ARABIC/ URDU INFORMATION RETRIEVAL SYSTEM : ARABIC/URDU OPTICAL CHARACTER RECOGNITION SYSTEM

By

Ahmar Hayat Khan Tareen

(2000-NUST-BIT-795)

A report submitted in partial fulfillment of

the requirements for the degree of

Bachelors in Information Technology

In

NUST Institute of Information Technology

National University of Sciences & Technology

Rawalpindi, Pakistan

(2004)

Certified that the contents and form of the project report entitled "**Arabic/ Urdu Information Retrieval System : Arabic/Urdu Optical Character Recognition System**" submitted by Ahmar Hayat Khan Tareen (2000-NUST-BIT-795) have been found satisfactory of the requirement of the Bachelors in IT Degree at NUST Institute of Information Technology.

COMMITTEE

1. Advisor sign _____

   (Mr. Saqib Mir)

2. Co-Advisor sign _____

   (Mr. Shahzad Khan)

3. Committee Members

   (Mr. Tauqeer Ahmed) sign _____

   (Mr. Mohammad Umer) sign _____

Dedicated to

my parents

who have always been there to guide and support

me

ACKNOWLEDGEMENTS

All thanks to Allah Almighty who has given me the strength and will to accomplish such a feat. Without His support even a trivial task proves difficult and this was one big task.

I owe my sincerest gratitude to Mr. Saqib Mir, my advisor, for his guidance, encouragement, support and most of all for his patience throughout the life of the project.

My gratitude to Director General NIIT, Dr. Arshad Ali for providing me facilities and support necessary for the completion of the project.

I would like to thank Mr. Shahzad Khan for guiding me in times when the project seemed doomed and for his constant inspiration.

A special thanks to Mr. Tauqeer Ahmed and Mr. Mohammad Umer whose input and comments helped make this project much better.

My appreciation also goes to Mr. Wahid Murad Afridi and all the library staff who lent a helping hand in the procurement of books and references valuable to this project.

I would also like to include my family in this appreciation who were the source of my strength and my greatest motivation.

My friends also share my sincerest gratitude who have always urged me forward to conquer new frontiers, who have always encouraged me and provided support when I needed it most. A special thanks to Qambber, Imran, Gohar, Zia, Aftab, Zahra, Amjad (for helping me get some sense out of MS Word) and to Haseeb who besides doing all the above also provided me access to his hard disk for backing-up my project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The presence of Urdu information on the Internet is in the form of images due to the absence of a standard Urdu font, thus rendering the retrieval of such information on demand almost impossible. To counter this problem an Optical Character Recognition (OCR) system is needed which when combined with a customized search engine[1] can retrieve textual Urdu information from images and furnish it on demand. There has been only limited success in the field of Urdu Optical Character Recognition owing mostly to difficulties such as the cursive-ness and overlapping of most Urdu fonts. The proposed system aims to overcome these problems but due to the time limitation can only reasonably accurately identify only one font type though the structure of the product allows for the inclusion of more fonts through the use of a modular system.

The approach to this involves the use of Artificial Neural Networks, as image classification is tedious when it comes to normal computational means. Artificial Neural Networks are computational models that emulate the human brain at a very small scale [22]. The effectiveness of Artificial Neural Networks in image identification and classification has been proven [20]. The system will be fed an image as input which will then be broken down and reflected within to be identified in which case Urdu characters in the form of Unicode will be outputted.

[1] An Urdu image-based search engine, part of the Arabic/Urdu Information Retrieval System developed by Qambber Hussain Syed, BIT-2(B), NUST Institute of Information Technology, National University of Sciences and Technology.

Chapter 1

INTRODUCTION

1.1 THE URDU LANGUAGE

Urdu language which originated in South Asia, developed from the interaction between the local South Asian languages and the languages of the Middle East. It is widely spoken today in both India and Pakistan and all countries having a sizeable South Asian diaspora.

Urdu is the second most popular 'first' language and second most popular 'first or second' language in the world. Urdu by itself is the twentieth most popular 'first' language in the world [25].

1.2 NEED FOR AN ARABIC/URDU OPTICAL CHARACTER RECOGNITION SYSTEM

Information in the World Wide Web is vastly decentralized and spread across millions of computers. The ability of search engines therefore to retrieve required information is greatly useful. Currently popular search engines scan text as part of web-pages to retrieve required information. This approach holds well for searches native to the English language and a number of other languages that have become standard on the internet, but in the case of Urdu language this approach falls short. This is because of the fact that Urdu text on the internet mostly takes the

form of images due to the reason that a standard Urdu font is only now starting to be used on Urdu-based sites and older sites still cannot convert their images to text yet. Thus searching for information on Urdu-based sites is next to impossible for current search engines. Enter the Arabic/Urdu Information Retrieval System. Using a combination of an image search engine and an Arabic/Urdu OCR system, this search engine is capable of searching and retrieving Urdu-based information quite easily from websites. The project in discussion i.e. the Arabic/Urdu Optical Character Recognition is thus part of a much bigger system that contributes to the support of Urdu on the Internet. The project is in a modular form and can be deployed as part of other systems or if desired operate independently.

1.3 DIFFICULTIES INHERENT TO URDU FONTS

Urdu is ranked at number 20 in the world according to its popularity as a 'first' language [25] yet there very little research going on in the field of Urdu character recognition. This is due to the difficulties encountered when dealing with the Urdu font in general. These difficulties include:

1.3.1 Cursiveness

Unlike most type-written languages, Urdu characters are connected to each other to form words.

1.3.2 Overlapping

Characters and words in Urdu frequently overlap. This makes it difficult to ascertain boundaries.

1.3.3 Multiplicity of Character Shapes

Another problem with Urdu alphabets is the fact that the same character can take at least four different shapes according to its position in any word (example shown in figure 1.1). This is in contrast to English in which an alphabet can either be capital or small. The four possible shapes of an Arabic character are:

- *Isolated:* The character is not linked to either the preceding or the following character.
- *Final:* The character is linked to the preceding character but not to the following one.
- *Initial*: The character is linked to the following character but not to the preceding one.
- *Middle*: The character is linked to both the preceding and following characters.

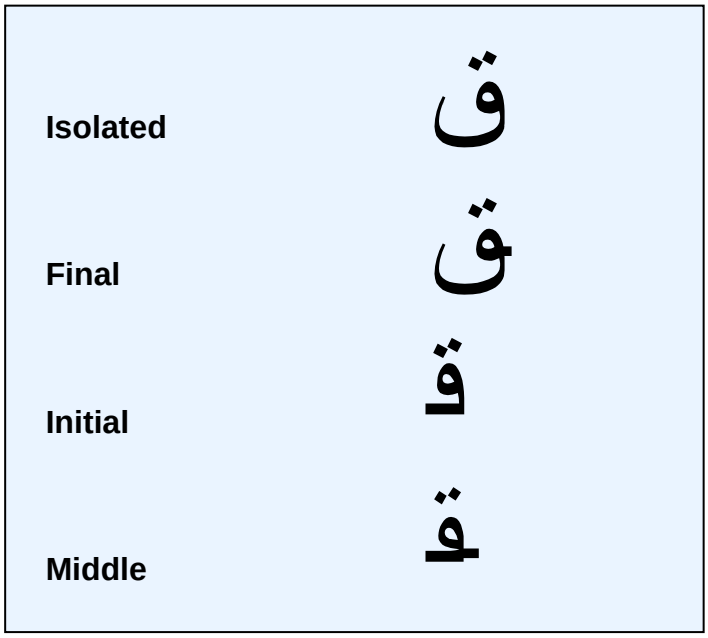


Figure 1.1 – Arabic Character Shapes

1.4 SCOPE OF RESEARCH

The aim of the project is to recognize and retrieve Urdu characters from images with above 90 per cent accuracy for text in normal noise conditions. The scope of the project has been limited to one font type due to time constraints. The chosen font is used on the Urdu news website urdu.paknews.com.

