

# **AUTOMATED COIN RECOGNITION SYSTEM USING ANN**

*Thesis submitted in partial fulfillment of the requirements for the award  
of degree of*

**Master of Engineering**  
in  
**Software Engineering**

*Submitted By*  
**Shatrughan Modi**  
**(800931019)**

Under the supervision of:  
**Dr. Seema Bawa**  
Professor



**COMPUTER SCIENCE AND ENGINEERING DEPARTMENT  
THAPAR UNIVERSITY  
PATIALA – 147004**

**June 2011**

## Certificate

---

I hereby certify that the work which is being presented in the thesis entitled, "Automated Coin Recognition System using ANN", in partial fulfillment of the requirements for the award of degree of *Master of Engineering in Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Seema Bawa* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.


  
(Shatrughan Modi)


This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

  
(Dr. Seema Bawa)  
20/06/2011

Professor,  
CSED

Countersigned by

  
(Dr. Maninder Singh)  
Head 9316161  
Computer Science and Engineering Department  
Thapar University  
Patiala

  
(Dr. S. K. Mohapatra)  
Dean (Academic Affairs)  
Thapar University  
Patiala

## Acknowledgement

---

First of all I would like to thank the Almighty, who has always guided me to work on the right path of the life.

This work would not have been possible without the encouragement and able guidance of my supervisor Dr. Seema Bawa. I thank my supervisors for their time, patience, discussions and valuable comments. Their enthusiasm and optimism made this experience both rewarding and enjoyable.

I am equally grateful to Dr. Maninder Singh, Associate Professor & Head, Computer Science & Engineering Department, a nice person, an excellent teacher and a well – credited researcher, who always encouraged me to keep going with work and always advised me with his invaluable suggestions.

I will be failing in my duty if I don't express my gratitude to Dr. Abhijit Mukherjee, Director of the University, for making provisions of infrastructure such as library facilities, computer labs equipped with net facilities, immensely useful for the learners to equip themselves with the latest in the field.

I am also thankful to the entire faculty and staff members of Computer Science and Engineering Department for their direct-indirect help, cooperation, love and affection, which made my stay at Thapar University memorable.

Last but not least, I would like to thank my family whom I dearly miss and without whose blessings none of this would have been possible. To my parents, I own thanks for their wonderful love and encouragement. I would also like to thank my sister, since she insisted that I should do so. I would also like to thank my close friends for their constant support.

## Abstract

---

Coins have been the integral part of our day to day life. We use coins almost everywhere like in grocery stores, banks, trains, buses etc. So it is a basic need that coins can be recognized, counted, sorted automatically. There are three types of coin recognition systems available in market based on different recognition methods: Mechanical method based systems, Electromagnetic method based systems and Image processing based systems. Mechanical and Electromagnetic method based systems recognize coins based on their magnetism and physical dimensions like radius, thickness etc. Due to which these systems can be fooled easily by fake coins that have same properties as original coins. So, this thesis mainly focuses on Image processing method based systems.

We have developed an ANN (Artificial Neural Network) based Automated Coin Recognition System for Indian coins of denominations `1, `2, `5, `10 using MATLAB 7.11.0 (R2010b). This system takes as input a RGB coin image of either side *i.e.* whether head or tail. Then, features are extracted from image using Hough Transform for circle detection, Pattern Averaging, Cropping, Trimming etc. Then the extracted features are passed as input to trained Neural Network. Training of the neural network has been done by randomly selected 5040 sample images of coins rotated at  $5^{\circ}$ ,  $10^{\circ}$ ,  $15^{\circ}$ , ...,  $355^{\circ}$ . In this thesis we have created two Neural Networks: One takes 400 features as input, name it as coin\_neural\_400, and other takes 100 features as input, name it as coin\_neural\_100. Recognition rate of 81.37% has been achieved during the experiments with coin\_neural\_100 whereas coin\_neural\_400 shows drastic improvement and gives recognition rate of 97.74% *i.e.* there is only 2.26% miss-recognition. This system is capable of recognizing coins from both sides rotated at any degree.

