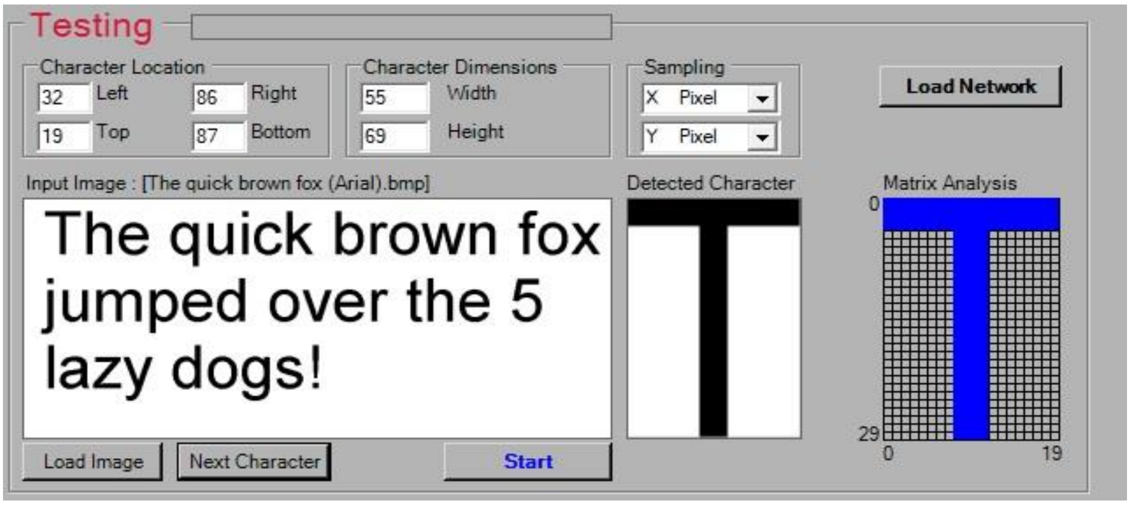

Figure 24. Segmentation.

In figure 18, the sentence in the image box is an image and the objective is to extract individual character once at a time. The extracted character is displayed on ”Detected character ” location and it also displays the character location in the image.

c) Thinning(Skeletonization)

Thinning is the process of generation one pixel thick representation of the character. There a lot of different algorithm for thinning and Hilditch’s algorithm gave the best result. ”Image Processing Lab in C#” was used for generating skeletons of different characters.

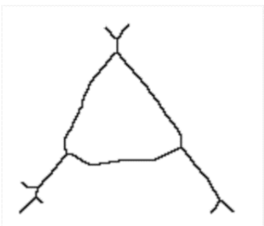

Figure 25. After Skeletonization process.

d) Training

For the purpose of training, random weights are supplied to the connections between the nodes. The back-propagation algorithm adjust the value of the connection depending upon the error value generated at the output. ”Neural Network OCR” was used for the training of different characters. It was used for training both the handwritten and machine written character.

