Handwritten Hindi Digits Recognition of Courtesy Amounts in Arabian bank checks Using Neural Networks

By

Remah Wajih Naji Al-Khatib

Supervised By

Prof. Nidal Shilbayeh

Master Thesis

Submitted in Partial Fulfillment of the Requirements for the Master Degree In Computer Science

Department of Computer Science
Faculty of Information Technology
Middle East University
Amman – Jordan

Jan, 2011

Middle East University

Authorization Statement

I, Remah Wajih Naji Al-Khatib, authorize Middle East University to supply hardcopies and electronic copies of my thesis to libraries, establishments, or bodies and institutions concerned with research and scientific studies upon request, according to the university regulations.

Name: Remah Wajih Al-Khatib

Date: 1/16/2011

Signature:

Middle East University

Examination Committee Decision

This is to certify that the thesis entitled "Handwritten Hindi Digits Recognition of Courtesy Amounts in Arabian bank checks using Neural Networks" was successfully defended and approved in Jan / 2011.

Examination Committee Members	**Signature**
Prof. Nidal F. Shilbayeh Department of Computer Science Middle East University	Nidal Shilbayeh
Prof. Mohammad M. AL-Haj Hassan Department of Computer Science Middle East University	M. M. H. H.
Dr. Kamal R. Al-Rawi Faculty of Information Technology University Of Petra	

Acknowledgements

"In the name of Allah the Most Gracious the Most merciful". My guidance cannot come except from Allah, in Him I trust, to Him I repent, and to Him praise and thanks always go. I would like also to thank Prof. Nidal Shilbayeh for his valuable contributions, corrections, and discussions. Gratitude is also due to the Faculty of the Information Technology Department for their valuable information that was rendered to us during the course of our studies.

Dedication

I dedicate this thesis to my parents, husband, and my family for their full support, for their great patience, and for their great attention.

Table of Contents

List of tables

List of Figures

List of Abbreviations

A2iA	Artificial Intelligent Image Analysis
ANN	Artificial Neural Networks
CENPARM	Centre for Pattern Recognition and Machine Intelligence.
CNN	Convolution Neural Networks
DPI	Dot per Inch
FCC	Freeman Chain Code
GUI	Graphical User Interface
HMM	Hidden Markov Model
Logsig	Logistic Sigmoid
Matlab	Matrix Laboratory
MICR	Magnetic Ink Character Recognition
MLP	Multi Layer Perceptron
MLP-DDD	Multi Layer Perceptron – Directional Distance Distribution
MSE	Mean Square Error
OCR	Optical Character Recognition
PDA	Personal Digital Assistant
RBF	Radial-Basis Function Networks
SVM	Support Vector Machines

Abstract

The problem of processing bank checks automatically started to emerge in banks as the need for a system that processes bank checks and recognizes handwritten fields increased. In this thesis, a system is designed for recognizing the courtesy amount in Arabian bank checks. This system is divided into two main parts; the first part was for the offline handwritten Hindi digits recognition system that recognizes separated Hindi digits using the Multilayer Perceptron Neural Networks with backpropagation training algorithm. The samples were written by hand, 1000 of them for training and 600 for testing. The samples were scanned and preprocessed before entering the Neural Network for recognition. The system achieved a recognition rate of 74.4%. The second part is for the courtesy amount recognition system. The courtesy amount during this system passes through the segmentation and feature extraction stages. The segmentation process divides the string of digits into separated digits and the feature extraction process extracts the digits so that they get ready to enter the recognition system in the first part. Both stages performed properly and successfully.

Key words: Offline Hindi Digit Recognition, Courtesy Amount Recognition, Bank Checks Recognition, Courtesy Amount Segmentation, Feature Extraction, Neural Networks, Backpropagation Algorithm.

الملخص

ان معالجة شيكات البنوك اتوماتيكيا من احدى المشاكل التي بدأت بالظهور في البنوك بسبب زيادة الحاجة الى نظام يعالج هذه الشيكات ويتعرف على الكتابة المكتوبة بخط اليد. لقد تم بناء نظام فعال للتعرف على المبلغ الاجمالي في شيكات البنوك العربية. هذا النظام مقسم الى قسمين رئيسين. القسم الاول عبارة عن نظام للتعرف على الارقام الهندية المكتوبة بخيط اليد والمدخلة للجهاز عن طريق الماسح الضوئي(السكانر) و سيتم التعرف على الارقام عن طريق الشبكات العصبونية التي تستخدم خوارزمية ال backpropagation . لقد تم جمع العينات لهذا النظام و كانت 1000 عينة لتدريب النظام و 600 عينة لاختباره. لقد تم فحص النظام و تم الحصول على دقة بمعدل 74.4%. اما القسم الثاني فهو للتعرف على المبلغ الاجمالي في شيكات البنوك العربية. يمر المبلغ الاجمالي في هذا النظام بمرحلتين هما مرحلة التقسيم و مرحلة استخراج الخصائص. في مرحلة التقسيم يتم تقسيم المبلغ الاجمالي المكون من مجموعة ارقام الى ارقام منفصلة وفي مرحلة استخراج الخصائص يتم استخراج الارقام كل رقم على حدة. لقد تم تقسيم و استخراج الارقام عن طريق المرحلتين السابقتين بنجاح. بعد هاتين المرحلتين تدخل الارقام المنفصلة والمستخرجة في النظام الذي صمم للتعرف على الارقام الهندية في القسم الاول للتعرف على الارقام.

Chapter 1

Introduction

1.1 Introduction

Handwritten Arabian bank checks recognition is the main title of this thesis. But before talking about recognizing bank checks, Handwritten Hindi digit recognition should be first mentioned. Since, it is the basis for recognizing Arabian bank checks.

Handwritten digits and character recognition is becoming the focal point for researchers recently. It is the ability of a computer to recognize handwritten digits and characters as a clear and precise input. Handwritten recognition is divided into two sections: Online Handwritten Recognition and Offline Handwritten Recognition. Online means using pens in PDAs or Mouse in computers for writing texts or digits, then special sensors will pick up mouse or pen movements and save the image. However, offline means to have texts, documents, digits, and others written by hand and then entered to the computer by any acquisition machine such as scanners or digital cameras. Then they will be saved as images into the computer.

Handwritten digits and character recognition are important for all languages. Researchers have worked through many languages such as English, Spanish, French, and many others. But the work has been limited for some languages such as Arabic, Farsi, and others. Pan, Bui, & Suen, (2009) were interested in the recognition of Farsi numerals which are similar to Arabic ones. In this thesis the main concern here is for

Arabic language and recognizing its numerals which are called the Hindi digits and shown in table1.1.

٠	١	٢	٣	٤	٥	٦	٧	٨	٩

Table 1.1 Hindi Digits from Zero to Nine

There are many applications of handwritten digit recognition such as bank checks, Postal codes, and car plates. The recognition of bank checks also interested many researchers lately; however, the research has been limited to Arabian bank checks recognition. The recognition of bank checks branches into many areas depending on the different fields that are found in bank checks. The bank check consists of many fields to be recognized such as name, date, legal amount (the amount written in words), courtesy amount (the amount written in numbers), and the signature. An image of Arabic bank check is shown in figure 1.1. As shown that the empty fields that need to be filled by hand are the ones that need to be recognized. Researchers choose the fields that they are interested in, some of them were interested in legal amount recognition, others in courtesy amount recognition, or in both.

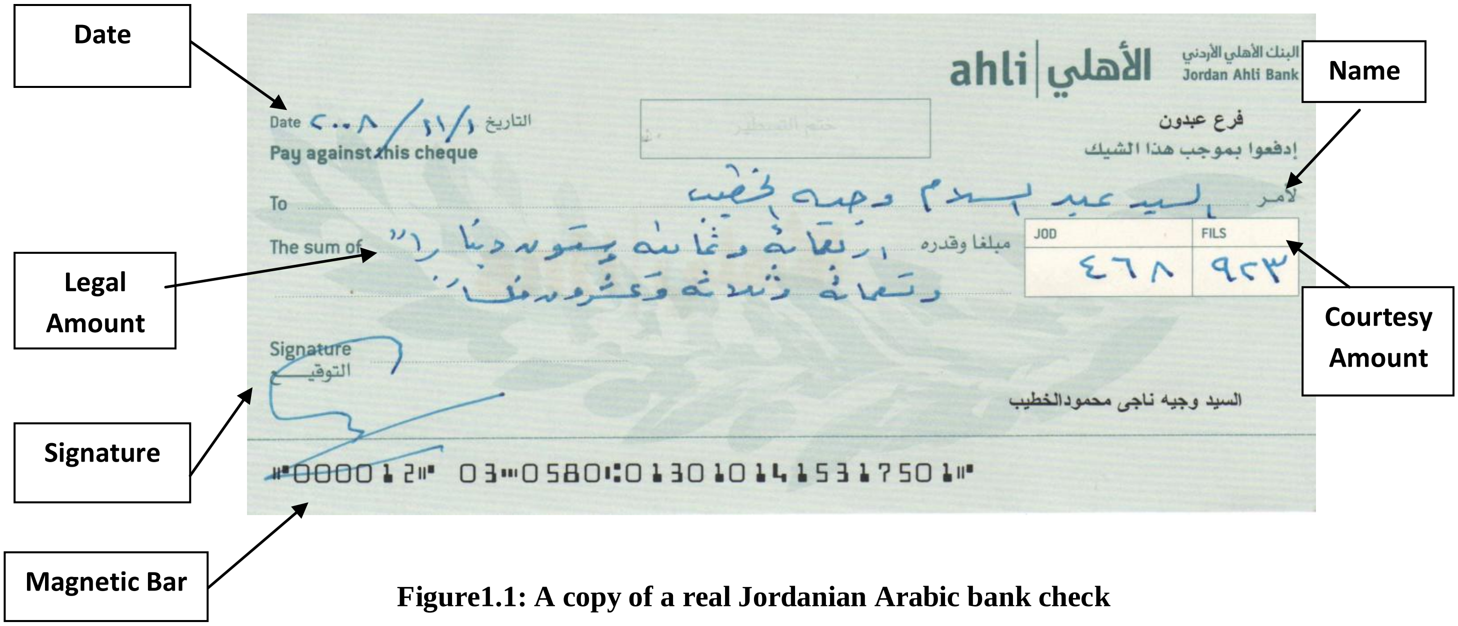


Figure1.1: A copy of a real Jordanian Arabic bank check

Bank checks are processed in banks by recognizing the Magnetic bar that is found at the end of the check. This magnetic bar is a string of digital numbers that is recognized by the Optical Character Recognition (OCR) machine in banks, since this string is written using Magnetic Ink Character Recognition (MICR). This string contains the account number and the bank code which can be processed automatically (Palacios, Gupta, & Wang, 2003). The main interest in this thesis is in recognizing the courtesy amount (the amount written in digits). Since when checks are processed in the bank, the employee at the bank looks at the numerical field and ignores other fields (Palacios, & Gupta, 2002).

The designed system will be able to recognize offline handwritten Hindi digits as the first part of the system; it will also be able to recognize the handwritten Hindi digit courtesy amount in Arabian bank checks. The system will use segmentation and feature extraction algorithms to segment and extract the courtesy amount from the check. The segmentation and feature extraction methods will be applied to different types of handwritten digits and characters, and perform the purpose of segmentation and extraction efficiently and successfully.

1.2 Statement of problem

Since most of bank checks need to be partially processed by hand, there is a significant interest in the banking industry for new approaches that can be used to read paper checks automatically. The Handwritten Courtesy Amount Recognition System should be able to recognize the courtesy amount in Arabian bank checks. This requires the system to recognize at the beginning the Arabic numerals (Hindi digits). Also to be able to recognize the check, the system should recognize the courtesy amount after performing many operations on it such as: detection, segmentation, and recognition, using different techniques.

1.3 Objectives of the Study

The following are the main objectives of this thesis:

- Designing a main system that reads the courtesy amount in Arabian bank checks.
- Designing and implementing an offline handwritten Hindi digits recognition system that recognizes separate offline Hindi digits.
- Designing and implementing a system that is able to recognize a handwritten Hindi digit written in Arabian bank checks.
- Implementing an effective segmentation method that is used to segment a string of digits or a string of characters into separate ones.

- Implementing a successful feature extraction method that is used to extract the digits in a string into separate extracted digits and save each digit as a new image.

1.4 Significance of the Study

- Offline handwritten recognition has a great significant benefit to man machine communication. In many applications such as mail sorting and checks reading.

- The system should provide a great benefit to Arabic Banks to help them process not only check amounts, but also other payment documents which consist of Arabic numerical amounts.

- It can assist in the automatic processing of handwritten documents, books, old cursives. It helps these handwritten treasures to survive from getting lost, or disappearing.
- Recognition of Arabic handwritten presents an importance to Arabic speaking countries. To provide the facilities to process Arabic word documents, Arabian bank checks, Arabic letters, large volumes of Arabic handwritten old books and so on.

1.5 Limitation of the Study

- The courtesy amount location must be constrained at a specific location, as most of the banks differ in the location of the courtesy amount.

- The following problem of segmentation may be excluded when two characters are connected with each other.

1.6 Thesis Organization

In addition to this chapter, the thesis includes four other chapters: Chapter 2 provides an overview about the offline handwritten recognition and the bank checks recognition fields. It also lists the related work to these two fields. Chapter 3 describes the designed system in details with respect to its different stages, modules, and Algorithms. Chapter 4 describes the experimental results of the designed system. Finally, chapter 5 contains the discussion of the experimental results; it also contains the conclusion, future work suggestions, and some recommendations.

Chapter 2

Literature Review

2.1 Conceptual Framework

In this section an overview will be introduced about the state of the art that is related to the study in this thesis. There are six sections; each section will be an overview to the related stage in the designed system in this thesis:

2.1.1 Offline Handwritten Hindi Digit Recognition

Handwritten digits and character recognition recently have been an interesting domain for researchers who were interested in recognizing characters and digits in many languages. Handwritten character recognition is divided into two parts; online character recognition and offline character recognition. The online handwritten recognition involves the automatic conversion of text as it is written on a special digitizer or PDA, where a sensor picks up the pen-tip movements as well as pen-up/ pen-down switching. Online handwritten uses not only pens in PDAs, but also mouse gesture in personal computers and fingers in some cases of touch screens. The offline handwritten recognition is more different than the online since the handwritten will be written using a pen and a paper. The texts and digits written by hand will be entered into the computers using many acquisition devices such as, scanners, digital cameras, and others. In offline, the recognition will be much harder than in online since the images are taken by acquisition devices which cause some effects on images like noises, different sizes, different colors, and others. These effects require a much harder preprocessing stage to deal with in the case of offline handwritten recognition. In this study the recognition will be on recognizing Hindi digits, where the researches around Hindi digits are limited especially in the case of offline handwritten Hindi digits. In the state of art many works have been introduced in the field of handwritten recognition. In Shilbayeh, Raho, & Alkhateeb, (2009) a recognizer was designed to recognize Hindi digits that were written by a mouse. Another work was introduced by Sadri, Suen, & Bui, (2003) in which they presented a method for recognizing isolated handwritten Arabic/Persian digits. This method is based on Support Vector Machines (SVMs). Also Naser, (2010) presented a system that recognizes online handwritten Hindi digits written by a mouse. There are many works for recognizing digits for different languages such as English, Bengali, Brazilian, French and many others, using many different classifiers such as Multi Layer Perceptrons (MLPs), Support Vector machines (SVMs), Hidden Markov Model (HMM), and many others.

2.1.2 Check Recognition

Check recognition is one application from many others for handwritten recognition. It's been an interesting field in researches. Many researchers worked through this field in many countries in different languages. Many researches in the following countries were done for bank checks recognition: Bangladesh, Brazil, France, Italy, and many English speaking countries. The work was limited to Arabic speaking countries especially in recognizing the courtesy amount field. Recognizing the legal amount in Arabic bank check was introduced by (Farah, et al., 2004). Recognition of the courtesy amount in Arabian bank checks is our interest in this thesis. Many researches introduced the courtesy amount recognition and were mentioned in the related work.

2.1.3 Preprocessing

Preprocessing in computer science means to have a program that processes a set of input data to produce an output data that is used as an input to another program. The preprocessing stage is a very important one. After the process of data collection and gathering images is done, the scanned image may contain noise and other undesired objects. Usually filters are used for this purpose, also in case of characters and digits morphological operations are used to make the character or the digit more precise in case of having cuts in lines, or deleted features. The preprocessing stage makes it easy for the pattern recognition system since the noise is removed instead of leaving everything to the recognition system to deal with (Sivanandam, Sumathi & Deepa, 2006). Palacios & Gupta, (2002), used in the preprocessing stage the following techniques (Slant Correction, size normalization, thickness normalization). Many stages will be used in the preprocessing stage in the case of having offline handwritten characters or digits. Since the images will be coming from the scanner, the images will contain too many problems. These problems will affect the recognition rate negatively. These problems are: noise, size, color, writing styles and others. The preprocessing stage is important because it fixes all the previous problems and makes the images ready to other stages.

2.1.4 Feature Extraction

Feature Extraction in pattern recognition and in image processing is a special form of dimensionality reduction. It is a part of image processing, where a part of the whole image is taken to recognize its specific pattern and then it is measured to classify the image on the basis of that particular measurement. This stage is a process of extracting the objects from images. In the case of handwritten extraction, it is the process of extracting the handwritten area from an image. The handwritten area of characters or digits should be the foreground of an image which should be extracted from the background. Extraction is an important stage that precedes the recognition stage. After digits and characters are extracted, they will be ready for recognition. There are many ways used in the state of the art for the extraction stage. In Haboubi & Maddouri, (2005), they presented a hybrid method of extraction of handwritten areas of Tunisian bank checks based on image processing and numerical pattern recognition. In Khan, (1998), in order to extract each component form the check, it is necessary to find a white pixel and then use a search algorithm to find all of the pixels which are connected to it. Once all of those pixels are recognized, they are grouped together into one connected component and then extracted. In Palacios & Gupta, (2002), in their system for extraction they determined the group of characters by minimum bounding rectangles (MBRs). The MBRs are the smallest rectangles that contain all the pixels of one component, and are described in terms of two pairs of coordinates. In short, the extraction is used to extract objects from images using many techniques. The extraction process is extracting the white pixels or the one’s pixels from the image.

2.1.5 Segmentation

Segmentation is the process of partitioning a digital image into multiple segments which are sets of pixels. In the system of recognizing the courtesy amount in Arabian bank checks, segmentation is the second stage after determining the location of the courtesy amount. Segmentation is considered in the state of the art the most challenging part in recognizing a string of digits or characters. It could be more challenging since there could be connected digits where the segmentation part becomes more difficult.

Image segmentation is the process of partitioning the digital image into multiple regions that can be associated with the properties of one or more criterion. These properties are gray level, color, texture shape, and others. In mathematical sense the segmentation of an image which is a set of pixels, is the partition of the image into n disjoint sets R_1, R_2, …, R_n, called segments or regions such that the union of all regions equals the image.

Image = R_1 U R_2 U … U R_n.