## Table of Contents

Sr. No.	Topic	Page No.
	<b>Certificate</b>	<b>i</b>
	<b>Acknowledgement</b>	<b>ii</b>
	<b>Abstract</b>	<b>iii</b>
	<b>Table of Contents</b>	<b>iv</b>
	<b>List of Figures</b>	<b>vi</b>
	<b>List of Tables</b>	<b>viii</b>
<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Neural Networks	2
1.1.1	Why to use Neural Networks?	2
1.1.2	Human and Artificial Neurons	3
1.1.3	Architecture of Neural Networks	4
1.1.3.1	Feed-forward Network	4
1.1.3.2	Feedback Networks	5
1.1.4	Network Layers	5
1.1.5	Back Propagation Model	6
1.1.6	Application Area	6
1.1.6.1	Application in Image Processing	6
1.1.6.2	Business Application	7
1.1.6.3	Application of Feed-forward Neural Network	8
1.2	Image Types	8
1.3	Resolution	9
1.4	Color Space	10
1.4.1	RGB Color Model	10
1.4.2	HSV Color Model	10
1.5	Image Processing	11
1.5.1	Sobel Edge Detection	12
1.5.2	Histogram	12
1.5.3	Segmentation	13
1.6	Thesis Outline	13
<b>2</b>	<b>Literature Review</b>	<b>14</b>
2.1	Approaches for Modern Coins	14
2.2	Approaches for Ancient Coins	18
2.3	Approaches for both Modern and Ancient Coins	19
2.4	Comparison	20
<b>3</b>	<b>Problem Statement</b>	<b>23</b>
3.1	Gap Analysis	23
3.2	Problem Statement	23
3.3	Objectives	23
3.4	Methodology	24

<b>4</b>	<b>Implementation Details</b>	<b>25</b>
4.1	What is MATLAB?	25
4.2	Architecture for ANN based Automated Coin Recognition System	26
4.2.1	Acquire RGB Coin Image	26
4.2.2	Convert RGB Coin Image to Grayscale	27
4.2.3	Remove Shadow of Coin from Image	28
4.2.4	Crop and Trim the Image	28
4.2.5	Generate Pattern Averaged Image	29
4.2.6	Generate Feature Vector	30
4.2.7	Give Feature Vector as Input to Trained NN	30
4.2.7.1	Introduction to the GUI	30
4.2.7.2	Selecting the Wizard according to the Problem Type	30
4.2.7.3	Select Input and Target	31
4.2.7.4	Select Validation and Test Data	32
4.2.7.5	Select the Number of Hidden Neurons	33
4.2.7.6	Train the Network	33
4.2.7.7	Evaluate Network	35
4.2.7.8	Save Results	35
4.2.8	Give Appropriate Result according to the Output of NN	36
4.3	ANN based Automated Coin Recognition System	36
<b>5</b>	<b>Experimental Results</b>	<b>37</b>
5.1	Training and Testing Data	37
5.2	Neural Network Architecture	37
5.3	Results of coin_neural_400	38
5.4	Results of coin_neural_100	39
5.5	Comparison	41
<b>6</b>	<b>Conclusion and Future Scope</b>	<b>42</b>
6.1	Conclusion	42
6.2	Future Scope	42
	<b>References</b>	<b>43</b>
	<b>List of Papers Communicated</b>	<b>46</b>