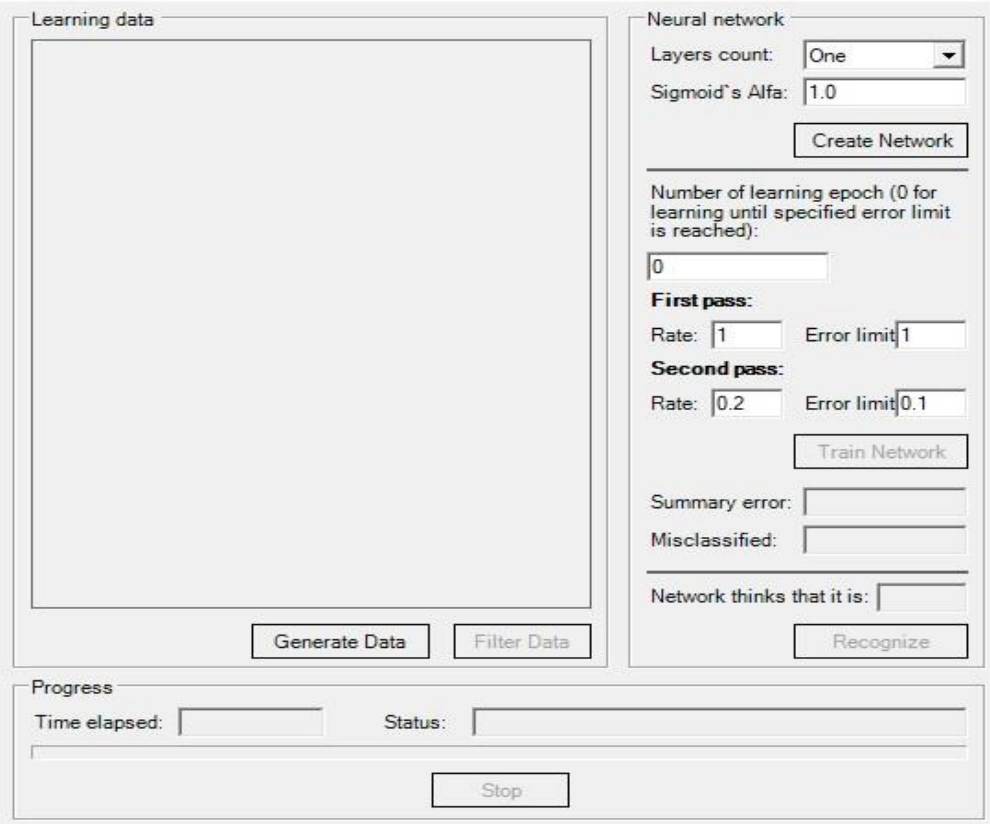

Figure 26. Training.

Before a character can be trained, data for learning should be generated and error limit should be set.

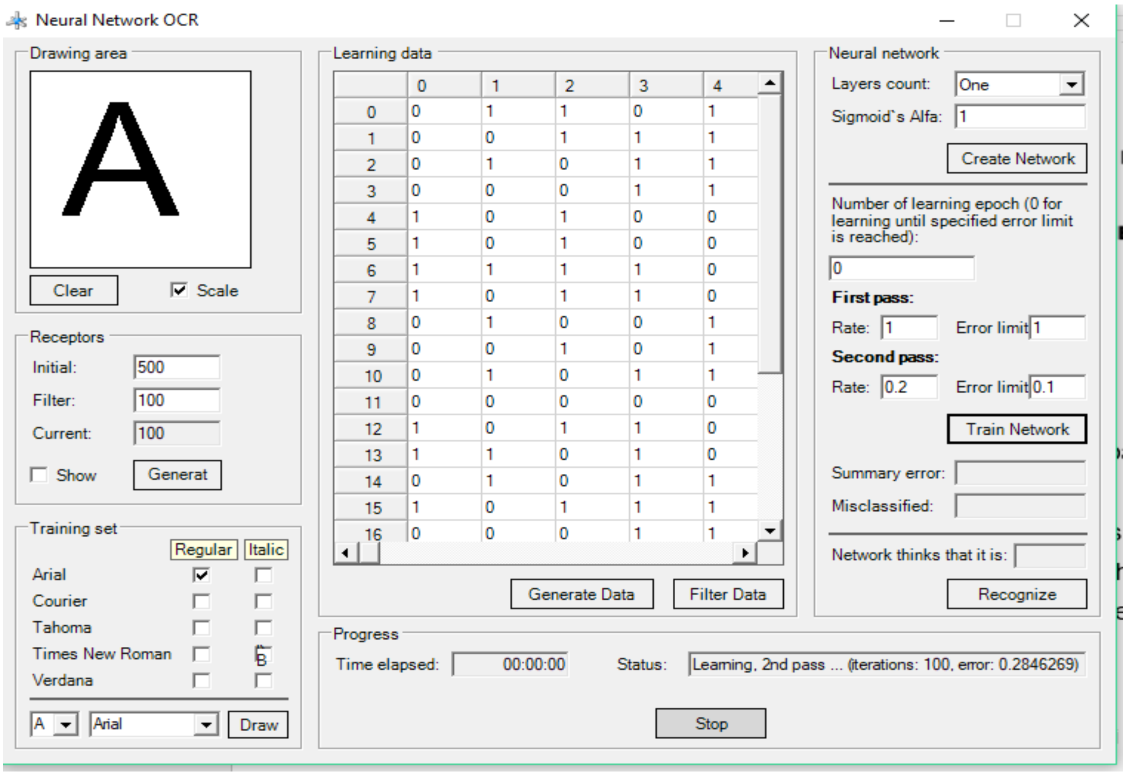

Figure 27. Training machine written a character.