Though the system is currently built to recognize only one Urdu font, the implementation is global enough to allow the inclusion of other Urdu fonts when required.

Chapter 2

REVIEW OF LITERATURE

2.1 OPTICAL CHARACTER RECOGNITION

Optical character recognition (OCR), according to one definition, involves computer systems designed to translate images of typewritten text (usually captured by a scanner) into machine-editable text--to translate pictures of characters into a standard encoding scheme representing them (ASCII or Unicode). OCR began as a field of research in artificial intelligence and machine vision; though academic research in the field continues, the focus on OCR has shifted to implementation of proven techniques.

Originally, the distinction between optical character recognition (using optical techniques such as mirrors and lenses) and digital character recognition (using scanners and computer algorithms) were considered separate fields. Since very few applications survive that use true optical techniques the OCR term has now been broadened to cover digital character recognition as well.

Character recognition has been an active area of computer science research since the late 1950s. It was initially perceived as an easy problem, but it turned out to be a much more interesting problem. It will be many decades, if ever, before computers will be able to read all documents with the same accuracy as human beings.

2.2 OCR TECHNIQUES

Discussed below are some approaches to the OCR problem.

2.2.1 Automated Character Recognition

Automated text recognition has increasingly gained popularity in the area of pattern recognition in recent years. The need for fast processing of documents, due to the increasing amount of produced information, makes this area more important than ever. Traditionally, the recognition algorithms were implemented on a digital computer. However, since character recognition is computationally intensive, using the digital computer does not meet the demand of fast processing of the task. Digital computers are good at handling problems which are explicitly formulated, but character recognition is not such a problem.

2.2.2 Conventional Pattern Recognition

The conventional approach to the area of pattern recognition utilizes a digital computer to implement the pattern recognition algorithms. An object image is first taken and image digitization performed on it. The process of digitization is to vertically and horizontally partition the image pixels, and assign a value to each pixel. The value assigned to a pixel of a monochrome image varies according to its brightness or gray level. The digitized image may need further processing using image processing techniques in order to perform the recognition task. To perform the pattern recognition the digital computer must first learn to distinguish objects of all types based on a set of learning samples. After learning, the computer is ready to classify the input samples. However, pattern recognition is a computationally

intensive and time consuming task due to the vast amount of image data and large number of computation steps. Using the conventional approach always demands a very high speed computer or a parallel processor system to perform a satisfactory recognition.

2.2.3 Neural Network Approach

Developments in the field of artificial neural networks have opened them to the possibility of being used as solution to the image processing problem. Modeled on the neural network present in the human brain, ANNs have been hailed as the perfect pattern recognizers in the computational world. Trained neural networks can recognize similar objects according to their learned knowledge. However, the large number of neurons required and the sophisticated interconnections of the networks used for the recognition task complicate the hardware implementation.

2.2.4 Combinational Approach

The approach used in this research uses the advantages of the conventional pattern recognition and the neural network approaches to complete the task. Since an object recognition system consists of the learning and the classification stages, one of the stages can be completed by the conventional approach and the other stage by the neural network.

2.2.5 Feature-Based Recognition

Object recognition is generally performed on either the raw image in the image plane or on the future representation in the future space. In the former case,

known as the low level image recognition, the system learns and recognizes an object according to the information given by all the pixels in the image plane. One of the drawbacks of this approach is the huge dimensionality which further puts burden on the system.

The feature-based recognition uses only the information that best characterizes the object. It extracts important information conveyed by some pixels and processes it to obtain the feature representation. Dimensionally of the input vector is greatly reduced, and the recognition can be invariant to some image transformations, such as image translation, rotating, and scaling, if the object features are properly selected. Various methods for object recognition based on object features have been proposed [2]-[19].

2.2.6 Stochastic Modeling

Stochastic simulation uses computer techniques to imitate or evaluate a model numerically in order to estimate the desired true characteristics of a system having random input components.

2.3 ENGLISH OCRS

There has been a lot of development in the field of English language OCRs. This is why a number of such products are available commercially. English OCRs employ a variety of techniques to get the job done including almost all of the above mentioned techniques.

2.4 ARABIC/URDU OCR SYSTEMS

Recent developments in the field of pattern recognition have allowed the development of OCR systems for Asian languages that use the Arabic script as base. OCRs for Arabic have been present for quite some time now and most have achieved commercial status but that is not the case for Urdu based systems. Due to the complexity of the Urdu font no complete OCR has yet been developed although research in this regard is being actively pursued. Some popular systems have been discussed below:

2.4.1 Arabic Optical Character Recognition (A-OCR) Sakhr by AramediA

Since 1993, Sakhr has been developing the technology of automatic recognition of printed Arabic text known as OCR. Although Sakhr has been targeting the Arab market, Sakhr realized from the very beginning the importance of developing a bilingual OCR for Arabic/English text.

The program is able to learn character shapes, which can be used to improve the recognition accuracy to its maximum. No other Arabic OCR system has this unique feature of combining both recognition technologies. After establishing its leadership in the field of Arabic/English printed text recognition, Sakhr internationalized its OCR system by supporting more languages. Sakhr has released a Persian version of its product for an Iranian distributor. Special versions for Arabic script languages, such as Urdu and Jawa can be developed based on the

same technology. In addition to the support of Arabic-like languages, Sakhr now supports all the 16 European languages.

2.4.2 Al-Qari' Al-Ali by al-Alamiah Software Company

This program is based on a very powerful algorithm which seems to combine vector and bit-map analysis. In its first upgraded version it offers a number of means, although still not quite enough, of controlling recognition performance. Thus it is possible to select desired level of accuracy and to train for the majority of fonts in Arabic and in most other scripts. The results of an OCR operation can be controlled with a spelling checker that, while far from what one might hope for, is surprisingly good, particularly for controlling words that have run together. To facilitate comparison between the original scanned image and the text document, the spelling checker highlights problem areas simultaneously in both.