## List of Figures

<b>Fig. No.</b>	<b>Figure Name</b>	<b>Page No.</b>
1.1	Components of Neurons	3
1.2	The Synapse	3
1.3	A Simple Neuron	4
1.4	Feed- forward Neural Network	5
1.5	Applications of Neural Network	7
1.6	Binary Image	8
1.7	True Color Image	9
1.8	Grayscale Image	9
1.9	RGB Coordinate System	10
1.10	RGB Color Model	10
1.11	HSV Coordinate System	11
1.12	HSV Color Model	11
1.13	Original Image	12
1.14	Image after Sobel Edge Detection	12
4.1	Architecture of ANN based Automated Coin Recognition System	26
4.2	Indian Coins of different Denominations	27
4.3	24-bit RGB Coin Image	27
4.4	8-bit Grayscale Coin Image	27
4.5	Grayscale Coin Image	28
4.6	Coin Image without Shadow	28
4.7	Cropped Coin Image	29
4.8	100×100 Trimmed Coin Image	29
4.9	50×50 Trimmed Coin Image	29
4.10	Neural Network Start Window	31
4.11	Neural Network Pattern Recognition Tool Window	31
4.12	Input / Target Data Window	32
4.13	Validation and Test Data Window	32
4.14	Network Architecture Window	33
4.15	Train Network Window	34
4.16	Neural Network Training Window	34
4.17	Evaluate Network Window	35
4.18	Save Results Window	35
4.19	ANN based Automated Coin Recognition System	36
5.1	Network diagram for 400 features (coin_neural_400)	37
5.2	Network diagram for 100 features (coin_neural_100)	38
5.3	Results after Training, Testing and Validation of coin_neural_400	38

5.4	Performance of Network coin_neural_400	38
5.5	Confusion Matrix for coin_neural_400	39
5.6	Results after Training, Testing and Validation of coin_neural_100	39
5.7	Performance of Network coin_neural_100	40
5.8	Confusion Matrix for coin_neural_100	40

## List of Tables

---

<b>Table No.</b>	<b>Table Name</b>	<b>Page No.</b>
1.1	Types of Images Based on Bid Depth	9
2.1	Comparison of Various Existing Techniques	20
5.1	Comparison of Networks	41

# Chapter 1

## Introduction

---

We can not imagine our life without coins. We use coins in our daily life almost everywhere like in banks, supermarkets, grocery stores etc. They have been the integral part of our day to day life. So there is basic need of highly accurate and efficient automatic coin recognition system. In spite of daily uses coin recognition systems can also be used for the research purpose by the institutes or organizations that deal with the ancient coins. There are three types of coin recognition systems available in the market based on different methods:

- Mechanical method based systems
- Electromagnetic method based systems
- Image processing based systems

The mechanical method based systems use parameters like diameter or radius, thickness, weight and magnetism of the coin to differentiate between the coins. But these parameters can not be used to differentiate between the different materials of the coins. It means if we provide two coins one original and other fake having same diameter, thickness, weight and magnetism but with different materials to mechanical method based coin recognition system then it will treat both the coins as original coin so these systems can be fooled easily.

The electromagnetic method based systems can differentiate between different materials because in these systems the coins are passed through an oscillating coil at a certain frequency and different materials bring different changes in the amplitude and direction of frequency. So these changes and the other parameters like diameter, thickness, weight and magnetism can be used to differentiate between coins. The electromagnetic method based coin recognition systems improve the accuracy of recognition but still they can be fooled by some game coins.

In the recent years coin recognition systems based on images have also come into picture. In these systems first of all the image of the coin to be recognized is taken either by camera or by some scanning. Then these images are processed by using various techniques of image processing like FFT, DCT, edge detection, segmentation etc and various features are extracted from the images. Then based on

these features different coins are recognized. We will mainly focus on image processing based coin recognition systems. But first let's discuss about some topics like neural networks and image processing, which we are going to use in our system for implementation.

## **1.1 Neural Network**

Human brain performs much better than any other AI system for tasks such as face recognition, speech recognition, pattern recognition etc. Neural networks is emerging as a field of study within AI and engineering via the collaborative efforts of engineers, physicists, mathematicians, computer scientists, and neuroscientists. Although the elements of research are many, there is a basic underlying focus on pattern recognition and pattern generation, embedded within an overall focus on network architectures. An **Artificial Neural Network (ANN)**, usually called **Neural Network (NN)**, is a mathematical model or computational model that is inspired by the structure and functional aspects of biological neural networks such as human brain. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well. By configuring virtual neural networks that function like the human brain, computers can perform tasks at greater speeds and with increased flexibility of application.