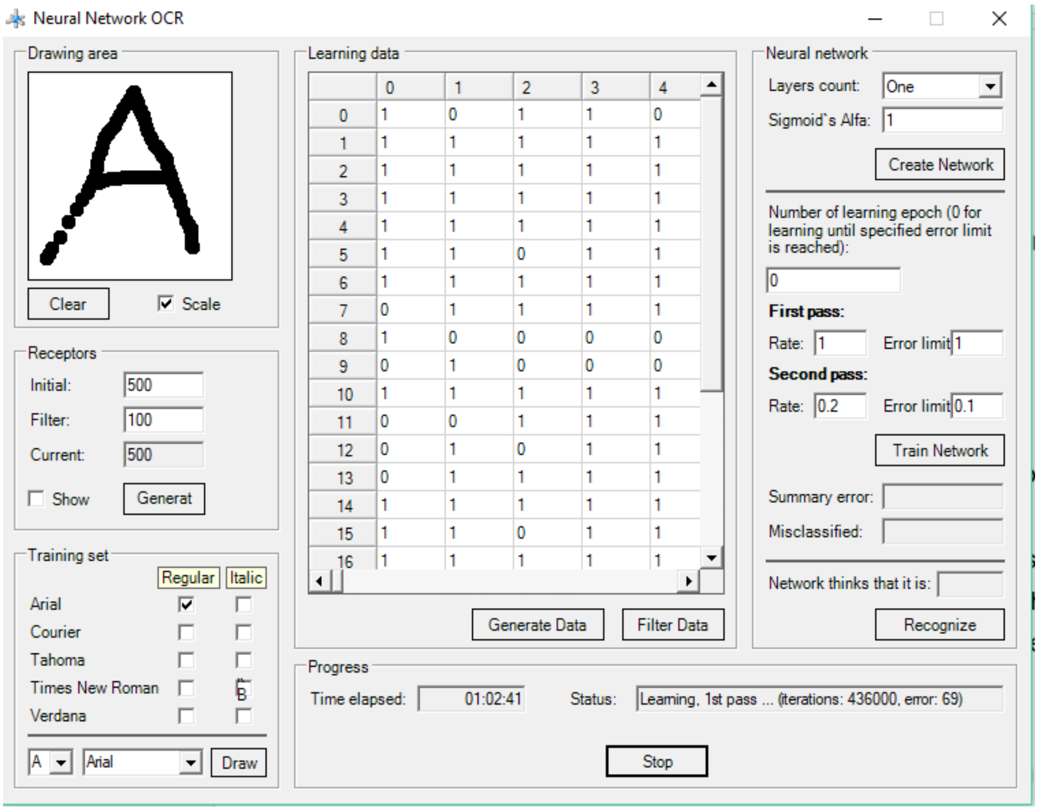

Figure 28. Training Handwritten character.

Figure 27 and figure 28, provides the details during the training of handwritten character. The progress section in the figure gives the time taken for the training, a number of iterations required achieving the goal of minimum errors.

The number of iteration for achieving the minimum errors for handwritten i.e. 436000, which was still under training, was very large than the one required for machine written character i.e. 100. The time needs to train a machine written character was found to be of few seconds whereas it took a couple of hours for training handwritten characters.

e) Recognition

Recognition is the final step in the OCR. "Neural Network OCR" was used for the recognition of different characters. The precision of the system recognizing the character depends on the training of the characters. After training a character, the system is capable of recognizing the same character.