Chapter 3

INTRODUCTION TO ARTIFICIAL NEURAL NETWORKS

Before discussing the details of the project it is wise to learn something about Artificial Neural Networks as they form the back-bone of the project\

3.1 ARTIFICIAL NEURAL NETWORKS (ANNs)

Artificial Neural Networks (ANNs) are information processing paradigm capable of emulating the human brain at a very small scale. ANNs are inspired by the computational capabilities of the human brain which in itself is huge neural network. The neurons in the human body are known by their large parallelism, which gives them the incredible capability of information processing [23]. The number of neurons in the human brain is estimated at approximately in the order of 10^{11}, which makes the computational ability of the brain as high as 10^{14} interconnects per second. Biological neurons not only can process information concurrently, but also have the ability to accommodate new knowledge. This ability has been used as the basis for ANNs which once trained can recognize similar objects according to their learning knowledge. ANNs have an adaptive nature, meaning that they learn by example rather than by programming [23]. ANNs are actually computational models that consist of a number of processing units that communicate by sending signals to each other over a large number of

weighted connections. These computational units are called neurons. Architecture of a simple ANN is shown in the figure 3.1.

As evident in figure 3.1, neurons in an ANN are arranged in the form of layers with every neuron in the former layer connecting to every neuron in the next layer. An ANN consists of at least one layer, the output layer. The input nodes lack any sort of computational ability and thus are not considered as neurons. The inclusion of one or more hidden layers in the network is optional. The hidden layers are useful as they increase the classification and learning ability of the network.

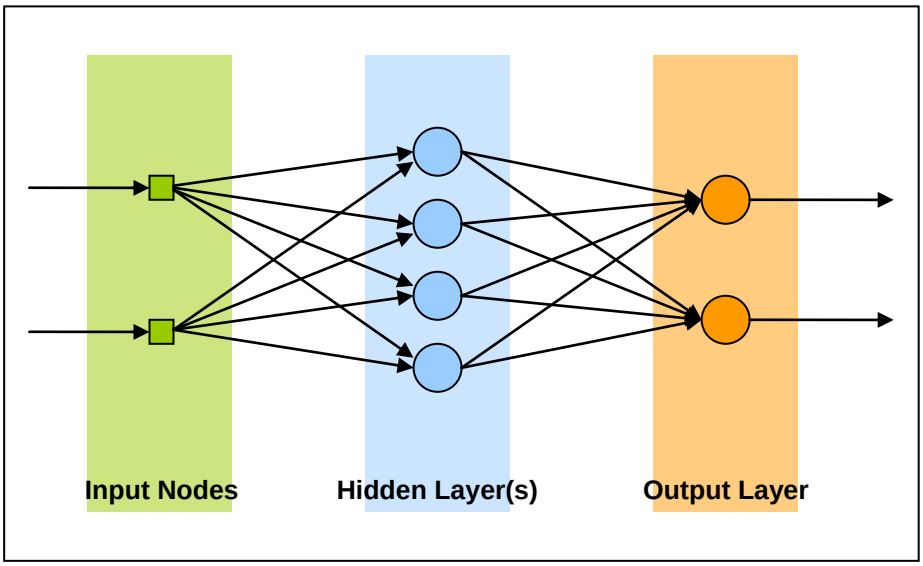


Figure 3.1 – Architecture of an Artificial Neural Network

With the properties of computational parallelism and learning ability, Artificial Neural Networks are widely used in solution to problems not accomplishable by normal computational means such as:

- Classification: radar returns; disease pattern; Image Processing.

- Noise Reduction: voice, images corrupted by noise
- Prediction/Forecasting: load, market, sale forecasting

Neurons are the basic computational units that make up any ANN. They are similar to biological neurons that form part of the neural network in the human brain. A structure of a neuron is shown in Figure 3.2. As shown in the figure a neuron can have many inputs but only one output. Each input comes from a neuron in the previous layer or is a loop-back. The neuron calculates a weighted sum of inputs and compares it to a threshold. If the sum is higher than the threshold, the output of the neuron is set to 1, otherwise to -1. A mathematical representation of a neuron is show in Figure 3.3.

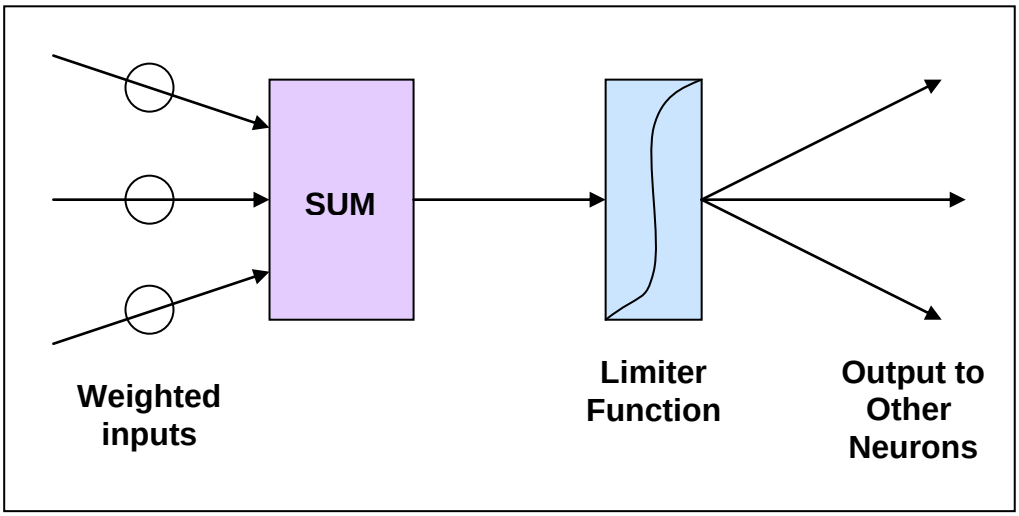


Figure 3.2 – Architecture of a Neuron

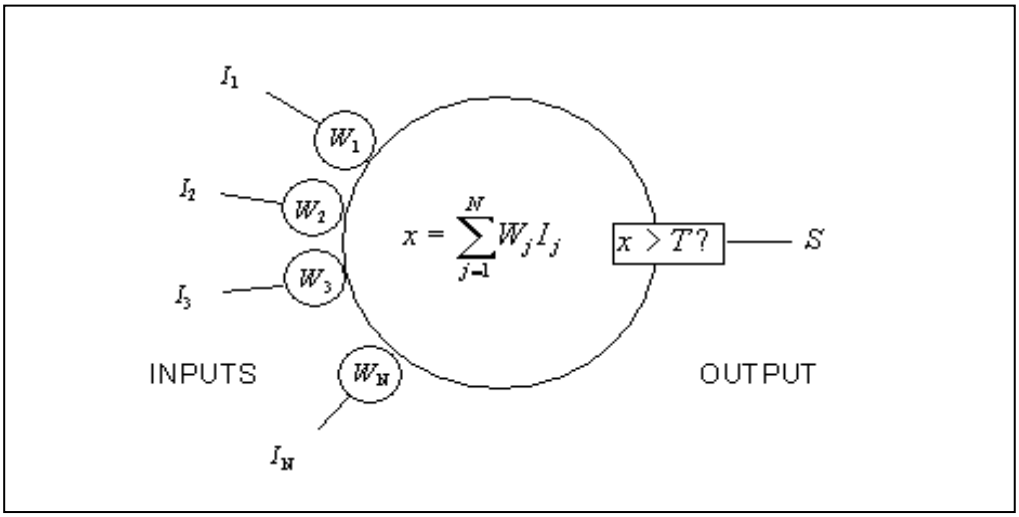


Figure 3.3 – Mathematical Representation of a Neuron

3.2 FEEDFORWARD NETWORKS

Feedforward Networks are special types of ANNs which have the following properties:

- Neurons are arranged in layers, with the first layer taking in inputs and the last layer producing outputs. The middle layers have no connection with the external world, and hence are called hidden layers.
- Each neuron in one layer is connected to every neuron on the next layer. Hence information is constantly "fed forward" from one layer to the next, and this explains why these networks are called feed-forward networks.
- There is no connection among neurons in the same layer

The basic feedforward network performs a non-linear transformation of input data in order to approximate the output data. The number of input and output nodes is determined by the nature of the modeling problem being tackled, the input

data representation and the form of the network output required. The number of hidden layer nodes is related to the complexity of the system being modeled. The interconnections within the network are such that every neuron in each layer is connected to every neuron in the adjacent layers. Each interconnection has associated with it a scalar weight which is adjusted during the training phase. The hidden layer nodes typically have sigmoid transfer functions.