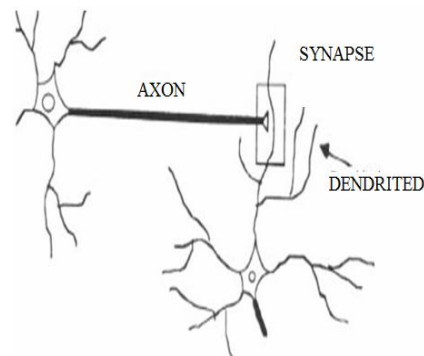
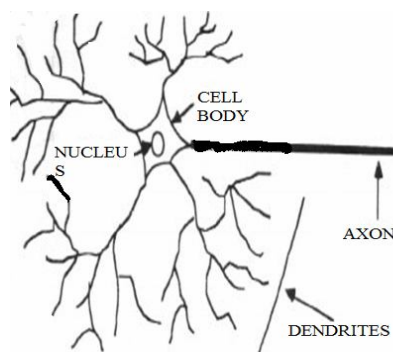
### **1.1.1 Why to use Neural Networks?**

Neural Networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This "expert" can then be used to provide projections given new situations of interest and answer "what if" questions. Other advantages include:

- **Adaptive learning:** An ability to learn how to do tasks based on the data given for training or initial experience.
- **Self-Organization:** An ANN can create its own organization or representation of the information it receives during learning time.
- **Real Time Operation:** ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
- **Fault Tolerance via Redundant Information Coding:** Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

### 1.1.2 Human and Artificial Neurons

Much is still unknown about how the brain trains itself to process information, so theories abound. In the human brain, a typical neuron collects signals from others through a host of fine structures called dendrites. The neuron sends out spikes of electrical activity through a long, thin stand known as an axon, which splits into thousands of branches as shown in Figure 1.1. At the end of each branch, a structure called a synapse (Figure 1.2) converts the activity from the axon into electrical effects that inhibit or excite activity from the axon into electrical effects that inhibit or excite activity in the connected neurons. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes.

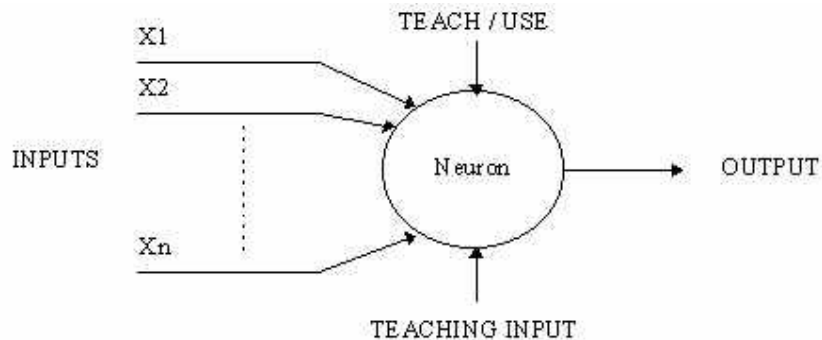


**Figure 1.1: Components of a Neuron [23]**      **Figure 1.2: The Synapse [23]**

This is not to say that localization does not exist in the brain. Neurons in the superior temporal of the cerebral cortex, for example, respond selectively to faces. But there is

no "grandmother cell", *i.e.*, no cell that responds specifically to the face of someone's grandmother. Instead, each neuron has a different response pattern to a set of faces. Ensembles of neurons encode the response to identify a particular face. And an overlapping ensemble may identify another face. A very real difficulty of correlating artificial neural networks with biological ones lies in the way weights are modified in the former and synaptic strengths are modified in the latter. Weights are altered mathematically in a computer network, based on differences in values. Synaptic strengths, on the other hand, are modified in response to synaptic activity.