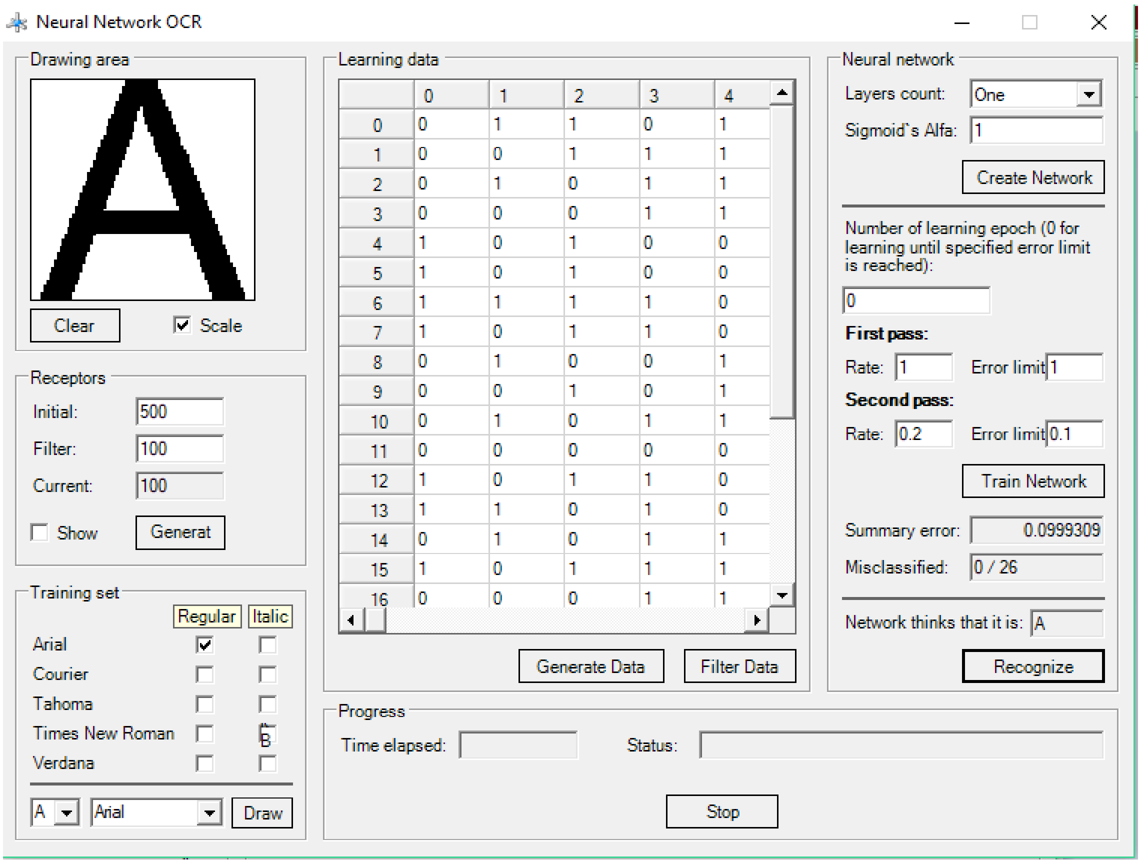

Figure 29. Recognition Machine Written.