There are many types of segmentation. The following are some of these types:

1. Column Segmentation: based on dividing the image according to its (X,Y) coordinates.
2. Thresholding: based on converting the image to black and white, where the background is black and the foreground is white or vice versa.
3. Region based: based on the regions of the image.
4. Edge based: based on the boundary of each object in the image.
5. Hough Transform: concerned with the identification of lines, positions of arbitrary shapes, most commonly circles or ellipses in an image.
6. Watershed approach: marks the foreground and background objects. Then applies a watershed segmentation function.

In the state of art for the case of segmenting the courtesy amount or even the legal amount, many types and techniques of segmentation were used. The following will be a number of examples of the segmentation techniques used in the state of the art. One example, in Parisi, Claudio, Lucarelli & Orlandi, (1998), they segmented the car plate numbers and characters by finding white areas between columns with higher density of black pixels. Isolated black pixels are wiped out and the characters are segmented. Another example, in Palacios, Sinha & Gupta, (2002) their approach for segmentation is based on a recursive function that uses splitting algorithms to divide blocks into isolated digits. The system begins the segmentation process by making a few obvious separations of characters, the primitives obtained are pre-classified as digit, fragment, multiple, or delimiters. Then the fragments are merged with digits or other fragments and analyzed again. In Khan, (1998), is another example for segmentation, where he used many algorithms for segmentation in order to segment digits apart from each others in the courtesy amount. The algorithms discussed are: Min-Max, Hit and Deflect, Drop Fall, and Structural Feature based segmentation. He also handled overlapping and connected characters. Also in Sivanandam, Sumathi & Deepa, (2006) they mentioned the segmentation algorithms, bottom up and top down for efficient segmentation of characters. The bottom up algorithm is very simple and very fast. It builds words out of their characters. The top down algorithm attempts to recognize the attributes present within words. It operates on a higher level of abstraction, extracting features from words rather than partitioning the word and attempting to recognize each part of it. In addition to previous examples, in Palacios, Gupta & Wang, (2003) used the feedback strategy for segmentation they also used splitting algorithms in order to divide multiple segments into pairs of digits. In the recognition stage they used Neural Networks technique which is the multi layer perceptron (MLP).

2.1.6 Neural Network Classifiers

The classification of handwritten character recognition depends on the types of classifiers which are Linear Classifiers, Quadratic Classifiers, K-Nearest Neighbor Classifiers and the Neural Network Classifiers according to (Chakravarty & Day, 200?). The last classification is based on the use of Neural Networks. Since handwritten character recognition is a complex process, it can't be accomplished by a simple neural network. It needs a multilayer feed forward networks, which consists of hidden layers between the input and the output layers. Palacios, Gupta, (2002) used for character recognition the multilayer perceptron. The following is a briefing of Neural Networks:

- **Artificial Neural Networks**: Sivanandam, Sumathi & Deepa (2006)

Artificial Neural Networks is an information processing system. In this system the elements, called neurons, process the information. The signals are transmitted by means of connection links. The links posses associated weights, which are multiplied along with the incoming signal for any typical neural net. The output signal is obtained by applying activations to the net input.

- **Neural Network:** consists of a large number of simple processing elements called neurons.

- **Neurons:** processing elements that are connected to each other by directed communication links, which are associated with weights.

- **Weight:** is information used by the neural net to solve a problem.

- **Activation function:** is used to normalize the output response of a neuron. The sum of the weighed input signal is applied with an activation to obtain the response. There are many types of activation functions: sigmoid function, threshold function, etc.

- **Network Architecture:** it is the arrangement of neurons into layers and patterns of connection within and in-between layers. There are various types of network architectures: Feed forward, feedback, fully interconnected net, competitive net, etc.

2.1.6.1 Training the Network

It is the process of modifying the weights in the connections between network layers with the objective of achieving the expected output. The method of setting the value for the weights enables the process of learning or training the network. The internal process that takes place when a network is trained is called learning. There are three types of training:

- **Supervised Training:** it is the process of providing the network with a series of sample inputs and comparing the output with the expected responses. In a neural net, for a sequence of training input vectors there may exist target output vectors. The weights then can be adjusted according to a learning algorithm.

- **Unsupervised Training:** In a neural net, if the training input vectors are known, the target output is not known. The net may modify the weight so that the most similar input vector is assigned to the same output unit. These networks are very complex and difficult to implement since they involve looping connections back into feedback layers and iterating through the process until some sort of stable recall can be achieved. They are also called self-learning networks or self organizing networks because of their ability to carry-out self learning. They are used in self- organizing feature maps, adaptive resonance theory (ART), etc.

- **Reinforcement Training:** In this method the teacher is assumed to be present, but the right answer is not present to the network. Instead the network is only presented with an indication of whether the output answer is right or wrong. Usually it's better to avoid this type of training and use the previous ones because they are more direct and their analytical basis is well understood.

2.1.6.2 Backpropagation Algorithm in FeedForword Neural Networks:

Backpropagation is a gradient decent algorithm where the network weights are moved along the negative of the gradient of the performance function. The definition backpropagation refers to the manner where the gradient is computed for nonlinear multilayer networks. FeedForword Networks often have one or more hidden layers of sigmoid neurons followed by an output layer of linear neurons. Multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear relationships between input and output vectors (Demuth, Beale & Hagan, 2007).

2.1.6.3 Training and Testing the FeedForword Network:

Once the network weights and biases are initialized, the network is ready for training. The network can be trained for function approximation, pattern association, or pattern classification. The training process requires a set of examples of proper network behavior. During training the weights and biases of the network are adjusted to minimize the network performance function (net.performFcn). The default performance function for feedforward networks is Mean Square Error (MSE) which is the average squared error between the network outputs and the target outputs. The MSE formula is described in equation 2.1.

$$F = mse = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \qquad \text{.. } \mathbf{2.1}$$

There are several training algorithms for feedforword network. All these algorithms use the gradient of the performance function to determine how to adjust the weights to minimize performance function which means minimizing the mean square error. The gradient is determined using a technique called backpropagation, which involves performing computation backward through the network. The backpropagation is derived using the chain rule of calculus which is described in (Jacobs, 2008). In the end the network will be tested using some samples to test the efficiency of the network (Demuth, Beale & Hagan, 2007).

2.2 Related Work

Handwritten recognition in general has been an attractive field for researchers for many years; the researchers have been interested in this field for recognizing many different languages in many different countries. In specific the applications of handwritten recognition also attract researchers in this field. These applications are bank check recognition, postal code recognition, word document recognition and many others. Bank check is an attractive application that interested many researchers from different countries, bank checks have been recognized for many countries such as, Bengal, Brazil, China, France, Italy, Japan, Korea, Tunisia, the United States and many others. The following literature survey shows the researchers who searched in this field.

Anisimov, et al., (1999) designed a system that recognizes legal and courtesy amount in French bank checks. They introduced the subject of the check reader A2iA that performs the same goal of their system.

Gorski, et al., (2000) described the A2iA check reader. A2iA stands for Artificial Intelligence and Image Analysis. It is a commercial bank check recognition system that recognizes all types of payment documents including the bank check in both English and French speaking countries. Their aim is to replace the human operator that reads bank checks and other payment document with the A2iA.

Dilecce, et al., (2000) presented a system that recognizes the legal amounts in Italian bank checks. They used the hybrid approach in their recognition. They used analytical techniques to segment the legal amount into a sequence of ‘core’ groups of words and a global technique to recognize these groups.

Freitas, et al., (2000) presented a system that recognizes the legal amount in Brazilian bank checks. The system consists of many stages the preprocessing stage where slant correction and smoothing phases are used, feature extraction and segmentation stage, the recognition stage. They used Hidden Markov Models for the recognition process.

Palacios & Gupta, (2002) described a system that reads Brazilian and US check banks by first reading the scanned image of the bank check to read the check during four stages which are: detecting the courtesy amount block within the image, segmentation of string into characters, recognizing characters, and post processing stage for insuring correct recognition.

Palacios, Gupta, & Wang, (2003) designed a system for reading the amount field in bank checks. Their system architecture consists of four stages which are: segmentation of the string into individual digits, normalization, recognition of each character using a neural network classifier, and syntactic verification. They used many algorithms and Neural Network techniques to read the courtesy amount of checks in many countries and other handwritten documents.

Sadri, Suen, & Bui, (2003) presented a method for recognition of isolated handwritten Arabic/Persian digits. This method is based on Support Vector Machines (SVMs), and a new approach of feature extraction. A CENPARMI Indian (Arabic/Persian) handwritten digit database is used for training and testing of SVM classifiers. This database provides 7390 samples for training and 3035 samples for testing from the real life samples. The recognition rate achieved using Support Vector Machines is 94.14%.

Farah, et al., (2004) discussed recognizing literal amount in Arabian bank checks. Their approach for recognition is performed by a multi-classifier system. This paper is written for publication as a chapter in the Artificial Intelligence: Methodology, Systems, and Applications.

Haboubi & Maddouri, (2005) proposed in their paper a hybrid method to extract handwritten areas in bank checks. This method is based on digit recognition by Fourier descriptors and different steps of colored image processing. It requires the bank recognition of its code which is located in the check marking band as well as the handwritten color recognition by the method of difference of histograms. The areas of extraction are then carried out by the use of some mathematical morphology tools.

Shilbayeh, & Iskandarani, (2005) presented a comprehensive mouse gesture system, which is designed and tested successfully. The system is based on UNIPEN algorithm in terms of mouse movements and applies its geometrical principles such as angles and transposition steps. The system incorporates Neural Networks as its learning and recognition engine.

Khan, & Hossain, (2005) described a system that recognizes Bengali bank checks using the automatic extraction of the user-entered data. Their system first extracts the user-entered data, the fields that are filled by hand. These fields are extracted using many techniques. Then the extracted sub images are recognized using Neural Network.

Morshed, et al., (2005) proposed an efficient system for recognizing Handwritten Bangla Numerals and its application in recognizing Bangladesh postal code in mail sorting systems. They used in their system Neural Network with one hidden layer, and applied the back propagation algorithm for clarification.

Lorigo, & Govindaraju, (2006) focused on handwritten Arabic characters recognition. They discussed the recognition methods used to recognize handwritten Arabic characters. It was the first survey to give testing procedures and recognition rates for as many systems as possible that focused on handwritten Arabic recognition.

Palacios, & Gupta, (2008) described another system that is more developed than the two systems developed in (2002, 2003) respectively. This system includes three modules to fully and automatically process bank check. The first module is used for detecting the strings in the image, the second one used to segment and recognize strings using feedback loops, the third one used for post processing issued and for insuring high accuracy of recognition. In their paper they do not focus on the result obtained by the system but they focus on the benefits of each module mentioned above.

Shilbayeh & Iskandarani, (2008) designed a system based on cross pruning (CP) algorithm to optimize number of hidden neurons and number of iterations in Multi Layer Perceptron (MLP) neural based recognition system. Their approach is to study the effect of varying the size if the network hidden layers (pruning) and number of iterations (epochs) on the classification and performance of the used MLP.

Karthikeyan, (2008) built a recognition model for online handwritten characters in Tamil and Devanagari languages. He solved the problem of online handwritten characters by converting it to offline handwritten characters. He used two stages in his project the first one is the stroke recognition and the other one is the stroke to character conversion. The stroke recognition was used to recognize Tamil language using Neural Network, and the other stage for recognizing Devanagari language.

Omari, et al., (2009) developed a system that recognizes handwritten Arabic numbers using Neural Networks. Their system as they mentioned is supposed to be used for many applications such as check recognition, postal code recognition, and car plates. Their system is built upon the following basic stages: Image acquisition, preprocessing which consists of (Noise removal, Normalization, Thinning and Skeletonization), Segmentation, Feature extraction, and Classification and Recognition.

Pan, Bui, & Suen, (2009) developed a method for recognizing Farsi Handwritten Numerals. This method uses the sparse and over-complete structure within the data. They used the Convolution Neural Networks (CNN) for this purpose.

Mahmoud & Awaida, (2009) described a technique for automatic recognition of off-line handwritten Arabic (Indian) numerals using Support Vector Machines (SVM) and Hidden Markov Models (HMM).

Shilbayeh, Raho & Alkhateeb, (2009) proposed a system that recognizes Hindi Digits using Freeman Chain Code (FCC) with the eight connectivity method. Their system is tested on a sample of 1350 digits written by 27 different writers selected from different ages, genders and jobs, each one wrote 10 digits 5 times. An experimental result shows high accuracy of about 89.5% on the sample test.

Naser, (2010) in her thesis proposed an efficient system for recognizing Hindi digits drawn by the mouse. The system was designed using Neural Networks classifiers. A recognition rate of 91% of accuracy was achieved.

Chapter 3

Handwritten Hindi digit Courtesy Amount (HHCA) Recognition System

3.1 Introduction

The system is designed to recognize the handwritten Hindi digit courtesy amount in Arabian bank checks. The system starts from cropping the location of the courtesy amount and ends up with recognizing the string of handwritten Hindi Digits. The HHCA recognition system is divided into two subsystems. The first subsystem is for recognizing individual offline handwritten Hindi digits, and the second subsystem is for recognizing Courtesy Amount written in Hindi digits.

The first subsystem is divided into three main stages: preprocessing, feature extraction, and digit recognition. The second subsystem consists of five main stages: preprocessing, cropping the location of the courtesy amount, segmentation of the string into individual digits, feature extraction, and digits recognition stage. Last stage is done by referring to the first subsystem that deals with the recognition of the individual Hindi digits. The HHCA recognition system is designed and implemented using Matlab Image Processing Tool Box and Neural Networks Tool Box.

3.2 Offline Handwritten Hindi Digit Recognition System:

The offline handwritten Hindi digit recognition system architecture is shown in figure 3.1 and follows a description in details for each stage.

Figure 3.1: Block Diagram for Handwritten Hindi Digits Recognition System

3.2.1 Preprocessing

The preprocessing stage concerns itself with processing input data to produce output data that is used as input to another stage. The importance of the preprocessing stage lies in preparing the digit to be in its final shape before entering the recognition stage. The images were first scanned using a scanner with a 300 dpi. These scanned images were colored images; the size of the samples was not defined and differs from one image to another since the digits were written in different handwriting styles from different people of different ages. There are many steps in the preprocessing stage which are shown in the figure 3.2:

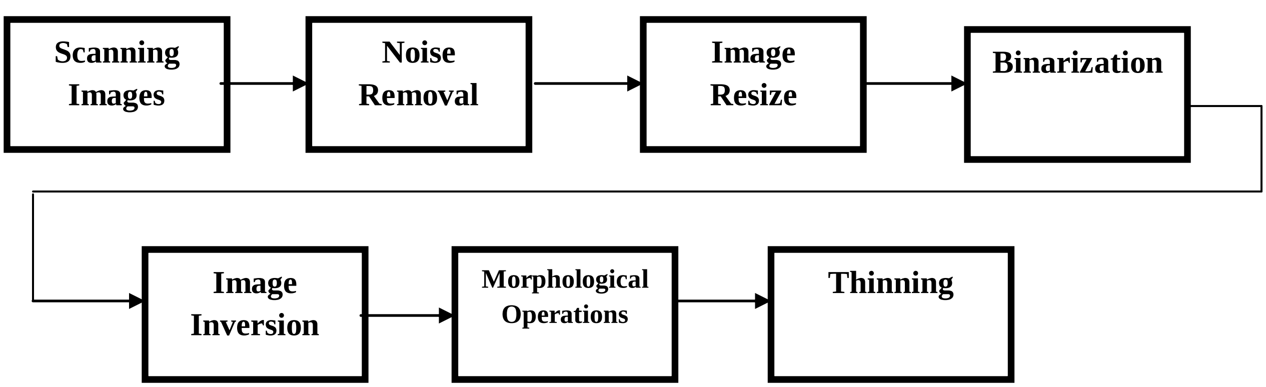


Figure 3.2: Block Diagram for Hindi digits Preprocessing Stage

- **Scanning Images**: using a scanner with 300 dpi, the resulting image is colored, very large, and has some noise.

- **Noise Removal:** Removing the noise is done by applying the median filter. The noise coming from the scanner couldn't be removed totally by the median filter. Therefore, the noise was removed by adjusting the contrast and sometimes the brightness using the windows picture manager. The figure 3.3 shows an image before and after noise removal.

Image Before Noise Removal   Image After Noise removal

Figure 3.3: a sample before and after noise removal step

- **Image Resize:** Resizing the image using down sampling and interpolation bilinear methods. Interpolation method was used to ensure that the image will not lose too many pixels because losing too many pixels, results in losing some features of the original shape of the image. The image is resized in its final size to [180 180] it was the best size to show the image without losing too many pixels.

- **Binarization**: it is the process of converting the image into black and white. The word binarization is coming from binary. To convert the image into binary image means to make all pixels zeros and ones. The technique used to convert the image to black and white is **(level = graythresh(image);)** this function from the Image

Processing tool box in matlab is used to compute a global threshold (level) that can be used to convert an intensity image to a binary image. **(level)** is a normalized intensity value that lies in the range [0, 1]. The graythresh function uses Otsu's method, which exhaustively search for the threshold that minimizes the intra-class variance, defined as a weighted sum of variances of the two classes. The following description for the Otsu's Method and Algorithm that were used for binarization:

Otsu's Method: Otsu, (1979), Otsu's Method expressed in equations 3.1 & 3.2

$$\sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t) \quad \text{.......................................} \ \mathbf{3.1}$$

Weights ω_i are the probabilities of the two classes separated by a threshold t and σ_i^2 variances of these classes.

Otsu shows that minimizing the intra-class variance is the same as maximizing inter-class variance:

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_1(t)\omega_2(t)\left[\mu_1(t) - \mu_2(t)\right]^2 \quad \text{.....................} \ \mathbf{3.2}$$

Which is expressed in terms of class probabilities ω_i and class means μ_i which in turn can be updated iteratively. This idea yields an effective algorithm.

Otsu's Algorithm: Otsu, (1979)

A. Compute histogram and probabilities of each intensity level

B. Set up initial $\omega_i(0)$ and $\mu_i(0)$

C. Step through all possible thresholds $t = 1 \ldots$ maximum intensity

   a. Update ω_i and μ_i

   b. Compute $\sigma_b^2(t)$

D. Desired threshold corresponds to the maximum $\sigma_b^2(t)$.

- **Image Inversion:** image inversion means inversing the pixels of an image the zeros pixels to ones and vice versa. Inversing the Image is necessary since after applying the binarization, the background of the image takes the ones pixels and the foreground of the image takes the zeros pixels. As known in Image Processing using matlab, they deal with image as ones pixels. Therefore, inversion is necessary to give the objects or digits in the image the ones and give the background the zeros. In this stage the inverse function is applied to the image for this purpose.

- **Morphological Operations:** Applying some morphological operations on the image in order to enhance the image and remove distortions from the image. A set of closing and dilating with a structuring element of one pixel is applied on images. Closing operation is used to close the holes in the image, while dilate is used to thicken the image and connect broken lines.

- **Thinning stage**: that uses the thinning algorithm to bring the digit into one pixel thick.