The back propagation model, in particular, is held to be biologically unrealistic insofar as it would require a supervisor and a violation of the unidirectional flow of information seen in axons. Some researchers have postulated parallel, backward-directed axons to return error information, but the modification of synaptic strength by these axons is still very hypothetical. These neural networks first try to deduce the essential features of neurons and their interconnections. We then typically program a computer to simulate these features. However because our knowledge of neurons is incomplete and our computing power is limited, our models are necessarily gross idealizations of real networks of neurons.



**Figure 1.3: A Simple Neuron [23]**

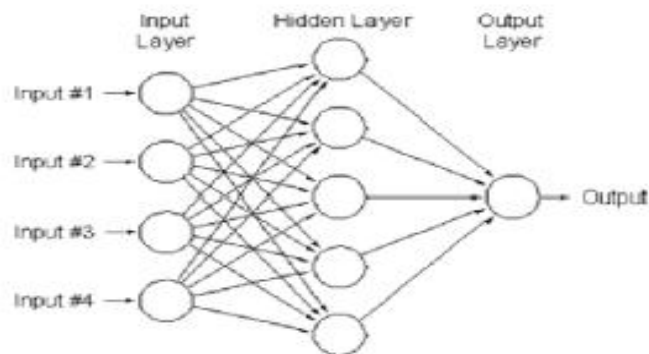
### 1.1.3 Architecture of neural networks

There are two types of architectures for neural networks as below:

#### 1.1.3.1 Feed-forward Networks

Feed-forward ANNs allow signals to travel one way only; from input to output. There is no feedback (loops) *i.e.* the output of any layer does not affect that same layer. Feed forward ANNs tend to be straight forward networks that associate inputs with outputs.

They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up or top-down.



**Figure 1.4: Feed-forward Neural Network [24]**

### 1.1.3.2 Feedback networks

Feedback networks can have signals traveling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single layer organizations.

### 1.1.4 Network Layers

The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of "**input**" units is connected to a layer of "**hidden**" units, which is connected to a layer of "**output**" units.

- The activity of the input units represents the raw information that is fed into the network.
- The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.
- The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and

hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents. The single-layer organization, in which all units are connected to one another, constitutes the most general case and is of more potential computational power than hierarchically structured multi-layer organizations. In multilayer networks, units are often numbered by layer, instead of following a global numbering.

### **1.1.5 Back Propagation Model**

Back propagation model is a model which can solve many more difficult problems. Back propagation has proven to be so powerful that it currently accounts for 80% of all neural network applications. In Back propagation, a third neurons layer is added (the hidden layer) and the discrete thresholding function is replaced with a continuous (sigmoid) one. But the most important modification for Back propagation is the generalized delta rule, which allows for adjustment of weights leading to the hidden layer neurons in addition to the usual adjustments to the weights leading to the output layer neurons.

### **1.1.6 Application Area**

The excitement in neural network started mainly due to difficulties in dealing with problem in the field of speech, image, natural language and decision making using know method of pattern recognition and artificial intelligence. Several of these problems have been attempted using the principle of neural networks [24].

#### **1.1.6.1 Application in Image Processing**

An image is represented as a two dimensional array of pixels, with some gray values or color associated with each pixel. Characteristics of an image are: (a) the local structure, dictated by the spatial correlations among nearby pixels, and (b) the global structure, handing over the semantics of the image. These local and global features are used in interpreting an image for recognition. Standard neural network models accept the input data in an unstructured manner, in the sense that the input to each unit in the input layer is considered independent. Thus when an image is fed as an input to a neural network the gray value of each pixel is provided as input, and the input units have no spatial structure reflecting the spatial correlation among the pixel values. Before feeding an image to a network, the image is size-normalized, since the

dimensionality of the input to a network is fixed. In some cases like handwriting, the normalization may be carried out at world level, in which case the size, slope and position variation of the individual characters will cause difficulty for recognition by the neural network [24].