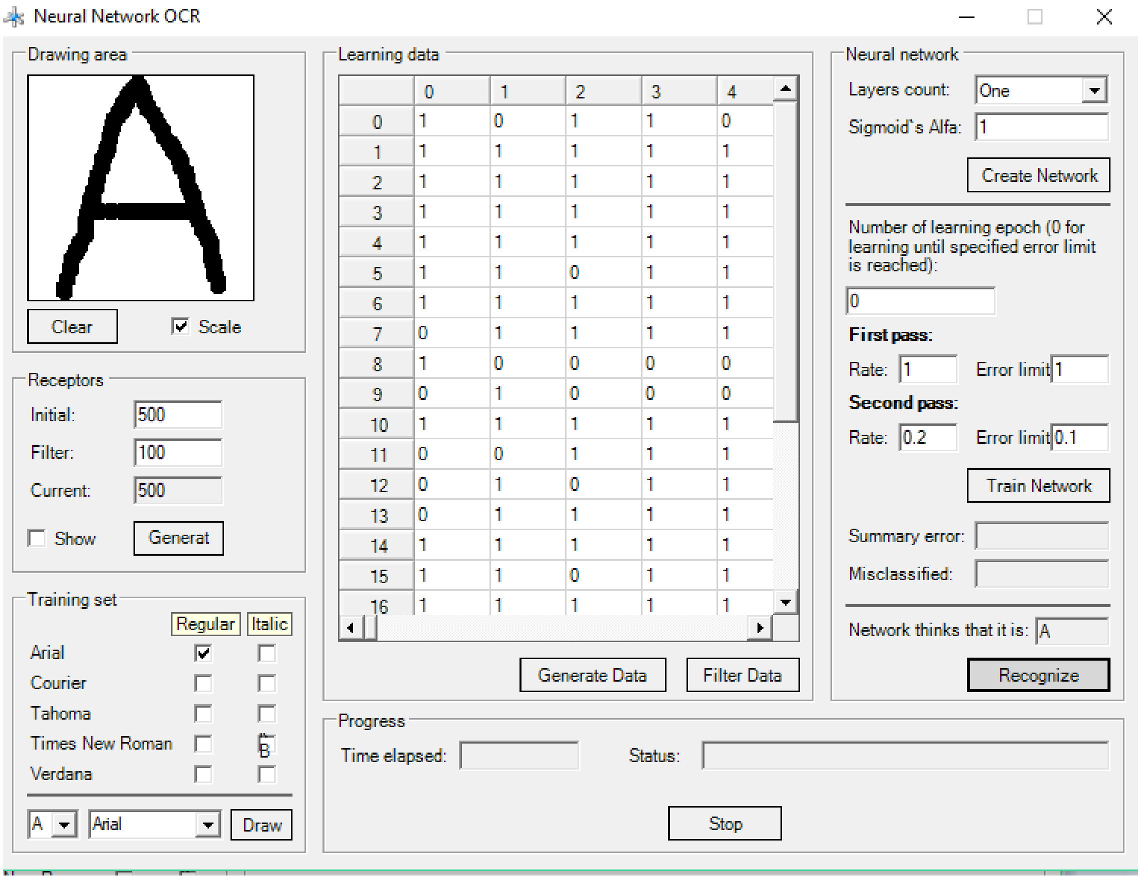

Figure 30. Recognition handwritten.

Figure 29 and figure 30, shows the recognition process of a character. Once the character is trained. The updated weights are stored in a weight database to be used for the recognition process. When a character is to be recognized, one should draw the character in the drawing area and then press recognize. The output for the recognized character will be displayed at the text box titled ”Network thinks that it is:”

7 CONCLUSION

Pattern recognition done using neural network can tolerate noise and trained properly, can be used to recognize unknown patterns. Neural networks if constructed using proper architecture and trained correctly can be used for different scientific and commercial applications. They can be used for data entry, text entry, data automation etc.

Various image processing algorithms were studied and tested and the algorithm that supplied the best result is included in the architecture and discussed. Segmentation is one of the difficult processes in image processing. Isolating a single character from a bunch of characters can be achieved using classical approach, recognition based or holistic approach.

A neural network using back-propagation algorithm is one of the most popular algorithms for training. It is a time-consuming algorithm for training network with a large number of nodes. Adjusting the size of the input, error margin and addition of hidden node will give a better result.

For the purpose of testing the application, software created by different developers were used. Multiple tests were performed to find the best option for the system. For the preprocessing phase, different algorithms of digital image processing were used. The median filter was found to be the best one to remove the noise from the image. Hilditch's Algorithm produced the best skeleton of the character.

For the test of classification phase, training was performed using machine written and handwritten characters. During the training, it was found that the time is taken and the number of iteration to reach the desired error limit for handwritten characters was very much greater than the one required for the machine written characters. The system was trained to recognize different handwritten characters as well as machine written characters.

The system was designed with few specific algorithms. The performance and the processing time can be further improved using the different algorithm for preprocessing and the classification phase. This system is capable of recognizing the English alphabets but can be further trained to recognize different characters sets from other languages as well.

REFERENCES

1. Arindam Chaudhary, Krupa Mandaviya, Pratixa Badelia and Soumya K Ghosh, "*Optical Character Recognition Systems for Different Languages with Soft Computing",* Springer International Publishing AG, Picassoplatz 4, CH-4052 Basel, Switzerland,2017.

2. Jayanta Kumar Basu, Debnath Bhattacharyya, Tai-hoon Kim,"*Use of Artificial Neural Network in Pattern Recognition*", International Journal of Software Engineering and its Application, Vol. 4, No. 2, April 2010.