Thinning Algorithm: Sivanandam, Sumathi & Deepa, (2006) The thinning algorithm in matlab is applied using the 'thin' function; the following syntax for using the thinning algorithm. BW = bwmorph (bw, ‘thin’, inf). This syntax means that the image will be thinned until infinity. Infinity means that the image will have its skeleton shape with one pixel thick. The following is the algorithm described in steps:

1. Divide the image into two distinct subfields in a checkerboard pattern.
2. In the first sub iteration, delete pixel **p** from the first subfield if and only if the conditions $\mathbf{G}_1$, $\mathbf{G}_2$, and $\mathbf{G}_3$ are all satisfied.
3. In the second sub iteration, delete pixel **p** from the second subfield if and only if the conditions $\mathbf{G}_1$, $\mathbf{G}_2$, and $\mathbf{G}_3'$ are all satisfied.

Condition G1:

$$X_H(p) = 1$$

Where **………………………….. 3.3**

$$X_H(p) = \sum_{i=1}^{4} b_i$$

$$b_i = \begin{cases} 1 \text{ if } x_{2i-1} = 0 \text{ and } (x_{2i} = 1 \text{ or } x_{2i+1} = 1) \\ 0 \text{ otherwise} \end{cases}$$ **……....…………3.4**

The values $\mathbf{x}_1$, $\mathbf{x}_2$... $\mathbf{x}_8$ in equations 3.3 & 3.4 are the values of the eight neighbors of **p**, starting with the east neighbor and numbered in counter-clockwise order.

Condition G2:

$$2 \le \min\{n_1(p), n_2(p)\} \le 3$$

Where **……………………………….3.5**

$$n_1(p) = \sum_{k=1}^{4} x_{2k-1} \vee x_{2k}$$

$$n_2(p) = \sum_{k=1}^{4} x_{2k} \vee x_{2k+1}$$

Condition G3:

$$(x_2 \vee x_3 \vee \bar{x}_8) \wedge x_1 = 0$$ **……………………………….3.6**

Condition G3':

$$(x_6 \vee x_7 \vee \bar{x}_4) \wedge x_5 = 0$$ **……..…………………………..3.7**

In this algorithm after applying all equations the user will get the skeleton shape of an image since the iterations are repeated until the image stops changing.

After applying all the steps of the preprocessing stage on the image the image will have its final shape that will be as an input to the recognition and classification stages. Figure 3.4 shows an image of the digit four after passing all preprocessing stages.

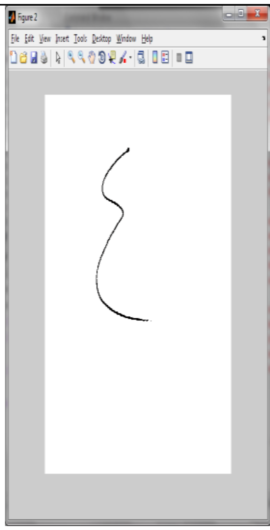
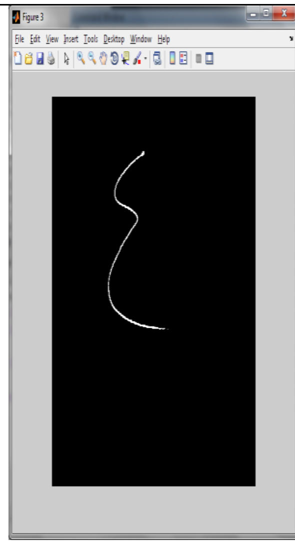
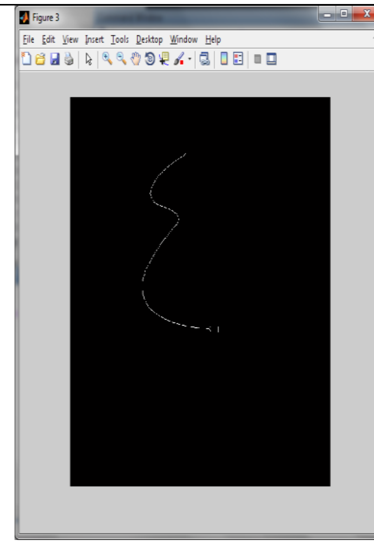
Binarization	Inversion	Dilating and Closing	Thinning
			

Figure 3.4: the digit four after passing preprocessing stages

3.2.2 Feature Extraction

This stage is very important in the recognition stage. The digit is extracted depending on its blob analysis. Blob means the group of pixels that make up the digit. To get its analysis means to measure its properties such as area, center, perimeter, diameter, intensity, and many others. In feature extraction it determines the size of the input vector that enters the neural network. Instead of taking the whole image, features of the image will be extracted depending on its properties. These features implement the input vector to the neural network stage. In this system the image of size 180*180 pixels is divided into 9 blocks, each block of size 60*60 pixels.

The following three functions have been used to extract the features from each block:

- The first function is the region props function: this function depends on the properties of the image. The properties taken for each region are the area, center, perimeter, diameter, and the bounding box. The function first labels each region in the image. Then it calculates the properties of each region. The main property is taken into account here is the bounding box which is the smallest box containing the region. After the bounding box is taken, the features extracted will be of different sizes since each digit has different bounding box containing it depending on its geometrical shape.
- The standard deviation and average functions. These functions are calculated to normalize the features extracted by the region props function mainly the features extracted using the bounding box.

The features extracted from the region props function will be a vector of different sizes because the properties of each image differ from image to another. Therefore, the average and standard deviation are taken to normalize the size of the vector for all digits or images. The vector will be of size 1*18 for all images. This vector will be the input to the neural network for recognition stage.

3.2.3 Digit Recognition

In this stage a Multilayer Perceptron NN with backpropagation architecture was designed for digit classification and recognition. The used architecture has been chosen since it’s the most widely used in handwritten recognition fields. The rest of this section is a description of the Neural Network architecture in details, training the network, and testing it.

1. **Neural Network Architecture:**

The designed NN architecture consists of an Input layer, one hidden layer, and an output layer. The performance function used for measuring the network's performance is the MSE (Mean Square Error) which is the average squared error between the network outputs and the target outputs. The MSE formula is described in equation 2.1.

.The activation function used for both hidden and output layers is the log-sigmoid transfer function. This network architecture was chosen after a deep study of designing many other networks with different numbers of hidden layers, different numbers of neurons, and different types of activation functions. After training those different networks, it appeared that the network that was chosen is the best among others in performance and minimum error rate **(Naser, 2010)**. The training algorithm used in this architecture is the Gradient Descent Backpropagation with adaptive learning rate. The network is designed and trained to recognize the Offline Hindi Digits from Zero to Nine. Figure 3.5 shows the architecture of the designed network.

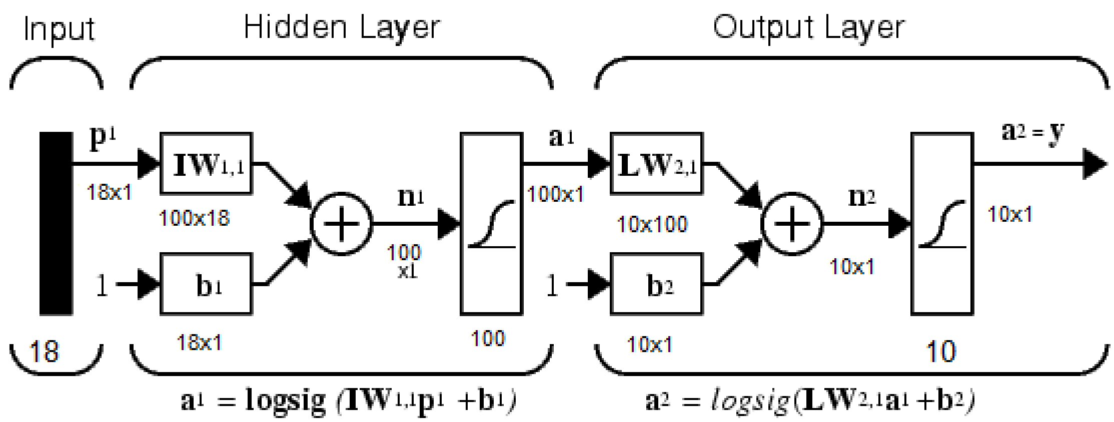


Figure 3.5: The Neural Network Architecture

The neural network needs 18 inputs in the input layer and 10 neurons in its output layer to identify the Hindi Digits. The network is a two-layer log-sigmoid/log-sigmoid network. The log-sigmoid transfer function was picked because its output range (0 to 1) is perfect for learning to output Boolean values. The log sigmoid transfer function calculates a layer's output from its net input. Figure 3.6 shows the log-sigmoid graph and symbol.

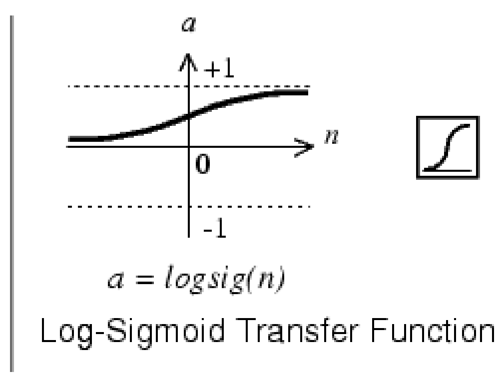


Figure 3.6: Log-Sigmoid Graph and Symbol:

Activation Function Algorithm:

Logsig (n) = 1 / (1 + exp (-n)) **...3.8**

The network consists of three layers:

- Input Layer: The Input Layer consists of 18 Boolean values as 18-element input vector. These 18 element input vectors come from dividing the original image of size 180*180 into 3*3 blocks. Each block is of 60*60 pixels, and for each block the properties are calculated for each region and then average and standard deviation are calculated to normalize the features extracted by the region props function as mentioned in the previous section. When calculating both the average and the standard deviation for all 9 blocks, we get two (the average and the standard deviation) multiplied by Nine (the total number of blocks in the image) the result is 18 element input vector.

  Nine (total number of blocks in the image) * Two (average and standard deviation for normalizing features) = 18 (input vector).

- Hidden Layer: the hidden layer has 100 neurons. This number was chosen after trying many networks with different numbers of hidden neurons and by finding the minimum mean square error (MSE). In most cases choosing the number of hidden neurons depends on guesswork and experience. It also depends on getting the least error rate after trying many networks with different number of hidden neurons.

- Output Layer: the output layer consists of 10 neurons to identify the 10 digits. The network is trained to output a one (1) in the correct position of the output vector and to fill the rest of the output vector with zeros (0s).

The Hindi digits will pass through the three layered NN architecture for recognition as shown in the following scenario in figure 3.7. The input sample is number three in Hindi numerals, it will be entered to the input layer as an 18 vector element, and then it will be transmitted to the hidden and output layers to give a correct and recognized output digit.

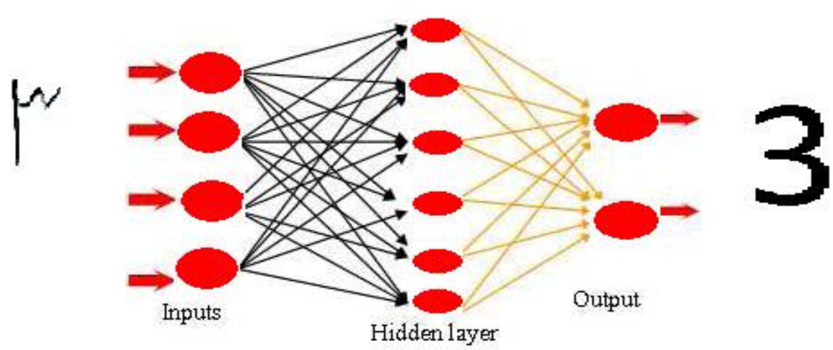
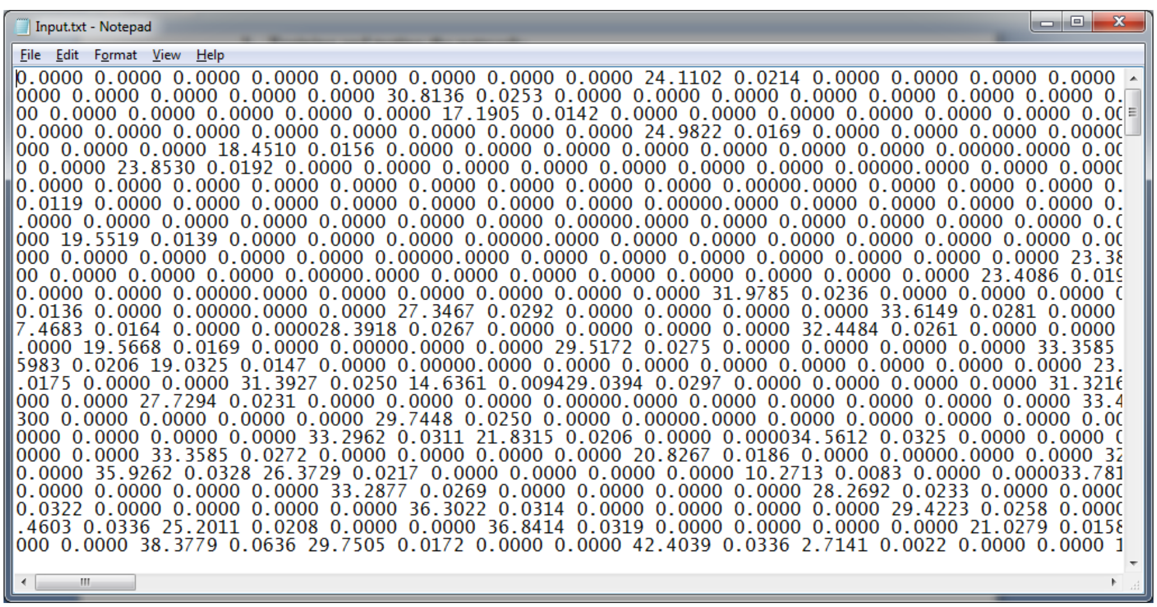


Figure 3.7: this figure shows the recognition of handwritten characters using NN

2. **Training and testing the network:**

The Network has been trained and tested using samples of handwritten Hindi digits. The training and testing stages were achieved after the samples passed through the preprocessing and the feature extraction stages. The samples were stored as vectors. Each sample or image was stored as an 18 element vector; these vectors were stored into a text document. This text document was created to be the input set for training the network. The created input file contains the features for each digit. As shown in figure 3.8 the images were stored into a text document as vectors. These vectors were the result of the feature extraction function and they will be the input for the NN to recognize each digit.


Input.txt - Notepad
File Edit Format View Help

```
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 24.1102 0.0214 0.0000 0.0000 0.0000 0.0000
0000 0.0000 0.0000 0.0000 0.0000 30.8136 0.0253 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.
00 0.0000 0.0000 0.0000 0.0000 0.0000 17.1905 0.0142 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 24.9822 0.0169 0.0000 0.0000 0.0000 0.0000
000 0.0000 0.0000 18.4510 0.0156 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.00000.0000 0.0
0 0.0000 23.8530 0.0192 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.00000.0000 0.0000 0.000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.00000.0000 0.0000 0.0000 0.0000 0.
0.0119 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.00000.0000 0.0000 0.0000 0.0000 0.0000 0.
.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.00000.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.
000 19.5519 0.0139 0.0000 0.0000 0.0000 0.00000.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0
000 0.0000 0.0000 0.0000 0.0000 0.00000.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 23.3
00 0.0000 0.0000 0.0000 0.00000.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 23.4086 0.01
0.0000 0.0000 0.00000.0000 0.0000 0.0000 0.0000 0.0000 0.0000 31.9785 0.0236 0.0000 0.0000 0.0000
0.0136 0.0000 0.00000.0000 0.0000 27.3467 0.0292 0.0000 0.0000 0.0000 0.0000 33.6149 0.0281 0.0000
7.4683 0.0164 0.0000 0.000028.3918 0.0267 0.0000 0.0000 0.0000 0.0000 32.4484 0.0261 0.0000 0.0000
.0000 19.5668 0.0169 0.0000 0.00000.0000 0.0000 29.5172 0.0275 0.0000 0.0000 0.0000 0.0000 33.3585
5983 0.0206 19.0325 0.0147 0.0000 0.00000.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 23.
.0175 0.0000 0.0000 31.3927 0.0250 14.6361 0.009429.0394 0.0297 0.0000 0.0000 0.0000 0.0000 31.321
000 0.0000 27.7294 0.0231 0.0000 0.0000 0.0000 0.00000.0000 0.0000 0.0000 0.0000 0.0000 0.0000 33.4
300 0.0000 0.0000 0.0000 0.0000 29.7448 0.0250 0.0000 0.00000.0000 0.0000 0.0000 0.0000 0.0000 0.0
0000 0.0000 0.0000 0.0000 33.2962 0.0311 21.8315 0.0206 0.0000 0.000034.5612 0.0325 0.0000 0.0000
0000 0.0000 33.3585 0.0272 0.0000 0.0000 0.0000 0.0000 20.8267 0.0186 0.0000 0.00000.0000 0.0000 3
0.0000 35.9262 0.0328 26.3729 0.0217 0.0000 0.0000 0.0000 0.0000 10.2713 0.0083 0.0000 0.000033.781
0.0000 0.0000 0.0000 0.0000 33.2877 0.0269 0.0000 0.0000 0.0000 0.0000 28.2692 0.0233 0.0000 0.000
0.0322 0.0000 0.0000 0.0000 0.0000 36.3022 0.0314 0.0000 0.0000 0.0000 0.0000 29.4223 0.0258 0.000
.4603 0.0336 25.2011 0.0208 0.0000 0.0000 36.8414 0.0319 0.0000 0.0000 0.0000 0.0000 21.0279 0.015
000 0.0000 38.3779 0.0636 29.7505 0.0172 0.0000 0.0000 42.4039 0.0336 2.7141 0.0022 0.0000 0.0000 1
```

Figure 3.8: this figure shows the input file entered to the NN for training

A description of the samples used for training and testing is explained in details in chapter 4. Also the training and testing procedures and results are mentioned in details in chapter 4.

3.3 Courtesy Amount Recognition System:

The courtesy amount recognition system architecture is shown in figure 3.9 and follows a description for each stage in details:

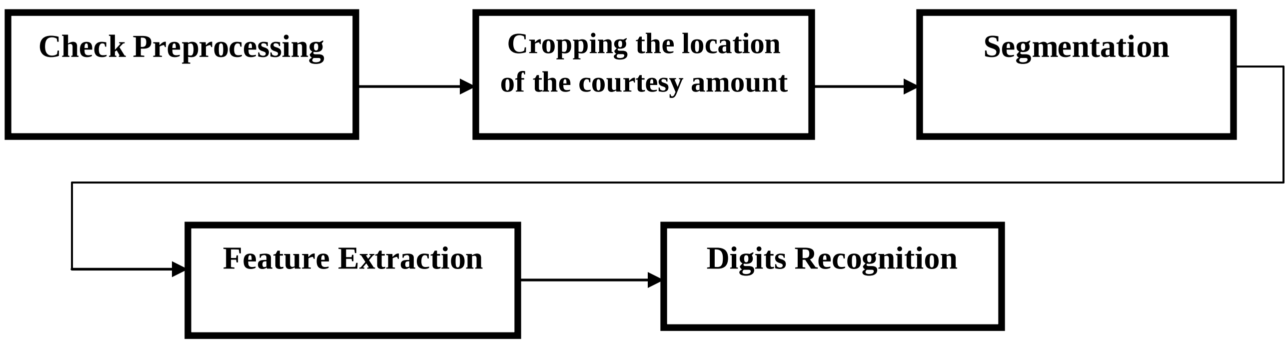


Figure 3.9: Block Diagram for Courtesy Amount Recognition System

3.3.1 Check Preprocessing

In this stage the check passes through many steps shown in figure 3.10 these steps are scanning, removing noise, Binarization, and Image Inversion.