3.3 BACK PROPAGATION (BP) LEARNING ALGORITHM

The BP learning process works in small iterative steps: one of the example cases is applied to the network, and the network produces some output based on the current state of its synaptic weights (initially, the output will be random). This output is compared to the known-good output, and a mean-squared error signal is calculated. The error value is then propagated backwards through the network, and small changes are made to the weights in each layer. The weight changes are calculated to reduce the error signal for the case in question. The whole process is repeated for each of the example cases, then back to the first case again, and so on. The cycle is repeated until the overall error value drops below some pre-determined threshold. At this point we say that the network has learned the problem "well enough" - the network will never exactly learn the ideal function, but rather it will asymptotically approach the ideal function.

In backpropagation learning, every time an input vector of a training sample is presented, the output vector o is compared to the desired value d.

The comparison is done by calculating the squared difference of the two:

$$Err = (d-o)^2$$

The value of Err tells us how far away we are from the desired value for a particular input. The goal of backpropagation is to minimize the sum of Err for all the training samples, so that the network behaves in the most "desirable" way.

$$\text{Minimize} \quad \Sigma \text{ Err} = (d-o)^2$$

We can express Err in terms of the input vector (i), the weight vectors (w), and the threshold function of the perceptions. Using a continuous function (instead of the step function) as the threshold function, we can express the gradient of Err with respect to the w in terms of w and i.

Given the fact that decreasing the value of w in the direction of the gradient leads to the most rapid decrease in Err, we update the weight vectors every time a sample is presented using the following formula:

$$w_{new} = w_{old} - n\,(\delta \text{ Err} / \delta \text{ w}) \qquad \text{where n is the learning rate (a small number} \sim 0.1)$$

Using this algorithm, the weight vectors are modified so that the value of Err for a particular input sample decreases a little bit every time the sample is presented. When all the samples are presented in turns for many cycles, the sum of Err gradually decreases to a minimum value, which is our goal as mentioned above.

Chapter 4

ARABIC/URDU OPTICAL CHARACTER RECOGNITION SYSTEM

The product titled Arabic/Urdu Optical Character Recognition System, as discussed earlier, forms part of the Arabic/Urdu Information Retrieval System which essentially contains the OCR and an image search engine[2]. Details pertaining to the implementation of this project are discussed below.

4.1 ARCHITECTURE

To accomplish the feat of Urdu optical character recognition the system relies heavily on the use of Artificial Neural Networks. Matlab by Mathworks and specifically the Mathworks Neural Network Toolbox was used for the purpose of implementing the ANNs. The project subsequently had to be coded in Matlab as well though a Java interface has also been developed and included to enable the easy integration of this system into Java-based systems. The system consists of four basic and separate modules. These modules are discussed in the next section. The interaction of modules with one another for the overall working of the project is shown in Figure 4.1.

[2] An Urdu image-based search engine, part of the Arabic/Urdu Information Retrieval System developed by Qambber Hussain Syed, BIT-2(B), NUST Institute of Information Technology, National University of Sciences and Technolog.

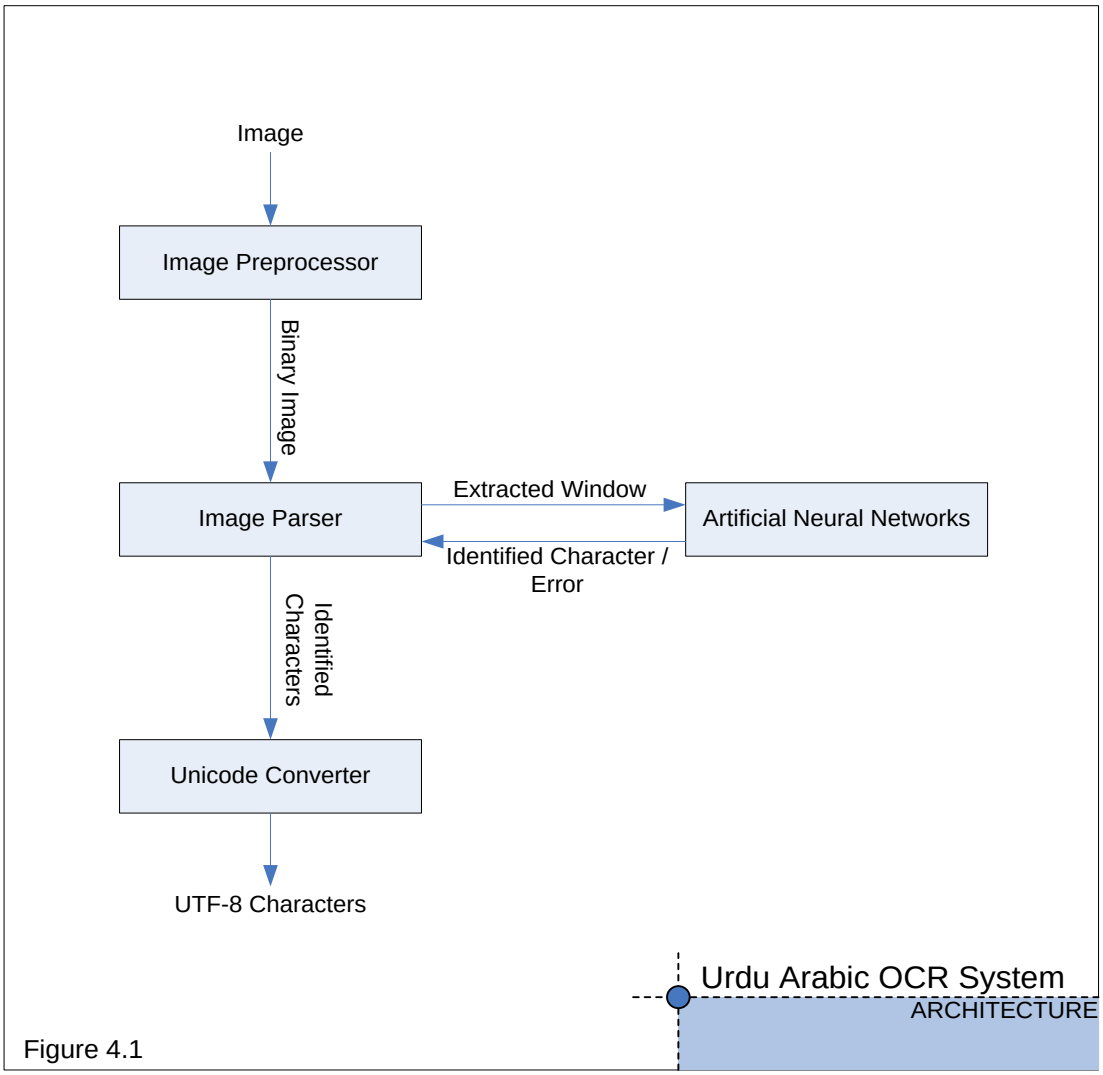


Figure 4.1 – Arabic/Urdu OCR System Architecture

4.2 MODULES

As evident in the figure 4.1, the project can be divided into four main modules which are:

- The Image Pre-Processor.
- The Image Parser.
- The Artificial Neural Networks.
- The Unicode Converter.

Details of each module is given below.

4.2.1 Image Pre-Processor

4.2.1.1 Module overview

- Input

  RGB Image

- Procedure

  The RGB is first converted to Grayscale and then divided by the threshold to reduce the image to its binary form.

- Output

  Binary image

4.2.1.2 Module description

This module is tasked with the preparation of the image for processing purposes. An image once read into the program as a multi-dimensional matrix is in RGB format, although the required form is binary (i.e. 1s and 0s only). The input image is defined in three channels (red, green and blue) with the third dimension of the multi-dimensional matrix representing the three channels. The first two dimensions represent the x and y coordinates if the pixels. The actual cells of the matrix hold the intensity of each channel. Intensity ranges from 0 to 255.

The first step of the process is to convert the image to its single-channeled gray form. This simplifies the three-dimensional image to its two-dimensional form

eliminating the channel dimension entirely. Once this step is accomplished the next step is to convert the image into its binary form, with 1 representing the text and 0 the background. This is done by defining a threshold and then dividing the image into two parts: those pixels that are higher than the threshold and others that are lower. The higher values are then assigned 0s and the lower ones 1s, thus effectively converting the image to its binary form. The threshold is chosen carefully to remove as much noise as possible while keeping damage to the text to a minimum.

4.2.2 Image Parser

4.2.2.1 Module overview

- Input

  Binary Image

- Procedure

  Image is scanned for present lines and then parsed line-wise. Parts of the image are extracted column wise for each line and sent to the ANNs for recognition after which the associated character value is appended to a data structure.

- Output

  Identified Characters

4.2.2.2 Module description

The image parser forms the core of the OCR system. This module is responsible for breaking up the image into small windows and getting these

windows identified by the neural networks. This feat is accomplished by first finding the lines of text in the image. This is done so that the image can later on be parsed line-wise ignoring the lines spaces however large they may be. The process of finding lines in an image is done through the following three steps:

Figure 4.2 – Conversion of Image to Skeletal Form

- Convert the image to its skeletal form using the Zhang-Suen thinning algorithm first developed by Zhang and Suen (1984) [1]. This is illustrated in figure 4.2.
- Parse the skeleton image row-wise.
- Select the row with maximum number of 1s in a cluster of rows. A group of rows having at least one pixel of 1 value crunched between all 0 valued pixel rows is called a cluster of rows. Lines of text in the image constitute such cluster of rows when read pixel-wise. The task is accomplished by first counting the number of ones in the image. Each row is then parsed and the row with maximum one value in a cluster is extracted and appended to the "Spine" vector (illustrated in Figure 4.3).

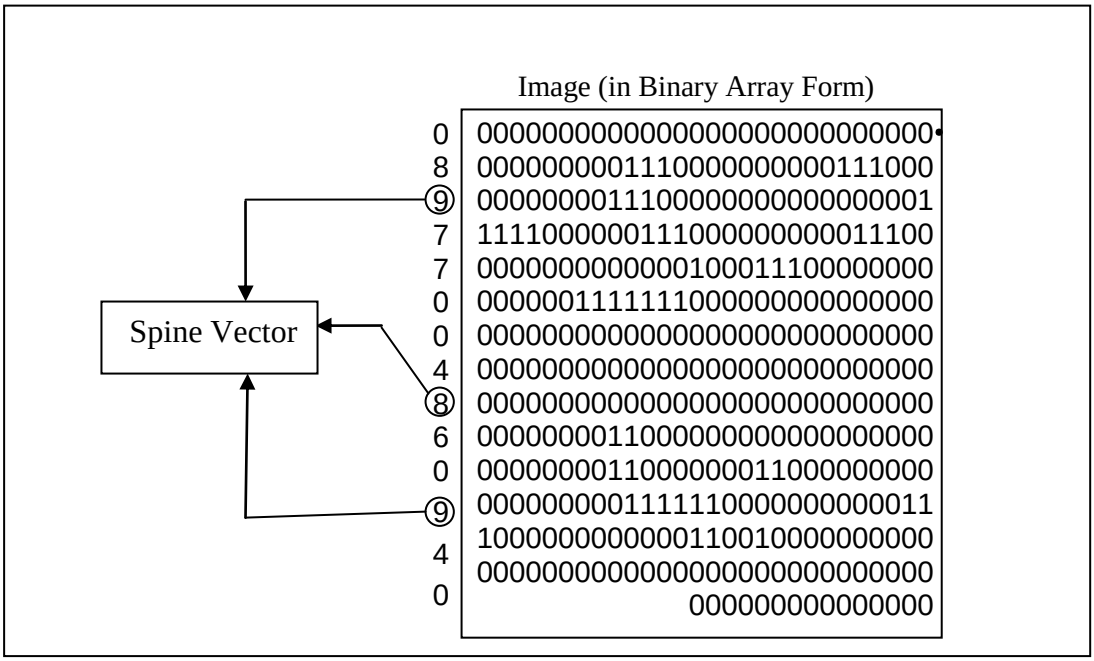


Figure 4.3 – Creation of Spine Vector

Thus for each identified spine we get an identified line of text in an image. The spine denotes the highest occurrence of "on" pixels. In the Urdu font this maps to the base of the word from which characters spring.

These lines in an image are documented with the number of the row on which they occur. Once this is done the lines are then parsed by the parser. The following steps take place (illustrated in Figure 4.4):

- Each row entry in the spine vector is then sought out in the image and parsed from right to left i.e., the Urdu way.
- Function to find the next '1' pixel moving towards the left determines where the next window will start.