### 1.1.6.2 Business Applications

- **Banking:** Check and other document reading, credit application evaluation.
- **Credit Card Activity Checking:** Spot unusual credit card activity that might possibly be associated with loss of a credit card.
- **Electronics:** Code sequence prediction, integrated circuit chip layout, process control, chip failure analysis, machine vision, voice synthesis, nonlinear modeling.
- **Medical:** Breast cancer cell analysis, EEG and ECG analysis, prosthesis design, optimization of transplant times, hospital expense reduction, hospital quality improvement, emergency-room test advisement.
- **Robotics:** Trajectory control, forklift robot, manipulator controllers, vision systems. **Speech:** Speech recognition, speech compression, vowel classification, text-to-speech synthesis.
- **Telecommunications:** Image and data compression, automated information services, real-time translation of spoken language, customer payment processing systems.

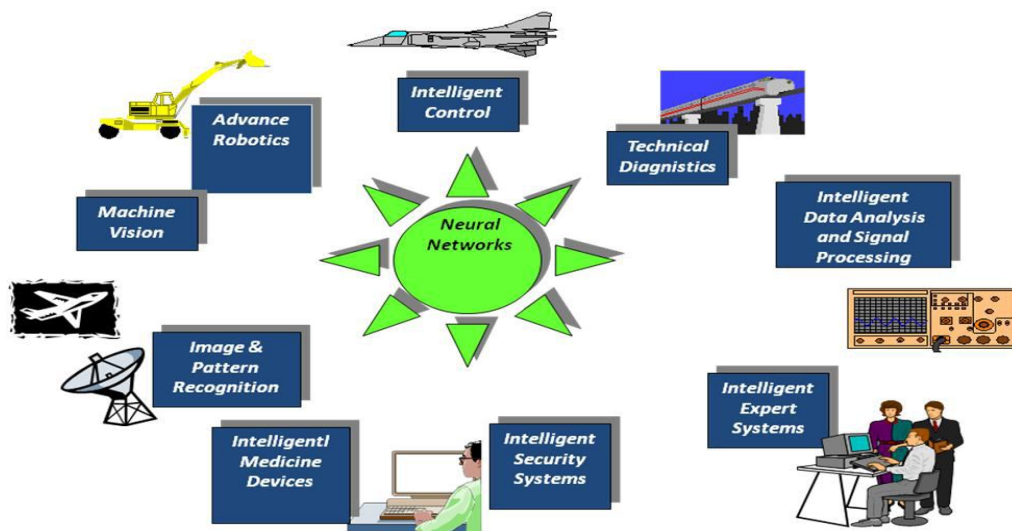


Figure 1.5: Application of Neural Networks [25]

### 1.1.6.3 Application of Feed-forward Neural Networks

Multilayered neural network trained using the BP algorithm account for a majority of application of neural network to real world problem. This is because BP is easy to implement and fast and efficient to operate. These include application domains such as: astronomy; automatic target recognition; handwritten digit string recognition; control; sonar target classification; software engineering project management; and countless others.

## 1.2 Image Types

There are three type of image, which is described below [26]:

- **Binary Image:** A binary image is a logical array of 0s and 1s. Pixels with the value 0 are displayed as black; pixels with the value 1 are displayed as white as shown in Figure 1.6.
- **Grayscale Image:** It is also known as an intensity, gray scale, or gray level image. Array of class uint8, uint16, int16, single, or double whose pixel values specify intensity values. For single or double arrays, values range from [0, 1]. For uint8, values range from [0,255]. For uint16, values range from [0, 65535]. For int16, values range from [-32768, 32767]. Figure 1.8 is an example of grayscale image.
- **True Color Image:** It is also known as an RGB image. A true color image is an image in which each pixel is specified by three values, one each for the red, blue, and green components of the pixel' scalar as shown in Figure 1.7. M-by-n-by-3 array of class uint8, uint16, single, or double whose pixel values specify intensity values. For single or double arrays, values range from [0, 1]. For uint8, values range from [0, 255]. For uint16, values range from [0, 65535].