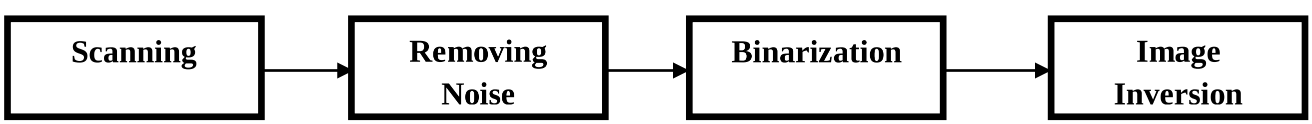


Figure 3.10: Block Diagram for Check preprocessing stage

1. **Scanning**: The check is scanned using a scanner with 300 dpi as shown in figure 3.11.

Figure 3.11: image of the original check taken by a scanner

2. **Removing Noise**: for noise removal the median filter is applied to remove noises such as salt and pepper noise. The median filter is taken after the image is converted from colored to gray as shown in figure 3.12. Then the image contrast is enhanced using histogram equalization. Histogram equalization is used in image processing tool box and it enhances the contrast of images by transforming the values in an intensity image, or the values in the colormap of an indexed image, so that the histogram of the output image approximately matches a specified histogram.

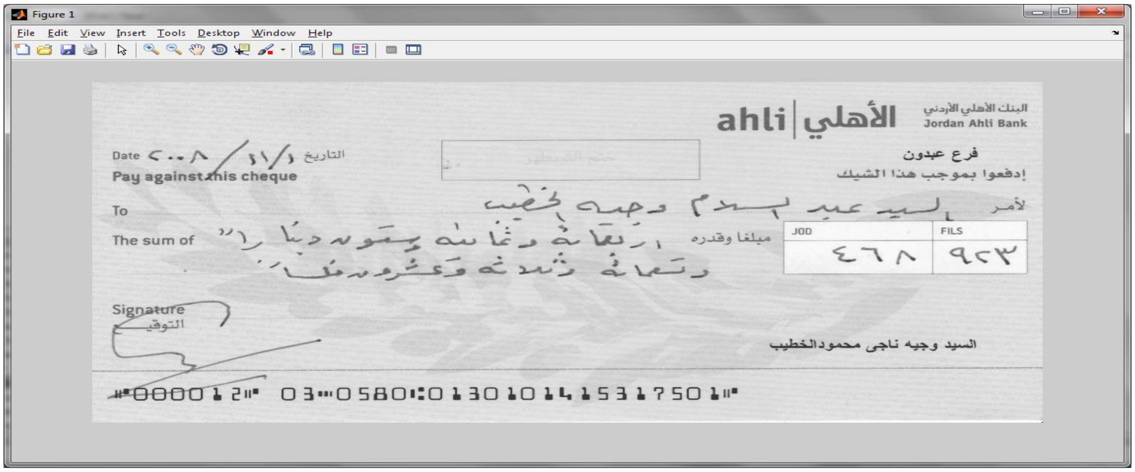


Figure 3.12: image of the check after conversion to gray scale and noise removal

The figure 3.13 shows the histogram of the check before enhancement:

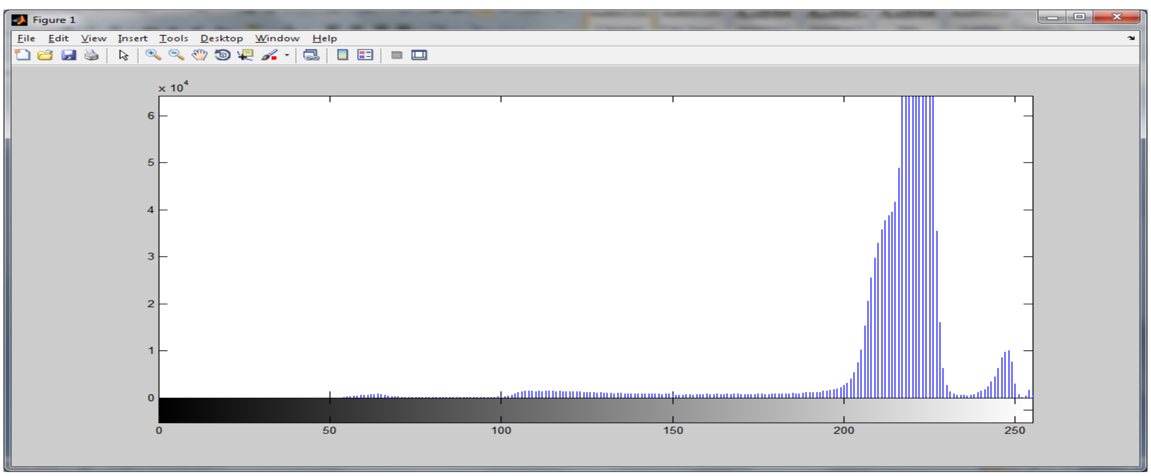


Figure 3.13: Histogram of the image in figure 3.12

After enhancing the image of the check using histogram equalization function, the histogram of the enhanced image looks as shown in figure 3.14, histogram of the check after enhancement:

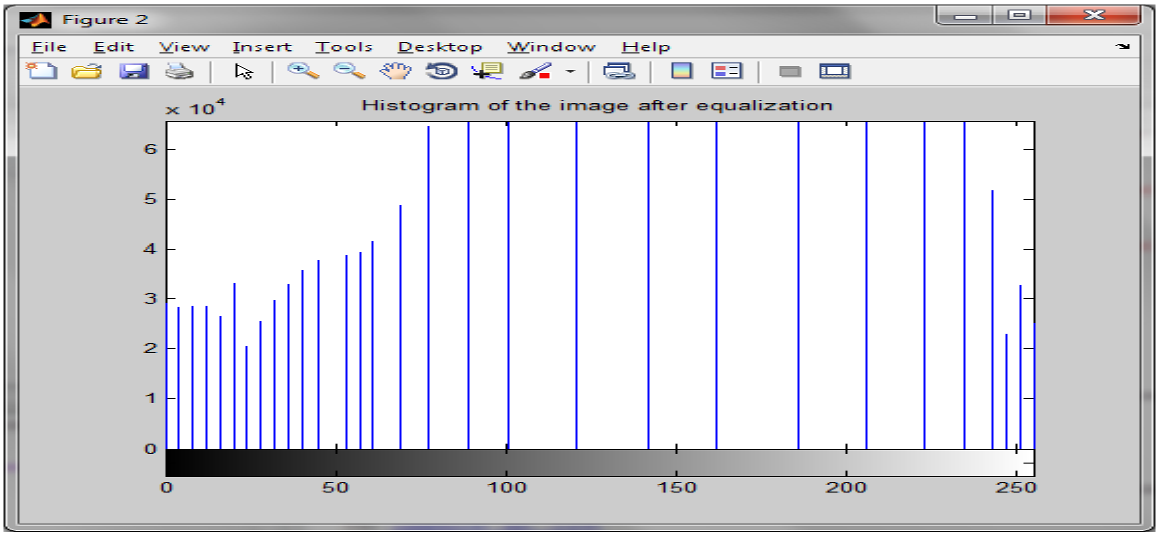


Figure 3.14: Histogram after performing equalization

The check after contrast enhancement is shown in figure 3.15:

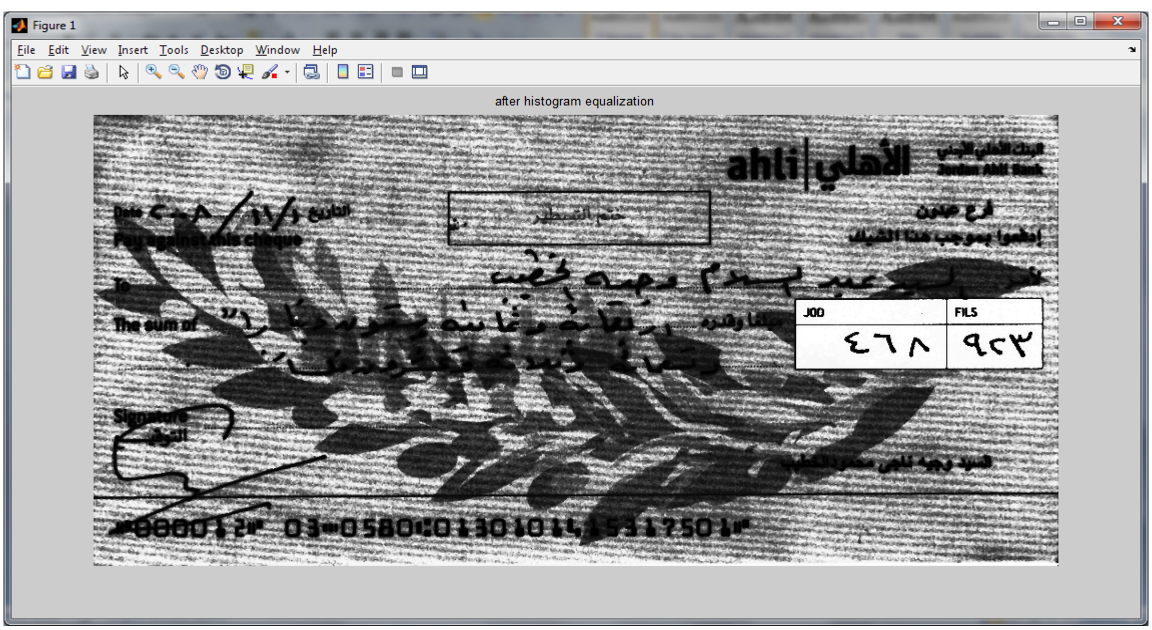


Figure 3.15: the check image after contrast enhancement using histogram equalization

In most of the state of the art, they remove the background of the image. But in this case the concentration is only on the courtesy amount. Therefore, the background removal will be ignored.

3. **Binarization**: This stage is similar to the binarization stage of the Offline Handwritten Hindi Digit Recognition system. Binarization is Converting the image into black and white. The technique used to convert the image to black and white is **(level = graythresh(image);)** this function from the Image Processing tool box in matlab is used to compute a global threshold (level) that can be used to convert an intensity image to a binary image with (im2bw) function. The level is a normalized intensity value that lies in the range [0, 1]. The graythresh function uses Otsu's method, which chooses the threshold to minimize the intraclass variance of the black and white pixels. In figure 3.16 the check is shown in black and white

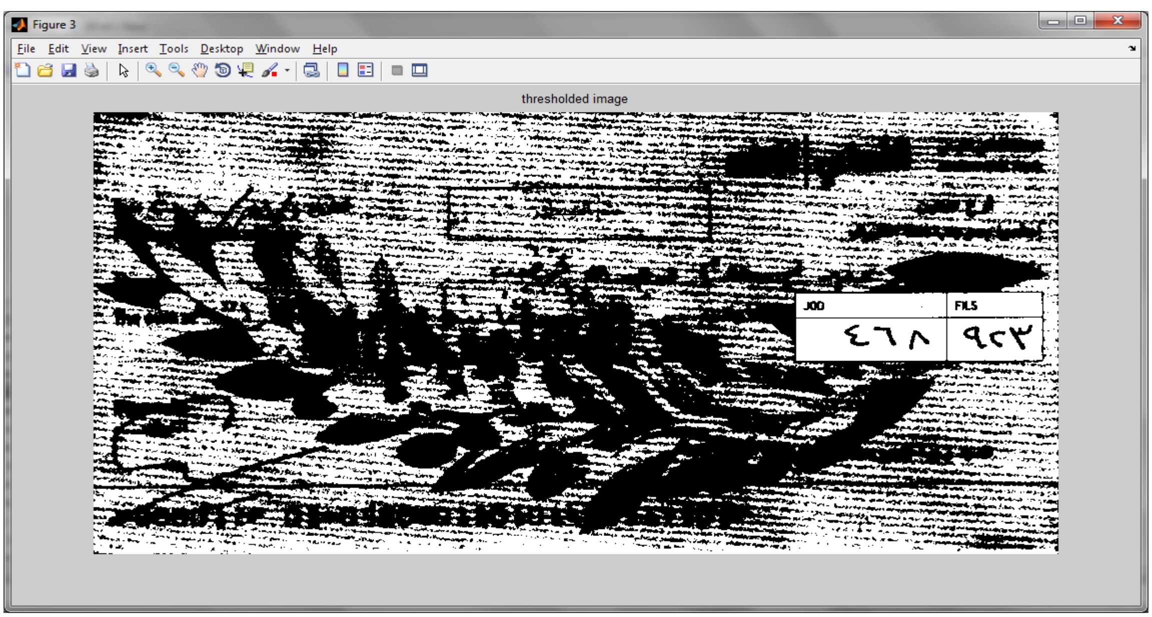


Figure 3.16: the check after conversion to black and white

4. **Image Inversion**: This stage is also the same as Image inversion stage in Offline Handwritten Hindi Digit Recognition System. Inversing the Image is necessary since after applying the binarization, the background of the image takes the ones pixels and the foreground takes the zeros pixels. As known in Image Processing using matlab, they deal with image as ones pixels. In figure 3.17 the inverse function is applied to the image for this purpose.

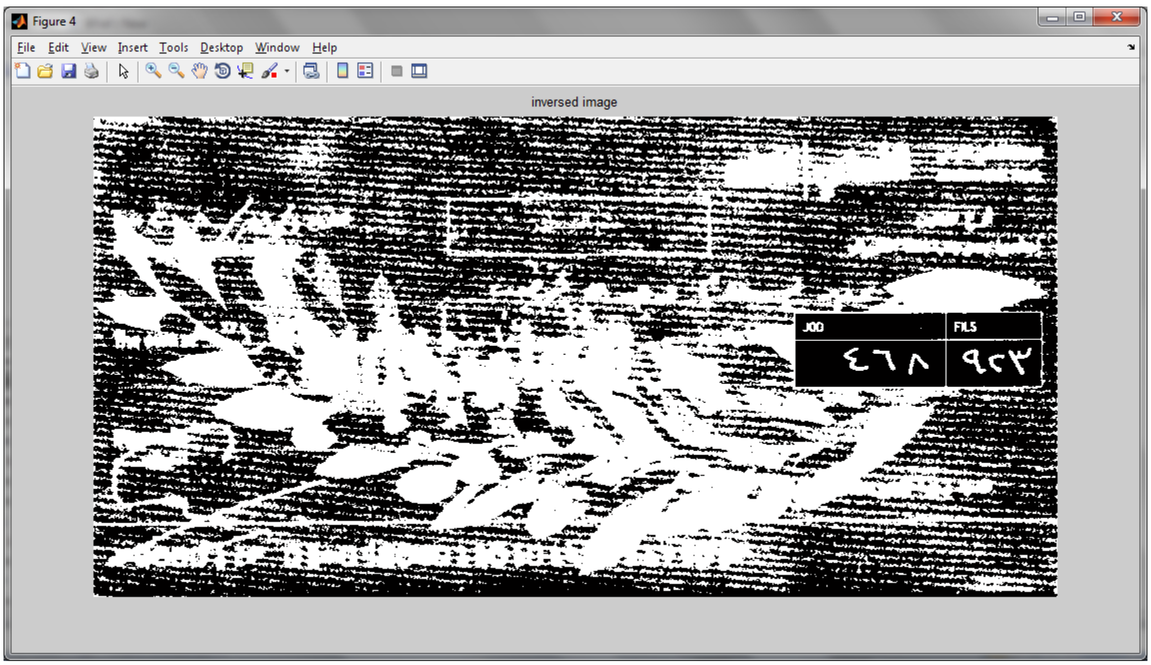


Figure 3.17: the check after performing image inversion

3.3.2 Cropping the location of the courtesy amount

The courtesy amount location now at this stage should be cropped in order to recognize the amount inside it. The image is cropped by the (imcrop) function in image processing tool box in matlab. After the preprocessing stages are performed, a message appears to the user to crop the location of the courtesy amount as shown in figure 3.18:

Figure 3.18: massage appears to the user for cropping

After the user clicks ok and crops the location manually, the location will be stored and recalled for other stages. The figure 3.19 shows the courtesy amount after cropping.

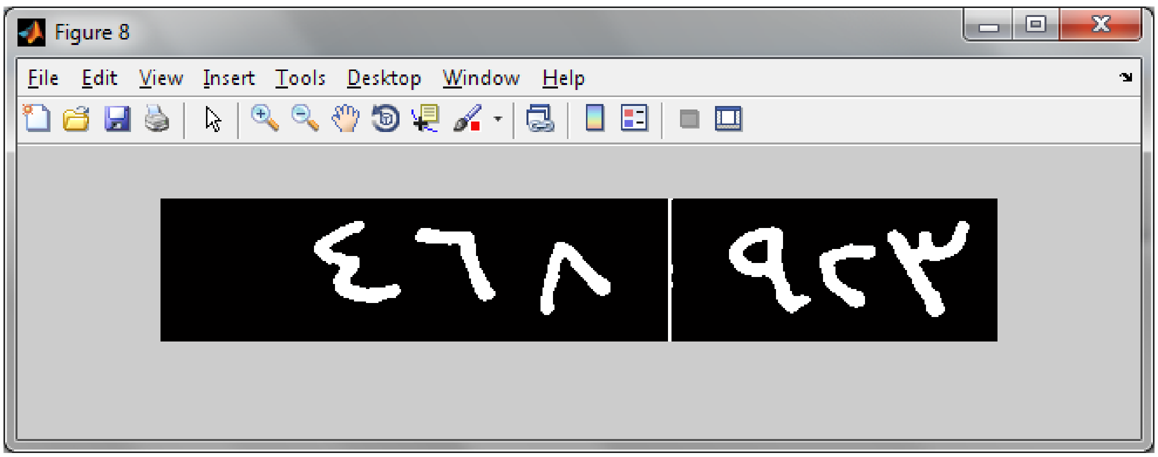


Figure 3.19: the courtesy amount after being cropped

3.3.3 Segmentation

In this stage a segmentation algorithm was designed and implemented to achieve the main goal of segmenting the string of digits in the courtesy amount location. The segmentation algorithm used is the Marker-Controlled Watershed Segmentation algorithm. It partitions the string of digits into individual digits depending on the regions and their properties. The segmentation was applied on different types of samples. It was applied on some samples from real bank checks; it was also applied on some samples written by hand these are 60 samples written as a string of digits separated by a comma.

Most of the samples were written in Hindi digits but few of them were written differently such as Arabic and English characters, Arabic and Hindi digits, and mix of characters and digits from a car plate. All these samples were used to test the efficiency of the segmentation algorithm used. In the following steps is a description for the Marker-controlled watershed segmentation algorithm in general.