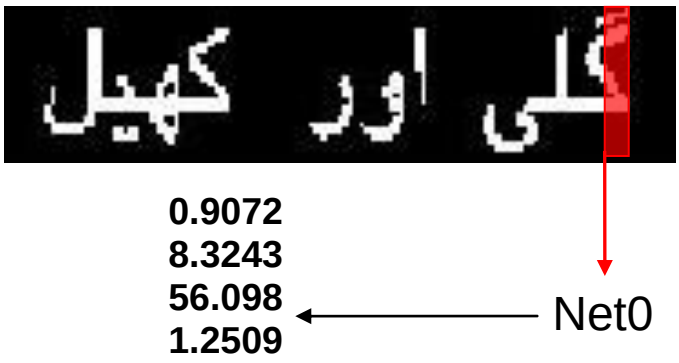
- A window is then selected starting from the smallest size and extracted from the image.
- The window is then sent for identification to the Artificial Neural Networks.
- On successful ID the associated value is appended to the character vector.
- On a failure the size of the window is increased until a match is found or incase a match is never found an error value is then appended.
- The parser then finds the next ‘on’ pixel and the whole process is reeated again.

This process continues for all lines in an image, till all the image is read and tagged.[3]

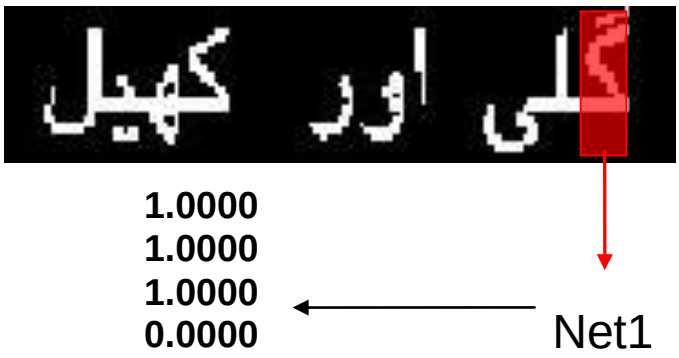
Table 4.1: Window Sizes

No.	Window Size	Total Pixels
1.	27 x 3	81
2.	27 x 5	135
3.	27 x 6	162
4.	27 x 7	189
5.	27 x 8	216
6.	27 x 10	270

[3] Consult Flowchart in Appendix B for details on the whole process



a) Window size is the smallest, Gaaf cannot be identified properly by the associated ANN and thus is rejected.



b) Window size is increased and this time Gaaf is identified by the associated neural network.

c) The parser moves to the next 'on' pixel and redeploys the extraction window starting from the smallest size and repeating the above process.

Figure 4.4 – Image Parsing

4.2.3 The Artificial Neural Networks

4.2.3.1 Module overview

- Input

  N x M pixels (where N x M is the window size)

- Procedure

  Pixels are fed as input to the appropriate ANN (i.e. the one having N x M inputs) and the output recorded.

- Output

  Binary String

4.2.3.2 Module description

The actual identification of patterns or the actual Urdu characters in this case is done by the ANNs. The neural networks take as input actual pixels of the image, process it and output a binary string corresponding to the identified alphabets. After considering several ANN models the Multilayer Feedforward network model was chosen. Such a network consists of more than one layer i.e. it is composed of one or more hidden layer of neurons. The flow of this network is also in one direction. The inputs move from input to hidden layer(s) and then to the output layer and are eventually outputted.

A total of eight ANNs were needed to solve the problem of Urdu character identification. ANNs were created and maintained in Matlab 5.3. The need for such numbers of neural networks was due to two reasons: a) the fact that for every

window size a different neural network was required, b) and keeping the number of hidden layer neurons to a minimum substantially hampers the learning ability of the neural networks. The number of hidden layer neurons was kept low so that training was possible in the absence of virtual memory in excess of 2 Gigabytes in the training environment.

Input to each neural network are the actual pixels extracted from the image. These values are fed into the neural network and an output received. Output is in the form of values equal in number to the neurons in the hidden layer of the artificial neural network in question. An output is considered valid if it corresponds to a set of accepted values. The neural networks are trained to output these values in case a certain input is encountered. All other values are considered as rubbish.

Details of each ANN used in the project is given below (the "shapes identified" below points to the different shapes an Urdu alphabet can take and not the Urdu alphabet itself – meaning that "aliph" at the end and "aliph" isolated are two different shapes).

Net 1

- Number of inputs = 81
- Neurons in hidden layer = 25
- Neurons in output layer = 4
- Shapes identified = 7

Net 2

- Number of inputs = 135
- Neurons in hidden layer = 35
- Neurons in output layer = 5
- Shapes identified = 25

Net 3

- Number of inputs = 162
- Neurons in hidden layer = 30
- Neurons in output layer = 5
- Shapes identified = 27

Net 4

- Number of inputs = 189
- Neurons in hidden layer = 30
- Neurons in output layer = 5
- Shapes identified = 22

Net 5

- Number of inputs = 216
- Neurons in hidden layer = 25
- Neurons in output layer = 5
- Shapes identified = 24

Net 6

- Number of inputs = 216
- Neurons in hidden layer = 25
- Neurons in output layer = 5
- Shapes identified = 30

Net 7

- Number of inputs = 270
- Neurons in hidden layer = 20
- Neurons in output layer = 4
- Shapes identified = 24

Net 8

- Number of inputs = 135
- Neurons in hidden layer = 35
- Neurons in output layer = 5
- Shapes identified = 18

The training for each neural network was done using the Back-Propagation Learning Algorithm (as discussed earlier). Training was also done using the "train" function in Matlab 5.3. For this purpose training sets were constructed for each neural network. Training set for a typical ANN consisted of four to five inputs of ideal shape and a similar number for its noisy version. This was done for each

shape that was to be identified by the neural net. For example, for neural network 2 the training set consisted of about 200 inputs (25 shapes with 8 inputs for each). This gives an idea of how complicated and time consuming the training process was. Given more time, the neural networks would have been able to accomplish an ever higher level of efficiency.

4.2.4 Unicode Converter

4.2.4.1 Module overview

- Input

  Identified Characters

- Procedure

  Matches characters to their Unicode values and outputs a Unicode string.

- Output

  Unicode String

4.2.4.2 Module description

The final module of the system is tasked with the conversion of identified characters to UTF-8 Urdu characters. The previous module is responsible for the identification of Urdu characters in an image and storing them. These identified characters are not yet in a form in which they may prove useful. To make them

useful the Unicode converter has the simple yet important task of replacing these identified characters with their corresponding UTF-8 values. The Unicode character set is read from permanent storage and imported into the program from where it is used to map the identified characters to their UTF values. A chart of the standard UTF mappings has been included in Appendix A. The output of this module and consequently of the whole program are the identified characters in Unicode format.

At this point it is important to mention that the UTF values of characters are loaded in the form of an array saved on the hard disk. This array holds the UTF values of all characters pertaining to one font-type. To use a different font or a different set of UTF values for output the user has only to replace the UTF character array with another.

4.3 TECHNOLOGIES

- Mathworks Matlab 5.3.
- Matlab Neural Network Toolkit.
- Java (for possible integration with mother-project).

Chapter 5

RESULTS

The completed system demonstrates reasonable amount of accuracy when identifying Urdu characters from images. The final product was tested in a variety of situations ranging from ideal images to images with a high intensity of noise. A summary of these tests is shown in figure 5.1.