3. Anil K. Jain,"Fundamentals of Digital Image Processing", Prentice-Hall of India Private Limited, New Delhi- 110 001, 1997.

4. Al Bovik,"*The Essential Guide to Image Processing*", Academic Press Elsevier, 30 Corporate Drive, Suite 400, Burlington, MA 01803, USA.

5. Rohit Verma and Dr. Jalid Ali," A comparative study of various techniques of image noise and efficient noise removal techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, October 2013.

6. Yide Ma, Kun Zhan, Zhaobin Wang," Application of Pulse-Coupled Neural Networks ", Higher Education Press, Beijing and Springer-Verlag Berlin Heidelberg, 2010

7. Masks For Acne. (n.d.). *Median Filter Mask Matlab - Masks For Acne*. [online] Available at: http://masks.haohaihk.com/median-filter-mask-matlab/ [Accessed 18 May 2018].

8. Komal Bishnoi, Preeti Gupta,"A Practical Approach of Parallel Thinning Algorithm on Numeral Patterns Using Matlab Simulation", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, Issue 5, May 2016

9. Stack Overflow. (n.d.). *How do you fit a curve to a skeletonized image?*. [online] Available at: https://stackoverflow.com/questions/21868132/how-do-you-%20fit-a-curve-to-a-skeletonized-image [Accessed 20 May 2018].

10. Rajvardhan Oak," A Study of Digital Image Segmentation Techniques", International Journal of Engineering And Computer Science, Volume 5 Issue 12 Dec. 2016

11. Psych.utoronto.ca. (n.d.). *What are Artificial Neural Networks*. [online] Available at: http://www.psych.utoronto.ca/users/reingold/courses/ai/cache/neural2.html [Accessed 14 Jan. 2018].

12. J.Burkill, (2017). *What does a threshold in Neural Networks imply? - Quora*. [online] Quora.com. Available at: https://www.quora.com/What-does-a-threshold-in-Neural-Networks-imply [Accessed 4 Jun. 2018].

13. L. Martins, (n.d.). *What is the sigmoid function, and what is its use in machine learning's neural networks? How about the sigmoid derivative function?.* [online] Quora.com. Available at: https://www.quora.com/What-is-the-sigmoid-function-and-what-is-its-use-in-machine-learnings-neural-networks-How-about-the-sigmoid-derivative-function [Accessed 29 May 2018].

14. J. Parekh, (2012). *Deployment of Neural Network on Multi-Core Architecture*. [online] researchgate.net. Available at: https://www.researchgate.net/figure/A-ta-xonomy-of-feed-forward-and-recurrent-feedback-network-architectures_fig1_297537592 [Accessed 17 May 2018].

15. C.H.Chen, L.F.Pau, P.S.Wang,"Handbook of Pattern Recognition and Computer Vision", World Scientific Publishing Co.Pte.Ltd., P O Box 128, Farrer Road, Singapore 9128, 1995

16. V.Karthikeyan,"Hilditch's Algorithm Based Tamil Character Recognition", International Journal of Computer Science and Engineering Technology, Vol. 4 No.03 Mar 2013

17. D.Azar, (1997). *Skeletonization*. [online] Cgm.cs.mcgill.ca. Available at: http://cgm.cs.mcgill.ca/~godfried/teaching/projects97/azar/skeleton.html [Accessed 8 May 2018].

18. Matt Mazur, (2015). *A Step by Step Backpropagation Example*. [online] A Step by Step Backpropagation Example Background. Available at: https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/ [Accessed 30 Jan. 2018].

19. Dike U. Ike, Adoghe U. Anthony, "Back-propagation Artificial Neural Network Techniques for Optical Character Recognition- A Survey", International Journal of Computers and Distribution Systems, Vol.No.3, Issue II, June-July 2013. (https://www.academia.edu/6704140/BACK-PROPAGATION_ARTIFICIAL_NEURAL_NETWORK_TECHNIQUES_FOR_OPTICAL_CHARACTER_RECOGNITION_A_SURVEY)

20. Barry W. Boehm, "Software Engineering Economics", Prentice Hall, 1981.