Marker-controlled watershed segmentation follows this basic procedure: Sivanandam, Sumathi & Deepa, (2006)

1. Compute a segmentation function. This is an image whose dark regions are the objects you are trying to segment.
2. Compute foreground markers. These are connected blobs of pixels within each of the objects.
3. Compute background markers. These are pixels that are not part of any object.
4. Modify the segmentation function so that it only has minima at the foreground and background marker locations.
5. Compute the watershed transform of the modified segmentation function.

The segmentation process in this system for the cropped courtesy amount follows the following procedure:

1. Applying the Gradient Magnitude as the segmentation function. By using the Sobel edge masks, imfilter, and some simple arithmetic to compute the gradient magnitude. The gradient is high at the borders of the objects and low (mostly)

inside the objects. In other words the gradient magnitude determines the edges of each digit.

2. Performing some morphological operations such as Image dilating and image filling to fill in empty edges and close openings and holes.

3. Labeling the image

4. Measuring properties of image regions using (regionprops), the property used is ‘Centroid’.

5. Performing the Watershed Transform. In this step the background markers will be determined. The background markers shouldn’t be too close to the edges of digits in order to be able to segment. This step will segment the objects in the image using thin lines.

6. Computing the Watershed Transform of the Segmentation Function. In this step foreground markers will be determined using the ‘imimposemin’ function it imposes minima. It modifies the intensity image, using morphological reconstruction so it only has regional minima wherever BW is nonzero. BW is a binary image the same size as the original image. After this step, the digits should be segmented.

7. Visualizing results using many visualization techniques:

a. One visualization technique is to superimpose the foreground markers, background markers, and segmented object boundaries on the original image.

b. Another useful visualization technique is to display the label matrix as a color image.

c. Another one is to use transparency to superimpose the pseudo-color label matrix described on (b) on top of the original intensity image.

A demonstration for these steps and their results will be shown in chapter 4.

3.3.4 Feature Extraction

The segmentation stage is defined previously as the process of partitioning the digital image into multiple regions that can be associated with the properties of one or more criterion. These properties are gray level, color, texture shape, and others. The result of the segmentation stage is to have the digits separated from each other by lines, colors or others. However this doesn’t mean to split digits or even have a separate image for each digit or to extract each digit as a separate image. In order to have this done a specific digit splitting function is built to extract the digits from the original image into sub images. The digit splitting function performs extraction depending on the properties of the image such as Area, Perimeter, Diameter, Centroid, image intensity, mean and others. It also performs extraction of digits mainly depending on the bounding box property where the bounding box is the smallest rectangle containing the digit with all its pixels. After the bounding box is determined, extracting the pixels becomes very easy. The digit splitting function was applied on the segmented image and successfully performed the purpose of splitting the digits each in an individual digit. Another function called (regionprops) was used from the image processing tool box, this function Measures properties of image regions (blob analysis). After the image is labeled into a label matrix, the (regionprops) function is used to measure the properties of the labeled region. It can measure one or more than one property at a time. It can even measure all the properties of an image. The digit splitting function used to perform extraction of the digits and split them into separate digits.

Extracting and splitting the digits from the original image is described in the following steps:

1. Labeling each region in the image including the digits and the separator, and giving each region a number.

2. Choosing number of properties required to extract the digits. In this case all properties are chosen by using (all) with the region props function. It will measure the following properties: Mean, Intensity, Area, Perimeter, Centroid, and Diameter.

3. The properties of each labeled region are displayed in the command prompt.

4. Calculating the boundaries of each label to make it easier for extraction.

5. Measuring the properties and extracting digits as sub images by finding the bounding box of each labeled region.

A demonstration of the previous steps and their results is described in chapter 4.

After the feature extracting stage is done, the courtesy amount is divided into separate digits. Now each separate digit is ready to be recognized by the Offline Handwritten Hindi Digit Recognition System which is used for recognizing separate Hindi digits.

3.3.5 Digits Recognition

In this section the digits will be recognized based on the Offline Handwritten Hindi Digit Recognition System described in section 3.2. After the string of digits is segmented and divided into separate digits as described in 3.3.3 and 3.3.4 sections, the digits will be treated separately by calling the recognition function that recognizes each digit apart.

The second subsystem calls the first subsystem in this stage to recognize each digit separately as shown in figure 3.20.

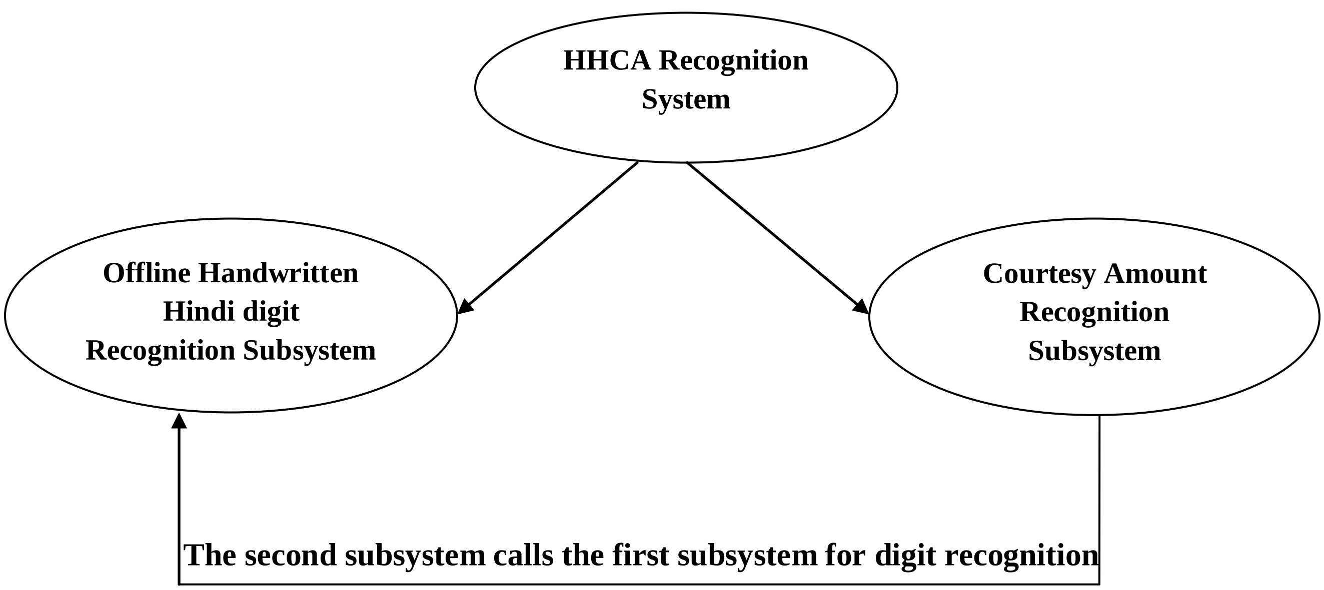


Figure 3.20: This figure shows the relationship between the two subsystems of the HHCA recognition system

Chapter 4

Experimental Results

4.1 Introduction

The HHCA recognition system is designed using Matlab R2008a, the Image Processing and NN Tool Boxes were mainly used. The system worked on samples of Hindi digits from zero to nine. The samples were written by hand and then entered to the computer using a scanner. The Image Processing Tool Box was used for manipulating the scanned images for preprocessing, segmentation, and feature extraction stages. The NN Tool Box was used for training and testing the samples for recognition stages.

The system is divided into two subsystems. Both subsystems contain number of stages, and both of them achieved good results in all stages. The results of each subsystem in all its stages will be mentioned in details in the following two sections.

4.2Results of the Offline Handwritten Hindi Digit Recognition System:

In the Offline Handwritten Hindi Digits Recognition System, the digits were recognized as separate digits. The Hindi digits from one to nine were taken from real world samples. These samples were written by hand and they are divided into two types. The first type contains 1000 samples that were written using a regular blue pen. The second type contains 1000 samples that were written using a black marker. These samples must pass through all stages of the Offline Handwritten Hindi Digit Recognition system. The samples first pass through the preprocessing stage to enhance the handwritten digits, then they pass through the feature extraction stage that was used to extract the digits from images and store them as vectors to be the input for the NN. The NN structure used is the Multi Layer Perceptron (MLP) with backpropagation training algorithm which consists of an input layer, one hidden layer, and an output layer. In the digit recognition stage the samples were used for training and testing. In the following paragraphs is a description for both training and testing stages and their results.

- **Training the network:**

The network is trained using two types of samples. The 1000 samples of Hindi digits that were written using a regular blue pen, and the 1000 samples of Hindi digits that were written using a black marker. In both cases the training results were excellent and the recognition rate ranges from 90% to 99%. In the training stage the samples first were entered to the network as input set. The digits were trained separately to record the rates accurately. While training a digit, a neural network training tool window is launched to show the progress while training as shown in figure 4.1, and to show the performance, training state, and the regression plot. Also the performance plot that is shown in figure 4.2 shows the network performance. Then the training state plot In figure 4.3 shows training state values which are Gradient, Validation, and Learning Rate. Finally the regression plot in figure 4.4 shows the linear regression of targets relative to outputs.

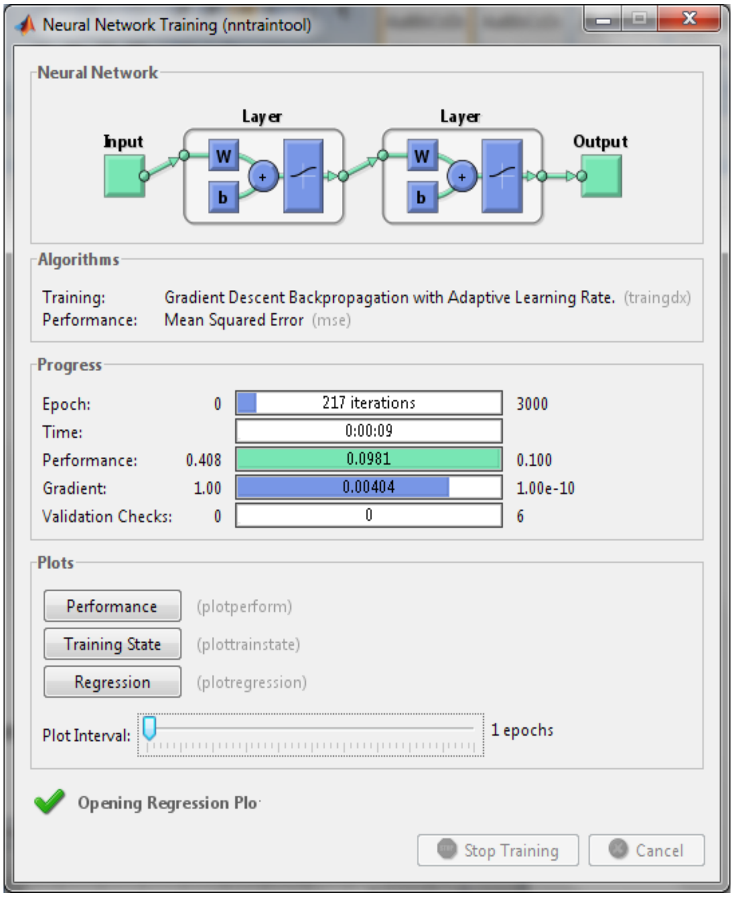


Figure 4.1: Neural Network training tool window

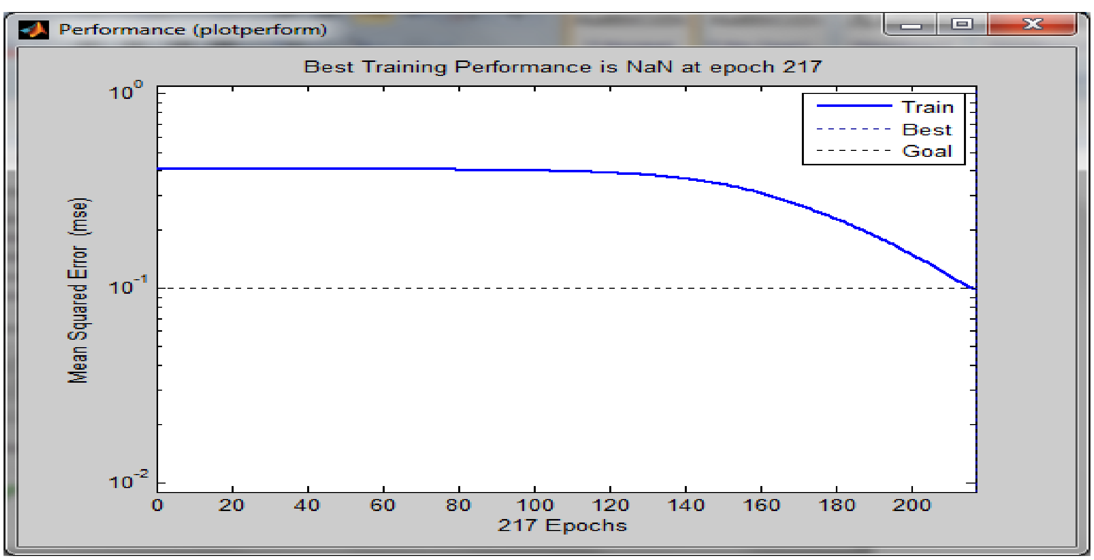


Figure 4.2: Performance Plot

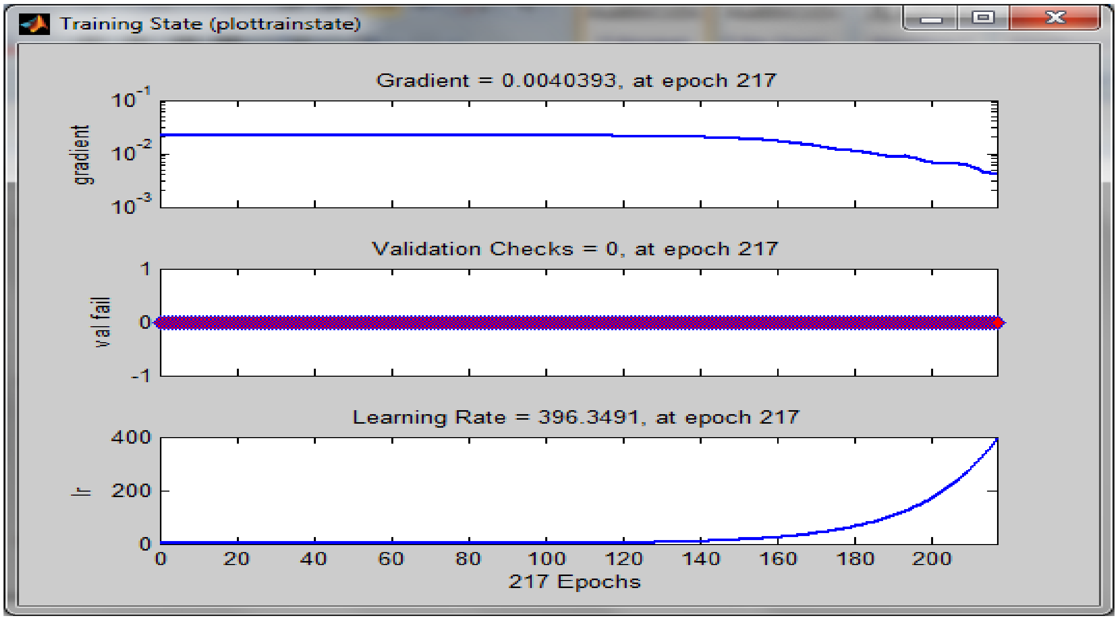


Figure 4.3: Training State plot

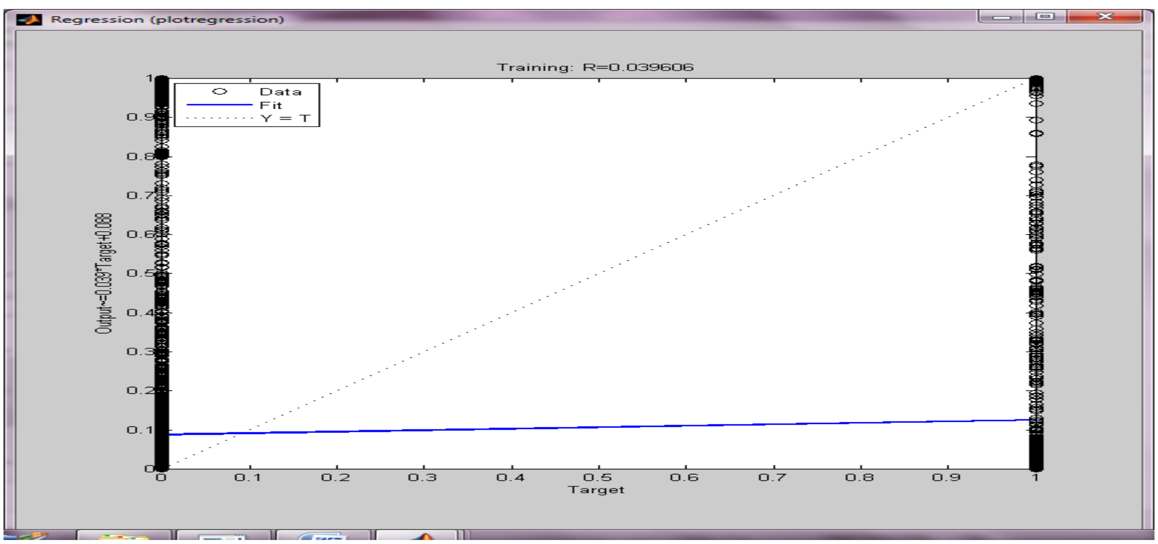


Figure 4.4: Regression plot

- **Testing the Network:**

At this stage the network was tested using the testing samples. As mentioned previously there are two types of testing samples, 40 samples for each type. The samples that were written by a regular blue pen were written and tested before the samples that were written by a black marker. The reason behind this is the testing results. After testing the samples of a regular blue pen, the recognition rate was very bad and unexpected. Many solutions and attempts were introduced and applied to the samples in order to enhance the recognition rate, but the recognition rate didn’t improve or increase. Therefore, other samples were written using a black marker to see if the recognition rate got better results. The samples were written by different people from different ages and having different writing styles. After testing the samples, the recognition rate increased significantly. It was 51.6%; however, this recognition rate was not as desired. Therefore, many attempts and solutions were introduced to enhance this rate. After many changes, the recognition rate improved gradually until it reached an acceptable rate of 74.4%. The rest of the section will be a description in details of how the recognition rate was enhanced gradually and got its final results.