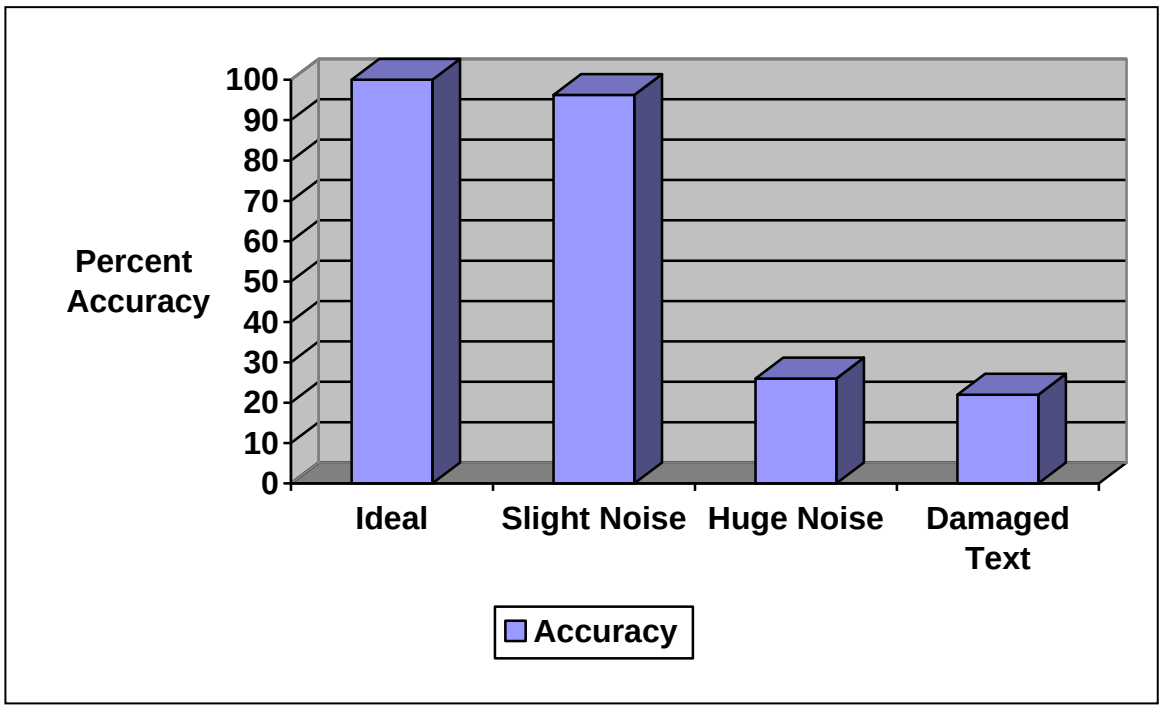


Figure 5.1 – Accuracy of the OCR System

As evident in the graph, for each environment a separate accuracy was observed. The details of the test sets and the results are given in table 5.1.

Table 5.1: Testing Details

Test Type	Total Samples	Errors	Accuracy
Ideal Clean Image	515	0	100 %
Slight Noise (Normally Occurring Image)	636	24	96.22%
Huge Noise	300	213	29 %
Damaged Text	200	148	26 %

It was observed that the product displayed higher accuracy when the level of noise in the image was at a minimum to moderate level. The level of noise in this case conforms to what most sites have to offer. Thus we can safely assume that this is the accuracy that can be expected when deploying the OCR system on the Internet. But in case of high level of distortion (self-added noise with a high threshold) to the image and in cases where the actual text was damaged the level of accuracy dropped considerably. Damaged text implies text in the image which is broken, has missing pixels and/or rough edges. This changes the actual form on the font in question thus deceiving the artificial neural networks. The relationship between the accuracy of the system to the amount of damage to the image (noise) is shown in figure 5.2.

As evident in Figure 5.2 the accuracy falls considerably as we move from moderate noise to higher levels of noise. This was due to the limited amount of

training that could be given due to time limitations. This problem can be rectified by inclusion of more test subjects to the training sets of the associated ANNs but sadly this will only increase the accuracy to a certain level, after which the ANNs' learning ability will start to decline.

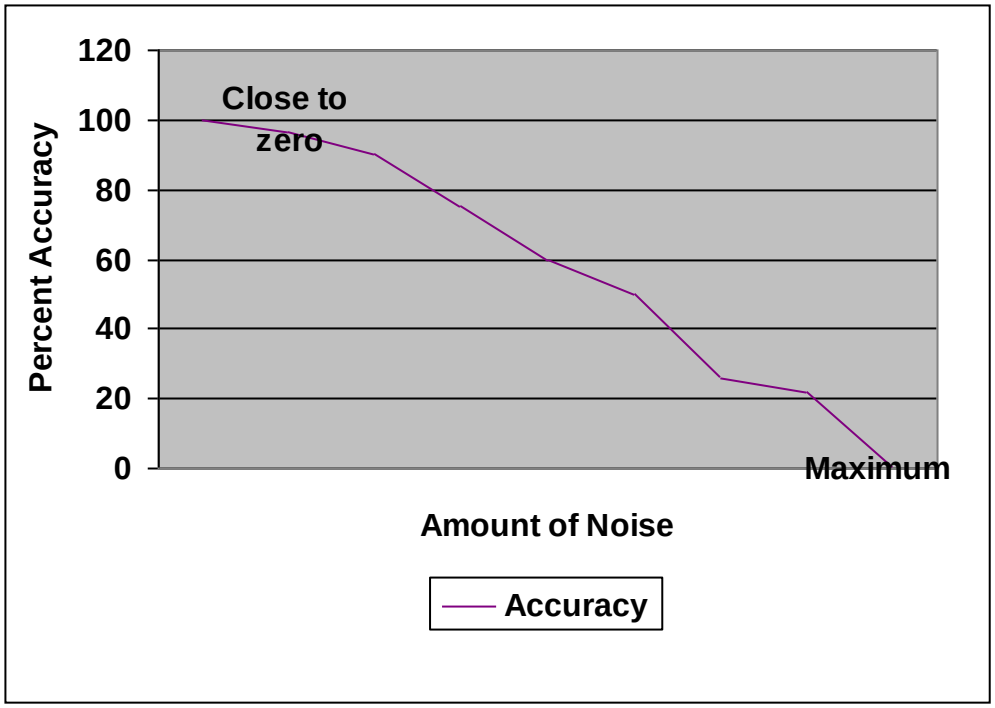


Figure 5.2 - Relationship b/w Accuracy & noise

Chapter 6

CONCLUSION

Using Artificial Neural Networks in the domain of image processing and Optical Character Recognition (OCR) is a tried and tested approach. But in case of Urdu this approach has also been put to the test due to the language's unique and difficult-to-read writing styles. This project provides an approach to overcoming this problem and has been reasonably effective in doing so. In doing this the project paves the way for the promotion of Urdu on the net.

The vast resource of the Internet is filled with Urdu based sites and to be able to search them as well will bring the Urdu speaking community one step closer to experiencing the World Wide Web in their own cherished and loved language. This will hopefully bring the Internet to people who are not literate in the English language too, thus providing the fulcrum to reduce illiteracy in Pakistan.