First, the samples that were written by a regular pen were tested, but the recognition rate was very bad and unexpected. Therefore; many solutions were introduced to enhance the recognition rate. First solution was performed on the preprocessing stage to change the size of the image. The imresize function was used to change the size. Using imresize directly was wrong since it deletes one raw with one column. Using this function changes the structure of the image and that's why the recognition rate was very low. Instead of using imresize function, down sampling with interpolation method was used. Also resizing the image into very small dimensions such as (30*30) or (60*60) was not acceptable since resizing causes losing too many pixels of the image which results in bad recognition. The problem of resizing was solved, also many changes were performed to remove noise from images and to make the handwriting style appear clearer and thicker using the morphological operations. However, the recognition rate remained low and with no enhancement. After all the previous attempts, a decision was taken to write new samples using a black marker. The new samples consist of 1000 samples for training and 40 for testing. After writing the new samples, they were used for training and testing. As a result of testing the new samples, the recognition rate increased significantly and was recorded as 51.6%. This result was a very big achievement compared with the results before changing the samples. However, the achieved recognition rate was not as desired, therefore, some changes were performed in order to increase the recognition rate. After trying many changes in the preprocessing stage, something was noticeable, which is the thinning step. The thinning step lets the digit have its skeleton shape. This step resulted in losing pixels from the original image and affected the shape of the original one. Therefore, the thinning step was excluded from the preprocessing stage as an attempt only to see if the recognition rate gets better. After these changes, training and testing were performed again. As a result of this change the recognition rate really increased and got recognition rate of 63.1%. Still this result was not as desired so another solution was taken in order to increase it. This step was increasing the testing samples. The testing samples were increased by adding 20 more samples to each digit to have a total of 60 samples for each digit. After this addition, the testing was performed again and the recognition rate increased and got better results than before. The recognition rate became after increasing the samples 74.4%. As described above, the testing stage passed through many changes in order to enhance the recognition rate. The recognition rate increased gradually after each step of enhancement until it reached the last recognition rate of 74.4%.

The following three steps will show the recognition rate resulted after the changes done for enhancement:

1. In table 4.1 shows the results of the recognition rate of each digit and the total recognition rate after using new samples written by a black marker.

Digit	Recognition rate
Zero	**100%**
One	**87%**
Two	**42%**
Three	**35%**
Four	**37%**
Five	**53%**
Six	**45%**
Seven	**39%**
Eight	**58%**
Nine	**20%**
Total Recognition rate:	**51.6%**

Table 4.1: recognition rate after changing the samples using a black marker

2. In table 4.2 shows the recognition rate after the second change which is removing the thinning step from the preprocessing stage:

Digit	Recognition rate
Zero	**100%**
One	**60%**
Two	**37%**
Three	**55%**
Four	**55%**
Five	**42%**
Six	**80%**
Seven	**65%**
Eight	**80%**
Nine	**57%**
Total Recognition rate:	**63.1%**

Table 1.2: recognition rate after removing the thinning step

3. Table 4.3 shows the last step applied for enhancing the recognition rate which is increasing the testing samples from 40 for each digit to 60 samples.

Digit	Recognition rate
Zero	**100%**
One	**73%**
Two	**73%**
Three	**75%**
Four	**65%**
Five	**60%**
Six	**82%**
Seven	**73%**
Eight	**86%**
Nine	**57%**
Total Recognition rate:	**74.4%**

Table 4.3: Accuracy of each digit and the total recognition rate

4.3 Results of the Courtesy Amount Recognition System:

In the Courtesy Amount Recognition System the courtesy amount was recognized by passing through different stages of preprocessing, segmentation, feature extraction and recognition. The system successfully achieved the purpose of recognizing the courtesy amount starting with cropping the location of the courtesy amount, passing through successful segmentation and feature extraction stages, and ending up with the offline handwritten recognition system that also successfully recognizes digits with a recognition rate of 74.4%. It is worth to mention the results of each step in this system:

Segmentation and Feature Extracting Stages:

These two stages performed successfully and very accurately in segmenting the courtesy amount and extracting the digits contained in it. These two stages were experimented on the courtesy amount that is extracted from real bank checks, and performed successfully and accurately. Also they were experimented and performed accurately on other samples that were written by hand as a string of Hindi digits. A demonstration will be shown for both the segmentation and the feature extraction performed on different types of samples.

The following is a demonstration for the segmentation algorithm applied on the cropped image of the courtesy amount location in steps:

1. The figure 4.5 is the courtesy amount cropped from the bank check.

Figure 4.5: the courtesy after cropping

2. The figure 4.6 is the courtesy amount after applying the gradient magnitude segmentation function.

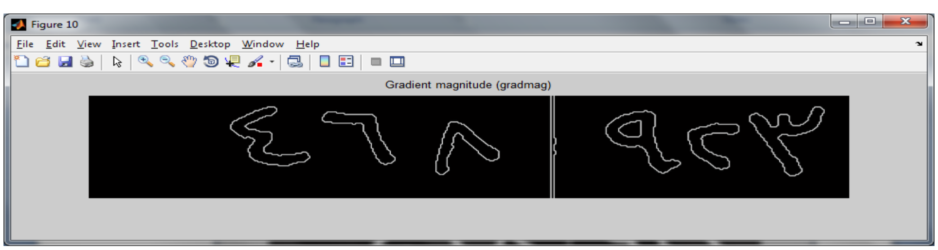


Figure 4.6: the courtesy after applying the segmentation function

3. The following two figures 4.7 and 4.8 show the courtesy after applying morphological operations on them (dilating and filling).

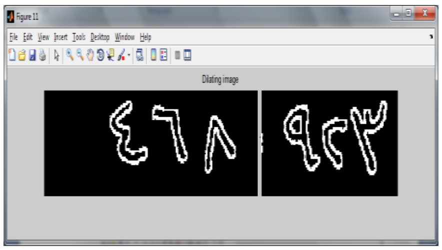


Figure 4.7: Courtesy after Dilate

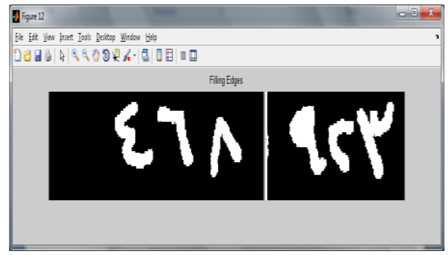


Figure 4.8: Courtesy after filling edges

4. In figure 4.9 the courtesy is shown after using the regionprops function using the centroid property.

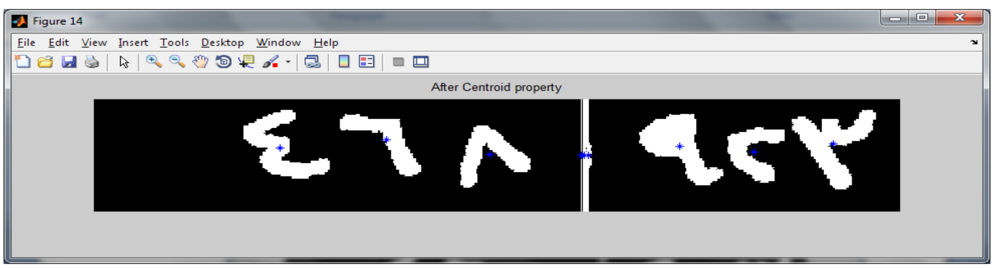


Figure 4.9: Courtesy after applying the centroid property

5. Applying the watershed transform function to determine background markers to segment digits using thin lines as shown in figure 4.10.

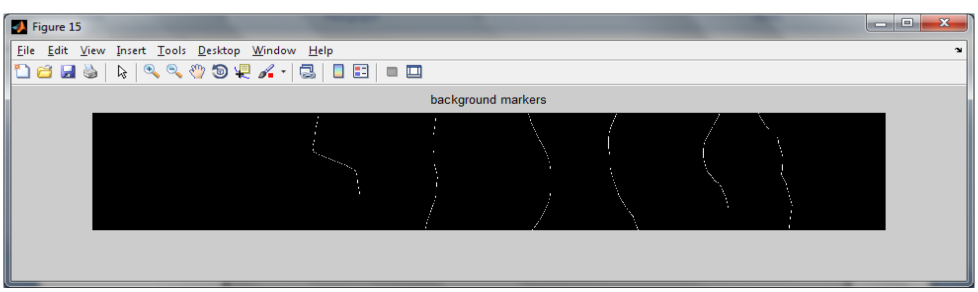


Figure 4.10: Courtesy after applying watershed transform function

6. Applying the water transform function using imimposemin function to determine the foreground markers as shown in figure 4.11.

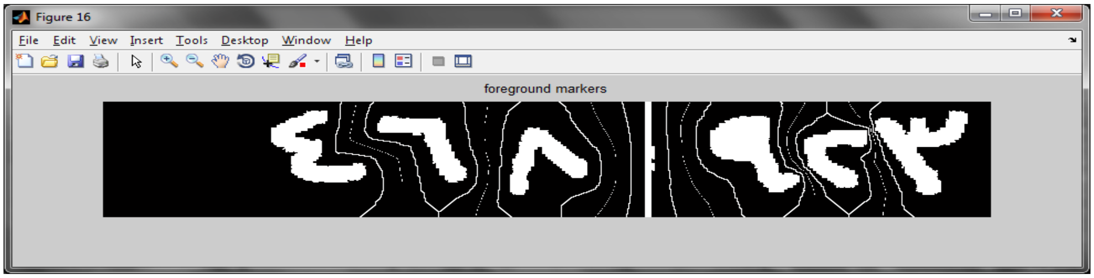


Figure 4.11: Courtesy after showing foreground markers

7. Visualizing results:

    a. Visualizing results as in figure 4.11, by showing the foreground markers, background markers, and segmented object boundaries on the original image.

    b. Visualizing results by displaying the label matrix as a color image as shown in figure 4.12.

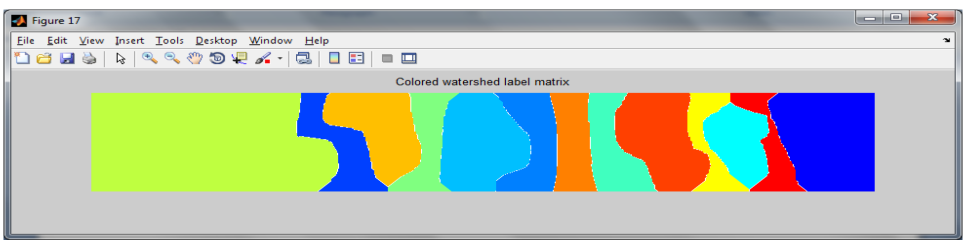


Figure 4.12: display result of segmentation using colored watershed label matrix

    c. Visualizing results by using transparency to superimpose the pseudo-color label matrix in figure 4.12 on top of the original intensity image in figure 4.5.

Figure 4.13: display results of segmentation using transparency over the original image

The following steps show the segmentation process that is applied on the samples of strings of digits written by hand:

These samples follow the same procedure of the cropped image of the courtesy amount, with slight differences at the beginning in the preprocessing stage such as image resize, color conversion, and many others. The following figures with steps show the procedure:

1. Preprocessing steps Image resize and binarization as in figure 4.14.

Figure 4.14: image after binarization

2. The string after applying the gradient magnitude segmentation function as in figure 4.15

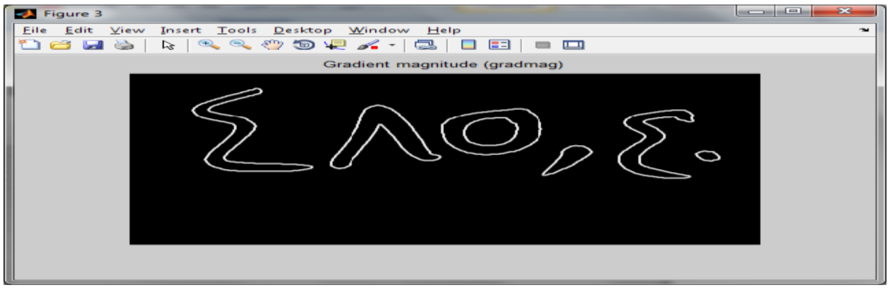
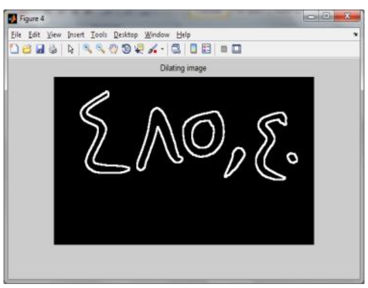
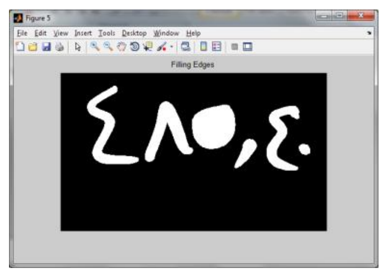


Figure 4.15: the string of digits after applying the segmentation function

3. The following two figures 4.16 and 4.17 after performing image dilating and filling.





4. Figure 4.18 after using the regionprops function using the centroid property.

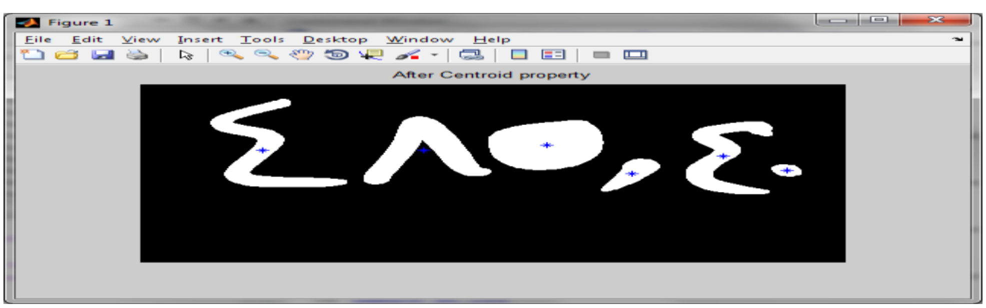


Figure 4.18: the string after applying the centroid property

5. Applying the watershed transform function to determine background markers to segment digits using thin lines as in figure 4.19.

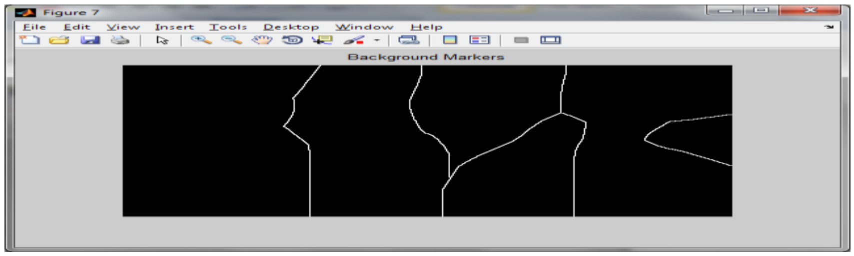


Figure 4.19: the string after watershed transform function is applied

6. Applying the watershed transform function using imimposemin function to determine the foreground markers as in figure 4.20.

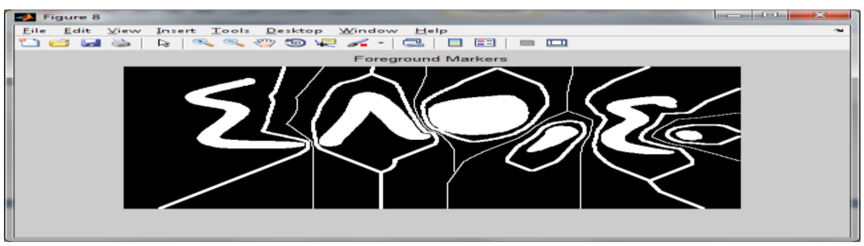


Figure 4.20: the string after showing foreground markers

7. Visualizing results by displaying the label matrix as a color image as in figure 4.21

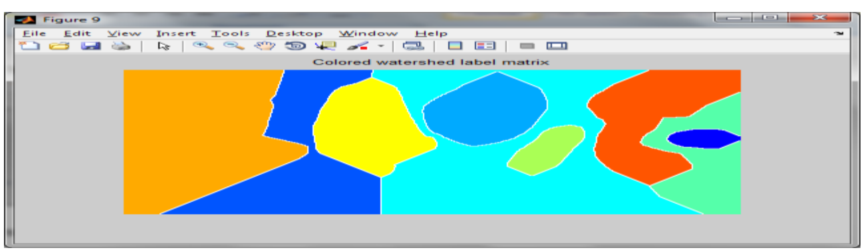


Figure 4.21: display result of segmentation using colored watershed label matrix

8. Visualizing results by using transparency to superimpose the pseudo-color label matrix in figure 4.22 on top of the original intensity image in figure 4.17 .

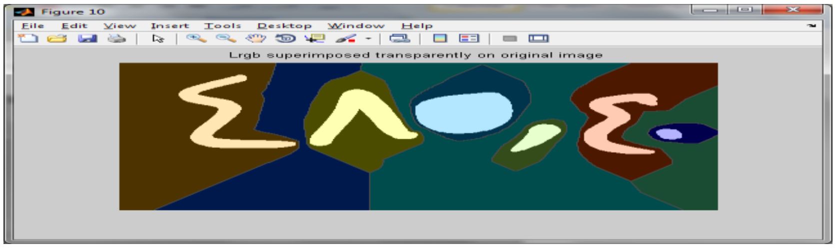


Figure 4.22: display results of segmentation using transparency over the original image

After this stage, the courtesy amount is segmented into separate regions, each region separates the digit from other regions. Now the image is ready to the next step, to be divided into separate digits and extracted.

The following is a demonstration of the feature extraction algorithm used for extracting and splitting the digits of the courtesy amount into separate digit:

1. The following image is a courtesy amount cropped from a real bank check, the figure 4.23 shows the boundaries of each digit, and how each digit is given a number.

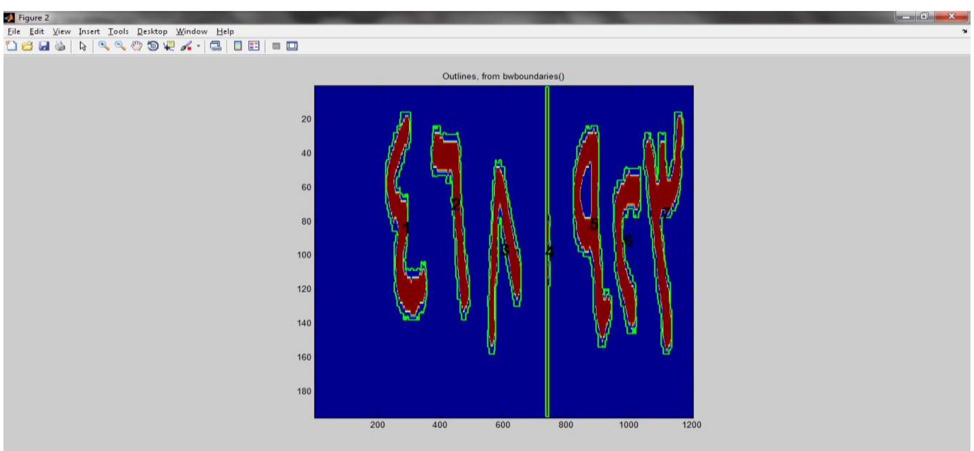


Figure 4.23: this figure shows how digits are bounded and labeled by numbers

2. In table 4.4 the result of displaying the properties of each digit in the image is according to the number given to each digit:

Blob #	Mean	Intensity	Area	Perimeter	Centroid	Diameter
# 1	163.9	7227.0	637.7	283.8	84.3	95.9
# 2	163.0	5322.0	490.4	443.0	70.2	82.3
# 3	160.2	5671.0	538.3	597.6	97.0	85.0
# 4	114.8	2279.0	422.7	741.4	98.0	53.9
# 5	163.3	8001.0	504.6	881.0	82.0	100.9
# 6	161.2	4886.0	475.5	992.8	91.8	78.9
# 7	167.2	8093.0	678.9	1110.4	75.7	101.5

Table 4.4: this table shows the properties of each labeled digit

3. The figure 4.24 shows how the string of the courtesy amount is divided into digits and each digit is extracted:

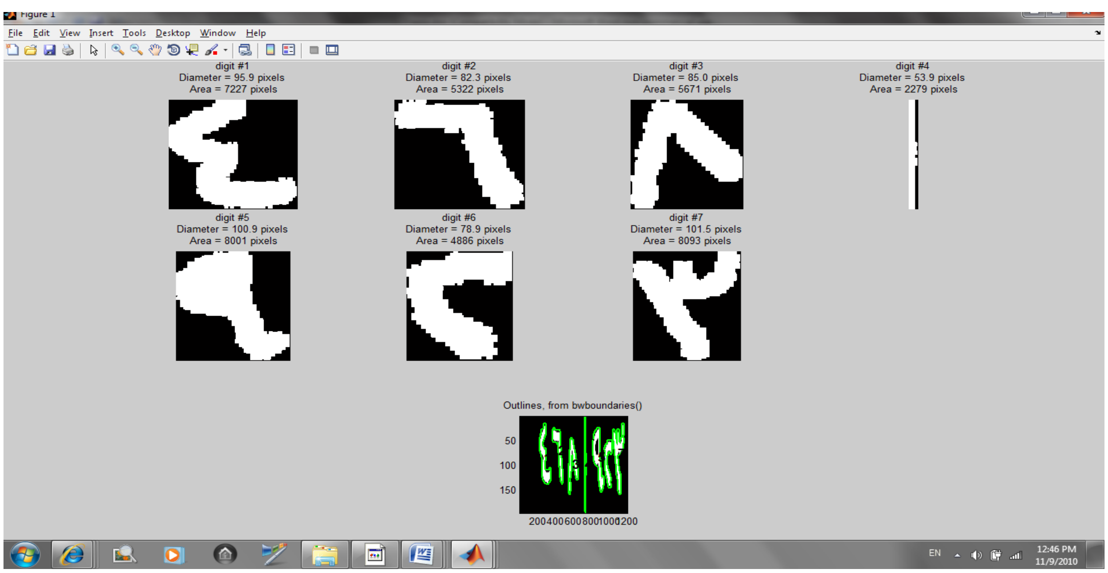


Figure 4.24: this figure shows the final extracted digits

The following steps show the Feature extraction method that is applied on the samples of strings of digits written by hand:

1. The figure 4.25 shows the boundaries of each digit, and how each digit is given a number.

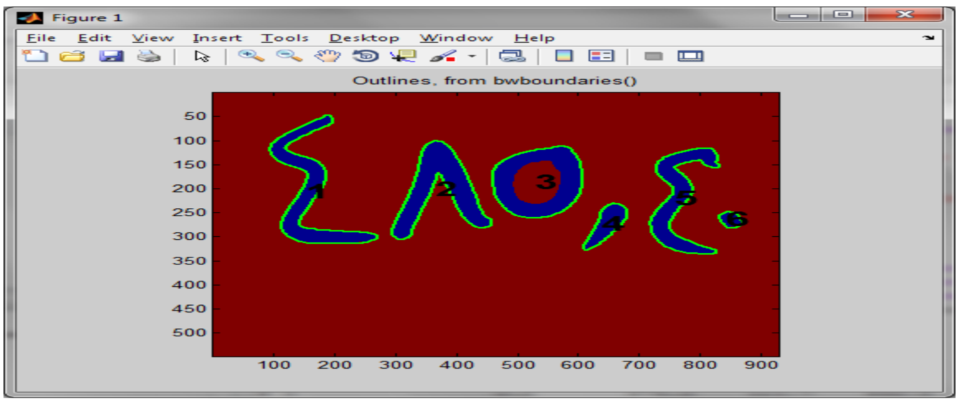


Figure 4.25: Handwritten digits are bounded and labeled by numbers

2. In table 4.5 is the result of displaying the properties of each digit in the image according to the number given to each digit:

Blob #	Mean	Intensity	Area	Perimeter	Centroid	Diameter
# 1	0.2	12034.0	1018.8	159.8	205.7	123.8
# 2	0.1	12914.0	806.4	372.7	202.2	128.2
# 3	0.4	17658.0	501.8	535.1	186.4	149.9
# 4	0.2	3225.0	274.4	647.2	274.1	64.1
# 5	0.1	9875.0	799.8	765.7	220.9	112.1
# 6	0.2	975.0	121.7	850.6	264.9	35.2

Table 4.5: this table shows the properties of each labeled digit

3. The figure 4.26 shows how the string of the courtesy amount is divided into digits and each digit is extracted:

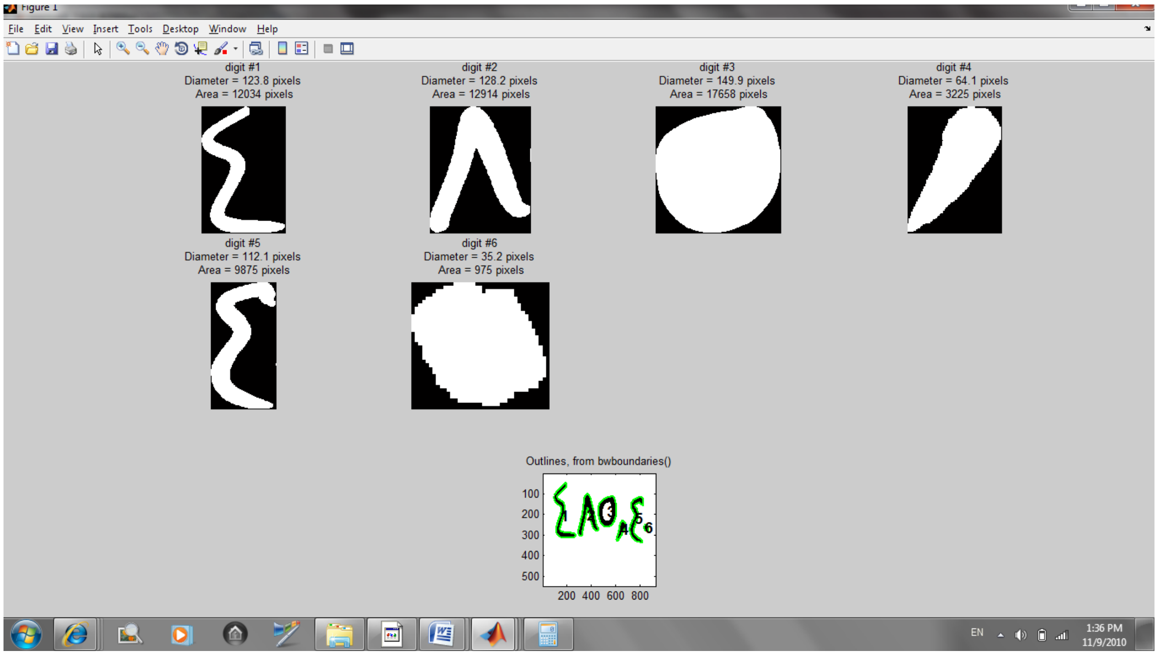
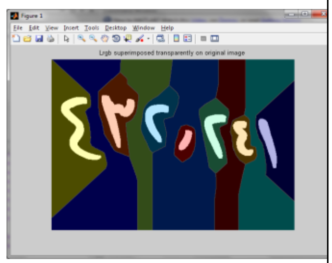
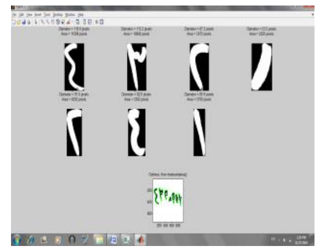
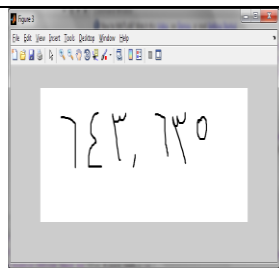
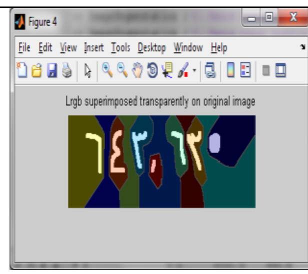
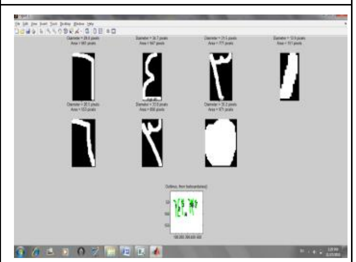
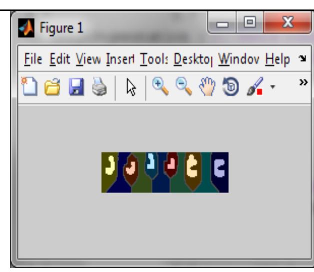
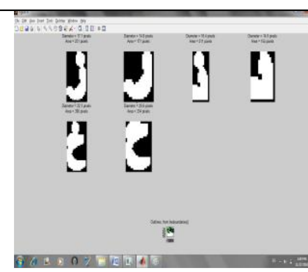
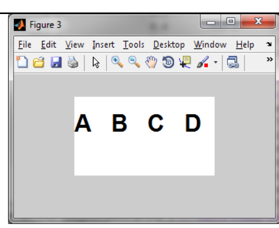
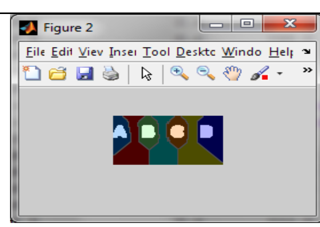
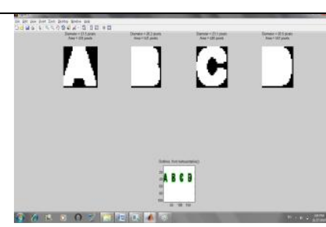


Figure 4.26: Handwritten digits are extracted as separate digits

In fact, these two stages not only work for the offline handwritten Hindi digits, but also on different strings of digits and characters of different languages. This advantage makes these two stages applicable not only for bank checks recognition, but also for other applications such as postal code recognition, and car plate recognition. New samples were taken to experiment these two stages. Some of these samples were online written by the mouse, and others were offline entered by a scanner. Also some of the samples were printed by a keyboard, and another sample was taken of a car plate. Some of the samples were written using Arabic letters and others using English letters. The system performed accurately for all kinds of the mentioned samples. On the other hand there is a limitation in the segmentation stage for connected and touching characters or digits. It doesn't segment touching or connected digits it only works for separated ones.

The table 4.6 shows all kinds of samples that were taken and segmented and extracted successfully:

Sample Type	Original Image	After Segmentation	After Extraction
Offline String of digits			
Online String of digits			
Printed Arabic Letters			
Printed English Letters			

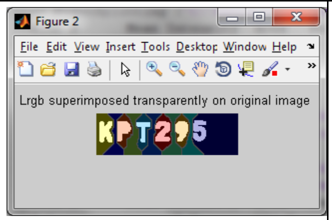
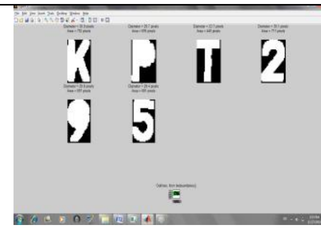
Car Plate			

Table 4.6: different image samples after applying both segmentation and feature extraction stages

As shown in table 4.6 that the designed methods for segmentation and feature extraction work successfully and efficiently for all types of characters and digits in different languages. This may help to use them in different applications other than bank checks, such as postal codes and car plates.

In short, the whole system in all its stages was performing properly. And it achieved the main purpose of recognizing the courtesy amount in Arabian bank checks. The system of recognizing offline handwritten Hindi digits also performed properly with a good recognition rate as mentioned in section 4.2.

4.4 The overall system described in a simple Graphical User Interface:

A simple GUI was designed to show the overall system how it works. Figure 4.27 illustrates the GUI and beneath it the description of it.

Figure 4.27: this figure shows the GUI after running

In this simple GUI, there are four bush buttons. The first one is the Load Button when clicking it; it will load the check image and the courtesy amount that was cropped by the user. The next button is the Segment Button, when clicking it; it will call the segmentation function to perform segmentation and will show the result in a separate figure window as shown in figure 4.28:

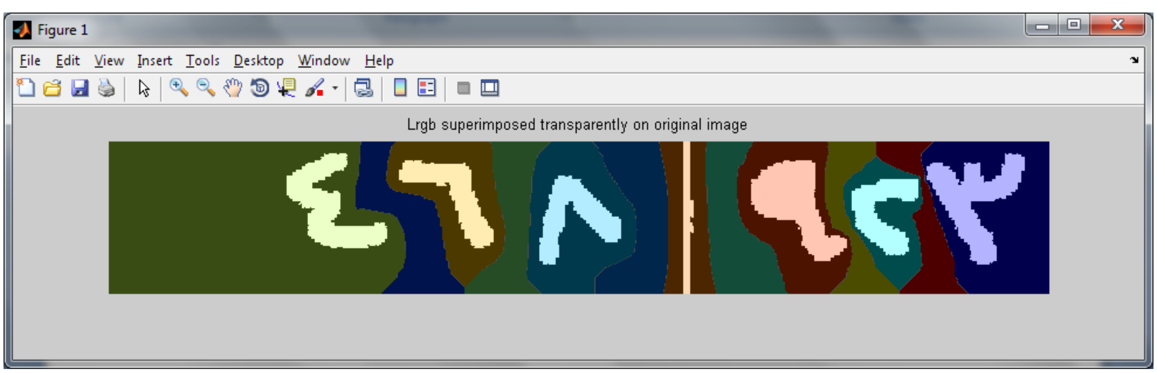


Figure 4.28: figure displayed after clicking segment button

The third button is the split button. When clicking it, it will call the splitting function to perform splitting and show the result in a separate figure window as shown in figure 4.29:

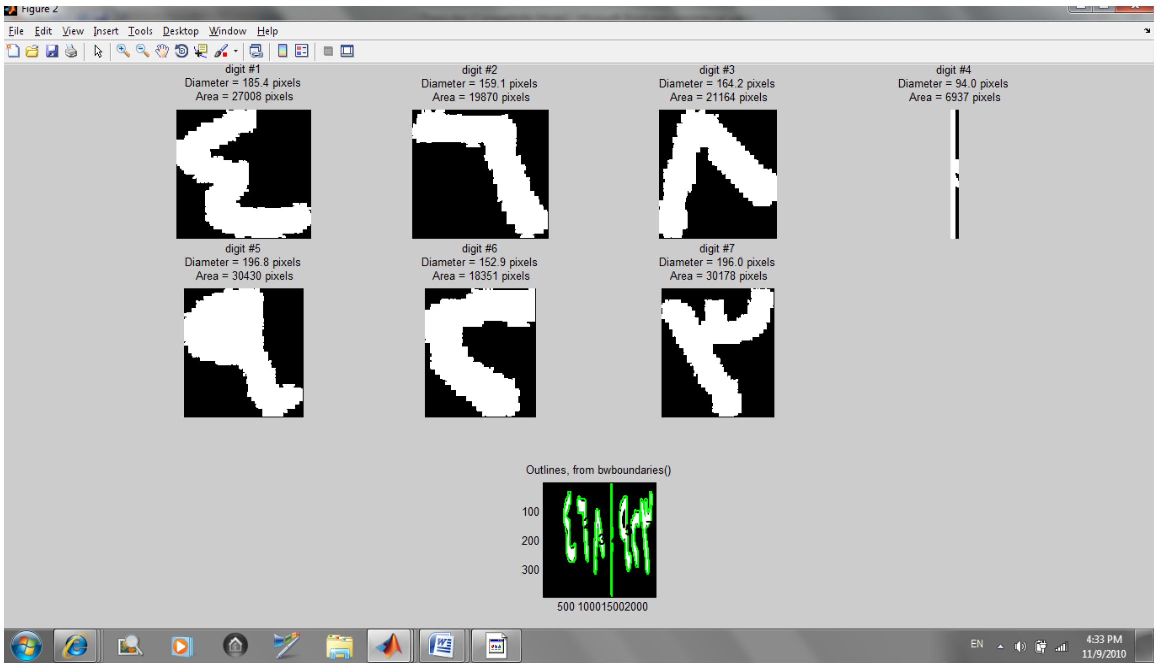


Figure 4.29: figure displayed after clicking split button

The fourth button is the recognition button. When clicking it, it will call the recognition function which will recognize the digits and print out the result in English numerals with a comma as a separator as shown in the check figure under the recognized digits tag. The last button is the recognized value. When clicking it, it will print out the value as a courtesy amount.

The recognition function will be called directly after the digits are extracted and divided into separate digits. The figure 4.30 shows how each digit is divided and recognized directly.

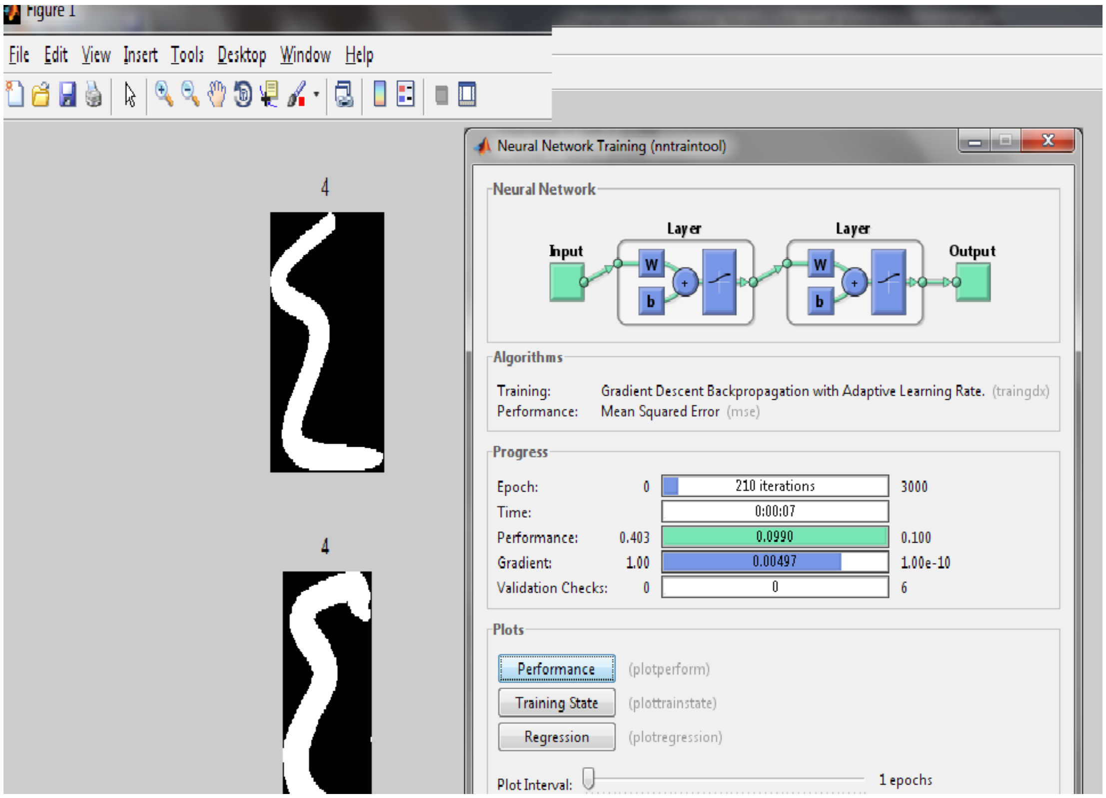


Figure 4.30: recognition of the digits after segmentation and feature extraction

As seen in figure 4.30 after each digit is segmented and extracted, the Neural Network training tool window will be launched to show how each digit will be recognized, and the resulted digit will be printed over the image of the digit. As mentioned in section 3.2 that the recognition rate achieved was 74.4% of testing samples. This means that the digits will be recognized around this rate. Now after this stage the overall designed system is completed starting from scanning the image of the bank check and ending with recognizing each separate digit apart, passing through many stages of cropping the courtesy amount, segmenting the string of digits, and extracting each digit.