Chapter 7

RECOMMENDATIONS

The accuracy of the OCR system can be increased further by training it with a larger variety of training sets. This is a very time consuming process though and will work only to a point, after which the learning ability of the ANN will diminish considerably. This is because of the small number of neurons in the hidden layer. Alternately the number of neurons in this layer can also be increased, although that is only possible in an environment with an excess of 2 Gigabytes of virtual memory as the training process of such a neural net can easily consume much more than that. Training must be incremental thus forcing the system to increase in efficiency while keeping the effects of all previous trainings intact.

A larger number of fonts can be incorporated into the system by adding new neural networks to the system trained to identify the new font type and modifying the parser code to recognize the new ANNs.

This system can also be extended to identify and classify hand-written Urdu text as well. Currently the product focuses on machine-generated text. This should not be considered as boundary of the project. Extending it to include hand-written fonts as well will increase the utility of the project and pave the way for incorporating this product as part of a scanning system. The system will then be capable of scanning hard-material and extracting Urdu text from it by using the

OCR. This is useful in reading text from hard forms, letters, hand-written documents etc.

Another extension worthy of noting will be the incorporation of a spelling checking system. The system can be used to further enhance the accuracy of the OCR system by identifying and subsequently dealing with all types of problem areas in an image. Thus the system will be able to deal with images that contain a normal amount of errors as well, thereby increasing its accuracy in retrieving text from images.

REFERENCES

[1] Zhang T.Y. and Suen C.Y., in: Gonzales and Woods R.E., *Digital Image Processing*, Addison Wesley, 1993.

[2] Elgammal and M. A. Ismail, "*A Graph-Based Segmentation and Feature extraction Framework for Arabic Text Recognition*", Sixth International Conference on Document Analysis and Recognition (ICDAR 01), Seattle, Washington, U.S.A. September 10-13, 2001.

[3] S. J. Perantonis and P. J. G. Lisboa, "Translation, rotation, and scale invariant pattern recognition by high-order neural networks and moment classifiers," *IEEE Transaction Neural Networks*, vol. 3, no. 2, pp. 241-251, March 1992.

[4] H. K. Sardana, M. F. Daemi and M. K. Ibrahim," Global description of edge patterns using moments," *Patten Recognition*, vol. 27, no. 1, pp. 109-118, 1994.

[5] A. K. Jain, *Fundamentals of digital image processing*, Prentice-Hall, Eaglewood Cliffs, New Jersey, 1989.

[6] I. Sekita, T. Kurita and N. Otsu,"Complex autoregressive model for shape recognition," *IEEE Transaction Pattern Analysis and Machine Intelligence*, vol. 14, no.4, pp. 489-496, April 1992.

[7] C. T. Zahn and R. Z. Roskies," Fourier descriptors for plane closed curves," . . *IEEE Transaction Computers,* vol. c-21, no. 3, pp. 269-281, March 1972.

[8] S.A. Mahmoud," Arabic character recognition using Fourier descriptors and character contour encoding," *Pattern Recognition*, vol. 27, no. 6, pp. 815-824, 1994.

[9] T. Taxt, J. B. Olafsdottir and M. Daehlen," Recognition of handwritten symbols," *Pattern Recognition,* vol. 23, no. 11, pp.115-116, 1990.

[10] T. S. Shpeherd, W. Uttal, S. Dayanand and R. Lovell," A method for shift, rotation and scale invariant pattern recognition using the from and arrangement of pattern-specific features," *Pattern Recognition,* vol. 25, no. 4, pp. 343-356, 1992.

[11] H. Nishida and S. Mori," Algebraic description of curve structure," *IEEE.*192 *Transaction Pattern Analysis and Machine Intelligence*, vol. 14, no. 5, pp. 516-533, May 1992.

[12] H. Rom and G. Medioni," Hierarchical decomposition and axial shape description," *IEEE Transaction Pattern Analysis and Machine Intelligence*, vol. 15, no.16, pp.973-981, October 1993.

[13] L. Gupta, M. R. Sayeh and R. Tammana, "A neural network approach to . . . . . robust shape classification," *Pattern Recognition*, vol. 23, no.6, pp. 563-568, 1990.

[14] G. Cortellazzo, G. A. Mian, G. Vezzi and P. Zamperoni, "Trademark shapes description by string-matching techniques," *Pattern Recognition*, vol. 27, no. 8, pp. 1005-1018, 1994.

[15] J. A. Starzyk and S. Chai," Vector contour representation for object recognition in neural networks," *IEEE International Conference on Systems, Man and Cybernetics*, vol., pp. 399-404, 1992...93.

[16] L. Gupta and M. D. Smith," Invariant planer shape recognition using dynamic alignment," *Pattern Recognition*, vol. 21, no.3, pp. 235-239, 1988.

[17] G. N. Bebis and G. M. Papadourakis," Object recognition using invariant object boundary representations and neural network models," *Pattern Recognition*, vol. 25, no. 1, pp. 25-44, 1992.

[18] I. Dinstein, G. M. Landau and G. Guy," Parallel (PRAM EREW) algorithms for contour-based 2D shape recognition," *Pattern Recognition*, vol. 24, no. 10, pp. 929-942, 1991.

[19] Y. Lin, J. Dou and H. Wang," Contour shape description based in an arch . . height function," *Pattern Recognition*, vol. 25, no.1, pp. 17-23, 1992.

[20] M. H. Hassoun, *Fundamentals of Artificial Neural Networks,* Prentice-Hall of India, Third Edition.

[22] S. Haykin, *Neural Networks: A Comprehensive Foundation,* Pearson Education, Second Edition.

[23] P. McCollum, *An Introduction to Back-Propagation Neural Networks*, Seattle Robotics Society.

[24] www.Nationmaster.com/Encyclopedia/Urdu

APPENDICES

Appendix A

Commonly used Arabic/Urdu Character Codes

Serial	Symbol	Unicode	Description
1		0200	Space
2	آ	0627	Alif Maddah
3	ا	0622	Alif
4	ب	0628	Beh
5	پ	067E	Peh
6	ت	062A	Taa
7	ٹ	0679	Taa with small Tah
9	ث	062B	Tha
10	ج	062C	Jeem
11	چ	0686	Tcheh
12	ح	062D	Haa
13	خ	062E	Kha
14	د	062F	Dal
15	ڈ	0688	Dal with small Tah
16	ذ	0630	Thal
17	ر	0631	Ra
18	ڑ	0691	Ra with small Tah
19	ز	0632	Zain
20	ژ	0698	Jeh
21	س	0633	Seen
22	ش	0634	Sheen
23	ص	0635	Sad
24	ض	0636	Dad
25	ط	0637	Tah
26	ظ	0638	Dhah
27	ع	0639	Ain

Serial	Symbol	Unicode	Description
28	غ	063A	Ghain
30	ق	0642	Qaf
31	ک	06A9	Caf
32	گ	06AF	Gaf
33	ل	0644	Laam
34	م	0645	Meem
35	ن	0646	Noon Ghunna
36	ں	06BA	Noon
37	و	0648	Waw
38	ؤ	0624	Waw with Hamza
39	ہ	0647	Ha
40	ھ	06BE	Do Chashmi Ha
41	ی	0649	Choti Yeh
42	ے	06D2	Bari Yeh
43	ء	0621	Hamza

Appendix B

Flowchart: Image Parser