Chapter 5

Discussion and Recommendations

5.1 Discussion:

The HHCA recognition system was described in detail in the previous chapters and so were the results. Now the results should be discussed and compared with the results of the related works. This chapter will explain the discussion of the experimental results compared with the related work results. As explained previously, the HHCA recognition system was divided into two parts. The first part was about the recognition of the offline handwritten Hindi digits and the second part was about the recognition of the courtesy amount in Arabian bank checks. The Offline Handwritten Hindi Digits Recognition subsystem was designed for recognizing separate Hindi digits written by hand and entered to the system by a scanner. These digits passed through the preprocessing and feature extraction stages before recognition. Multilayer Perceptron (MLP) architecture was used to recognize these digits, and a recognition rate of 74.4% was attained after many attempts to enhance this rate. Next, the Courtesy Amount Recognition subsystem was designed for recognizing the courtesy amount in Arabian bank checks. The goal was achieved successfully after passing the segmentation and feature extraction stages. The segmentation stage used for segmenting the string of digits into separate digits, this stage was efficient and successful for all kinds of strings of digits and characters in any language. But it was limited for touching and connecting digits or characters. Also the feature extraction stage was successful to extract handwritten digits, using region properties specifically using the Bounding Box property.

Most of the work that is related to the HHCA recognition system concentrate on different fields. Some of them concentrate on the segmentation and feature extraction stages and how they overcome their difficulties; others concentrate on the recognition rate of the classifier used for recognition. In the following sections the results of the HHCA recognition system will be discussed and compared to the related work results with respect to the subsystems of the designed system.

5.1.1 The comparative results with respect to the Offline Handwritten Hindi digit recognition subsystem:

Name	Approach	Handwritten type and database used	Recognition Rate
Nawas, Sarfraz, Zidouri, & Al-Khatib, (2003).	Offline Arabic Character Recognition Using RBF Network	Offline Handwritten	76%
Sadri, Suen, & Bui, (2003).	Handwritten Arabic/Persian digits Using Support Vector Machines (SVMs).	Offline and the database used contains 7390 isolated digits for training set, and 3035 digits for testing sets	94.14%,
Dibeklioglu, (2007).	Digit Recognition Using HMM and MLP	Pen-Based handwritten(Online)	MLP = 70.5% HMM = 62% MLP+HMM = 77.9%
Shilbayeh, Raho, & Alkhateeb, (2009).	Recognizing of Hindi digits using FCC with eight connectivity	Online Handwritten using a mouse	89.5%
Naser, (2010).	Recognition of Hindi digits using Neural Networks	Online handwritten using a mouse	91%
Alkhatib, (2011)	Offline Hindi digit recognition using MLP	Offline handwritten	74.4%

Table 5.1: Recognition of character and digit using different approaches and getting different results

The table 5.1 shows some comparative results of different approaches and different related works for the recognition of isolated digits or characters. These results will be discussed and compared with the Offline Handwritten Hindi Digit recognition subsystem. Firstly, in Nawas, Sarfraz, Zidouri & Al-Khatib, (2003), they used the RBF networks for recognizing Arabic Characters. RBF is Radial-Basis Function Networks. In implementing their function they used the Brain Construction Kit (BCK) tool. BCK is a Java Package that enables users to create, train, modify, and use Artificial Neural Networks. They obtained a recognition rate of 76%. On the other hand, the designed subsystem is different in approach, it used the MLP architecture using the NN tool box in matlab, it was designed to recognize Hindi digits, and it obtained a recognition rate of 74.4%. Second, in Sadri, Suen, & Bui, (2003), a system for offline handwritten Arabic and Persian digits recognition is designed using Support Vector Machines (SVM). They achieved a high recognition rate. Such rate was due to the use of the CENPARMI Indian digit database. This database contains 7390 isolated digits for training set, and 3035 digits for testing sets, and all these digits have been collected from real bank checks. On the other side, the database used in the first subsystem in this thesis is very small compared to the CENPARMI database. It contained 1000 samples for training and 600 samples for testing. Third, in Dibeklioglu, (2007), they recognized online handwritten digits written by a pen. They used for recognition the combination of the HMM and MLP to get higher rate than using an individual approach, the results shown in table 5.1. With comparison by the designed subsystem, it recognized offline digits entered by a scanner, and the approach used was the MLP without combination with other approaches. Another work was in Shilbayeh, Raho & Alkhateeb, (2009), they recognized online Hindi digits drawn by a mouse. They used the Freeman Chain Code (FCC) with eight connectivity and used templates to recognize the digits. Also, in Naser, (2010) they designed a system to recognize online Hindi digits using Neural Networks. In contrast with the subsystem designed in this thesis, it recognized offline Hindi digits. The difference between online and offline should be mentioned here. In online, the digits are written directly on the screen and saved as an image. In online there is no noise and no size problems since the user determines the handwriting space. Also less preprocessing problems appear. In offline, the digits are written on papers, the papers entered by a scanner, noise problems result from scanners and size problems appeared since the size will be very large and resizing images may result in loosing pixels and changing the shape of the digit. Also many other preprocessing problems appeared. All these problems affect the efficiency of the recognition systems and explain the differences in recognition rates achieved.

5.1.2 The comparative results with respect to the Courtesy Amount Recognition subsystem and similar applications such as postal codes and other recognition of chains of digits:

Name	Approach	Application used	Recognition Rate
Dzuba., Filatov & Volgunin, (1997).	Handwritten ZIP Code Recognition.	Postal Address	73%
Blumenstein & Verma, (1998).	Recognition of postal addresses using Neural Networks	Postal Addresses	Handwritten Charatcers: 54.9% Handwritten Postal Addresses: 56.71%
Ouchtati, Bedda, & Lachouri, (2007).	Segmentation and Recognition of Handwritten Numeric Chains	Handwritten Numeric Chains	91.3%
Morshed, Shahidul Islam, Abul Faisal, Abdus Sattar & Mitsuru Ishizuka, (2007).	Recognizing Handwritten Postal codes using Neural Network Architechtures	Postal codes	92.2%
Palacios & Gupta, (2008).	A system for Processing bank checks automatically	Bank checks	MLP: 79.9% MLP-DDD:85.2% Arbiter: 86.8% After structural Post Processing: 84%
Alkhatib, (2011)	Recognizing the courtesy amount in Arabian bank checks	Bank checks	MLP: 74.4%

Table 5.2: Recognition Rates of different application of digit recognition

Table 5.2 shows some of the comparative results to the Courtesy Amount recognition subsystem. First, in Dzuba, Filatov & Volgunin, (1997), they implemented a system for handwritten zip code recognition. The system has two stages, feature extraction and recognition, by matching the input phrase feature representation to all lexicon entry representations. They achieved 73% recognition rate. Their system compared with the designed system doesn't recognize digits using Neural network approach therefore they didn't need to segment the zip code into segments. Second, in Blumenstein & Verma, (1998), a segmentation algorithm is used to prepare raw training data for use with an Artificial Neural Network for Recognition of Real World Postal Addresses; their system recognized both characters and digits since the postal code may contain words and numbers. Their system works for both printed and handwritten characters and digits. In contrast with the designed system, the main concern is about handwritten digits. The system also was experimented on single characters, then on postal addresses. And they got results for both. They got a recognition rate of 54.9% for single characters. And a recognition rate of 56.71% for postal addresses. Also, in Ouchtati, Bedda & Lachouri, (2007), they introduced a system for segmentation and recognition of handwritten numeric chains. They segmented the chains, and then recognized the digits separately. They obtained a recognition rate of 91.3%. This work is very close to the designed work; however, they used different methods in preprocessing, feature extraction, and segmentation. Another work is in Morshed, Shahidul Islam, Abul Faisal, Abdus Sattar & Mitsuru Ishizuka, (2007), they develop neural-network architecture for recognizing handwritten Bangla digits and its application to automatic mail sorting machine for Bangladesh Postal system is presented. The performance of the network on single digits is 92.2%. The good accuracy they got may be due to the number of samples they used in training and testing. The samples were very large compared to the ones used in the designed system in this thesis. They used 5500 samples for training and 4900 samples for testing. Finally, in Palacios & Gupta, (2008), they designed a System for Processing Handwritten Bank Checks Automatically. In the recognition stage they used 3103 segments for training and 1444 for testing. They used two MLP networks and two MLP-DDD networks to increase the accuracy, and then they used another module called arbiter that analyzes the information and produces the output. They run multiple networks to have better accuracy rate. The results shown in table 5.2 for each network and at the end after using the arbiter and running multiple networks the rate becomes 84%. Compared with the designed system, that uses only one MLP network for recognizing separate digits and giving a rate of 74.4%.

5.2 Conclusion

In this thesis an HHCA recognition system was designed and implemented for the recognition of the courtesy amount of Arabian bank checks. The system consists of two subsystems. The first subsystem recognizes offline handwritten Hindi digits and the second subsystem recognizes the courtesy amount of Arabian bank checks. Both subsystems are related to each other where the second subsystem calls the first subsystem for Hindi digit recognition. The designed system satisfies the goal of recognizing the string of handwritten Hindi digits in the courtesy amount after passing through the preprocessing, segmentation, feature extraction, and recognition stages successfully.

Many achievements were obtained during the stages of the overall system. The first achievement was after designing and implementing the Offline Handwritten Hindi Digit Recognition subsystem. This system achieved successfully the purpose of recognizing separate Hindi digits with a good recognition rate of 74.4%. Other achievements were gotten after designing and implementing the Courtesy Amount Recognition subsystem. The system achieved two successful algorithms for segmentation and feature extraction. The segmentation algorithm performed successfully the operation of segmenting a string of digits or characters. Also the feature extraction algorithm performed successfully the operation of extracting and splitting digits or characters from the string containing them. Both algorithms can be used not only for offline handwritten Hindi digits, but also in other types of writings, printed, or online and on different types of digits and characters. This characteristic allows these algorithms to be applied on different applications other than the bank checks recognition but also on postal code and car plate recognition.

5.3 Future work and Recommendations:

As a recommendation for the designed system to be more efficient and achieve higher recognition rate, many suggestions can be followed. The system can be more efficient if the different stages were enhanced. For example, in the segmentation stage if the problem of touching components was solved, the system will be more efficient. Also for increasing the recognition rate, the samples of the training and testing should be increased. For a future work, the verification of Arabian bank checks can be done by recognizing other handwritten fields such as the legal amount, name, date, and signature. Also, the recognition of offline Hindi characters can be another work in future. Also other applications of offline handwritten Hindi digits recognition can be added to the future work such as postal codes and car plates recognition.

References

1. Anisimov, V., Gorski, N., Price, D., Barct, O. & Knerr, S. (1999), Bank Check Reading: Recognizing the Courtesy Amount, ***Image Analysis Applications and Computer Graphics***, Springer Berlin, Heidelberg,pp.161-172,(Online), available: http://www.springerlink.com/content/y875120347145t08.

2. Blumenstein, M. & Verma, B., (1998), A Segmentation Algorithm used in Conjunction with Artificial Neural Networks for the Recognition of Real World Postal Addresses, ***IEEE International Conference on Neural Networks,*** (online), available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.7472

3. Chakravarty, A. & Day, W. (200?), "Handwritten Digit Classification", (Online), available: http://www.citeseerx.ist.psu.edu.

4. Demuth, H., Beale, M. & Hagan, M. (2007), *Neural network toolbox 5 user's guide*, The MathWorks, Inc, U.S.

5. Dibeklioglu, H. (2007), Digit Recognition Using HMM and MLP, ***Informatics Institute, University of Amsterdam***, pp 1-6.

6. Dilecce, V., Dimauro, G., Guerriero, A., Impedovo, S., Pirlo, G. & Salzo, A. (2000), A New Hybrid Approach For Legal Amount Recognition, ***Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition, September 11-13 2000, Amsterdam***, International Unipen Foundation, pp 199-208

7. Dzuba, G., Filatov, A., Volgunin, A., (1997), Handwritten ZIP Code Recognition, ***Fourth International Conference Document Analysis and Recognition,*** pp.766.

8. Farah, N., Souici, L., Farah, L., & Sellami, M. (2004). Arabic Words Recognition with Classifiers Combination: An Application to Literal Amounts, ***Artificial Intelligence: Methodology, Systems, and Applications***, Springer Berlin, Heidelberg pp. 420–429.

9. Freitas, C.O.D., Yacoubi, A., Bortolozzi, F. & Sabourin, R. (2000), Brazilian Bank Check Handwritten Legal Amount Recognition, pp. 97-104.

10. Gorski, N., Anisimov, V., Augustin, E., Baret, O. & Maximov S. (2000),Industrial bank check processing: the A2iA CheckReaderTM ***International Journal On Document Analysis and Recognition***, Vol. 3, No. 3, pp. 196-206.

11. Haboubi, S. & Maddouri, S.S. (2005), "Extraction Of Handwritten Areas From Colored Image Of Banck Check By an Hybrid Method", (Online), available: http://hal.archivesouvertes.fr/docs/00/07/92/43/PDF/acidca_sofiene_samiamaddouri.pdf.

12. Jacobs, R. (2008), *Backpropagation Algorithm*, Department of Brain & Cognitive Sciences University of Rochester, Rochester, NY 14627, USA.

13. Khan, S.A. (1998), *Character Segmentation Heuristics for Check Amount Verification*,(Maters Thesis), Massachusetts Institute of Technology. (Online), available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.27.4503.

14. Karthikeyan, S. (2008), *Online Handwritten Character recognition for Tamil and Devanagari*, (Bachelor of technology), Indian Institute Of Technology, Madras.

15. Lorigo, L.M. & Govindaraju, V. (2006), Off-line Arabic Handwriting Recognition: A Survey, ***IEEE Transaction On Pattern Analysis And Machine Intelligence***. Vol. 28, no. 5, pp.712-724

16. Mahmoud, S.A. & Awaida, S.M. (2009), Recognition Of Off-Line Handwritten Arabic (Indian) Numeral Using Multi-scale Features And Support Vector Machines VS. Hidden Markov Models, ***The Arabian Journal for Science and Engineering.***Vol. 34, No. 2B

17. Morshed, A.S.M., Shahidul Islam, M.d., Abul Faisal, M.d., Abdus Sattar, M.d. & Mitsuru Ishizuka, M. (2007), Automatic Sorting Of Mails By Recognizing Handwritten Postal Codes Using Neural Networks Architectures, (Online) available:
http://www.miv.t.u-tokyo.ac.jp/papers/morshed-ICCIT07.pdf

18. Naser, M. (2010). *Mouse gestural hindi digits recognition using neural network*, (Unpublished Master Thesis), Middle East University, Amman, Jordan.

19. Nawas, S.N., Sarfraz, M., Zidouri, A., & Al-Khatib, W.G. (2003), An Approach to Offline Arabic Character Recognition Using Neural Networks, ***IEEE International Conference on Electronics, Circuits, and Systems (ICECS),*** Vol. 3, pp. 1328 – 1331.

20. Omari, S.A., Sumari P., Al-Taweel, S.A. & Husain, J.A. (2009), Digital Recognition using Neural Network, ***Journal of computer science***. Vol. 5, no. 6, pp. 427-434.

21. Otsu, N., (1979), A threshold selection method from gray-level histograms. ***IEEE Trans. Sys., Man., Cyber***. Vol. 9, pp. 62–66, (online), available:

http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4310076

22. Ouchtati, S., Bedda, M., & Lachouri, A., (2007), Segmentation and Recognition of Handwritten Numeric Chains, ***Journal of Computer Science,*** Vol. 4, pp. 242-248.

23. Palacios, R., Sinha, A. & Gupta, A. (2002), Automatic Processing of Brazilian Bank Checks, ***Massachusetts Institute of Technology,*** (online), available: www.iit.upcomillas.es/docs/02RPH.05.pdf

24. Palacios, R., Gupta, A. & Wang, P.S. (2003). Handwritten Bank Check Recognition Of Courtesy Amounts, ***International Journal of Image and Graphics***, Vol. 4, No. 2, pp. 1-20.

25. Palacios, R & Gupta, A. (2008), A System For Processing Handwritten Bank Checks Automatically, ***Image and Vision Computing***, Vol. 26, No. 10, pp. 1297-1313.

26. Pan, W.M, Bui, T.D. & Suen, C.Y. (2009), Isolated Handwritten Farsi numerals Recognition Using Sparse And Over-Complete Representations, ***Center for Pattern Recognition and Machine Intelligence, Concordia University***, IEEE Computer Society, Barcelona, Spain, pp.586-590.

27. Parisi, R., Claudio, E.D.Di., Lucarelli, G., & Orlandi, G. (1998), Car Plate Recognition By Neural Networks and Image Processing, ***University of Rome "La Sapienza".*** (Online) available: www. infocom.uniroma1.it/~parisi/papers/iscas98.pdf

28. Sadri, Suen, & Bui, (2003), Application of Support Vector Machines for Recognition of Handwritten Arabic/Persian Digits, ***Centre for Pattern Recognition and Machine Intelligence (CENPARMI),***Vol. 1.

29. Shilbayeh, N.F, & Iskandarani, M.Z. (2005), An intelligent multilingual mouse gesture recognition system, ***Journal of Computer Science***, Vol. 1, No.3, pp: 346-350.

30. Shilbayeh, N.F & Iskandarani, M.Z. (2008), Effect of Hidden Layer Neurons on the Classification of OCR Typed Arabic Numerals, ***Journal of Computer Science***, Vol. 4, No. 7, pp: 578-584.

31. Shilbayeh, N.F, Raho, G. & Alkhateeb, M. (2009). An Efficient Structural Mouse Gesture Approach for Recognizing Hindi Digits, ***Journal of Applied Science***, Vol. 9, No. 19, pp: 3469-3479.

32. Sivanandam, S.N, Sumathi, S. & Deepa, S.N. (2006), *Introduction to neural networks using matlab 6.0*, Tata McGraw, New Delhi.