1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	0	0	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	0	0	0	0	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	0	0	0	1	1	0	0	0	1
1	1	1	1	1	1	1	1	1	1

Figure 1.6: Binary Image [27]



**Figure 1.7: True Color Image [28] Figure 1.8: Grayscale Image [28]**

The Table 1.1 illustrates different image types in terms of bits (bit depth), total colors available, and common names.

**Table 1.1: Types of Images Based on Bid Depth [29]**

Bits per Pixel	Number of Colors Available	Common Name(s)
1	2	Monochrome
2	4	CGA
4	16	EGA
8	256	VGA
16	65536	XGA, High Color
24	16777216	SVGA, True Color
32	16777216 + Transparency	
48	281 Trillion	

### 1.3 Resolution

The images are divided into N rows and M columns. The region of intersection of a row and a Column is known as pixel. The value assigned to each pixel is the average brightness of the regions. The position of each pixel is represented by a pair of coordinates (x, y). The resolution of a digital image is presented in the form of number of columns  $\times$  number of rows. For example, an image with a resolution of 640 $\times$ 480 means that it display 640 pixels on each of the 480 rows. Some other common resolution used is 800 $\times$ 600 and 1024 $\times$ 728.

Resolution is one of most commonly used ways to describe the image quality of digital camera or other optical equipment. The resolution of a display system or

printing equipment is often expressed in number of dots per inch. For example, the resolution of a display system is 72 dots per inch (dpi) or dots per cm.

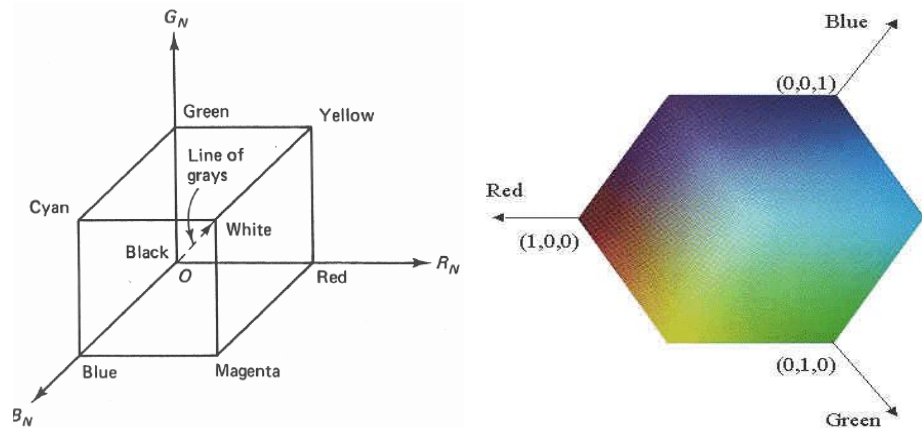
## 1.4 Color Space

A color space is defined as a model for representing color in terms of intensity values. Typically, a color space defines a one- to four-dimensional space. A color component, or a color channel, is one of the dimensions. A color dimensional space (*i.e.* one dimension per pixel) represents the gray-scale space. The following two models are commonly used in color image retrieval system [30]:

- RGB Color Model
- HSV Color Model

### 1.4.1 RGB Color Model

The RGB color model is composed of the primary colors Red, Green, and Blue. This system defines the color model that is used in most color CRT monitors and color raster graphics. They are considered the "additive primaries" since the colors are added together to produce the desired color. The RGB model uses the Cartesian coordinate system as shown in Figure 1.9. Notice the diagonal from (0, 0, 0) black to (1, 1, 1) white which represents the grey-scale. Figure 1.10 is a view of the RGB color model looking down from "White" to origin [30].



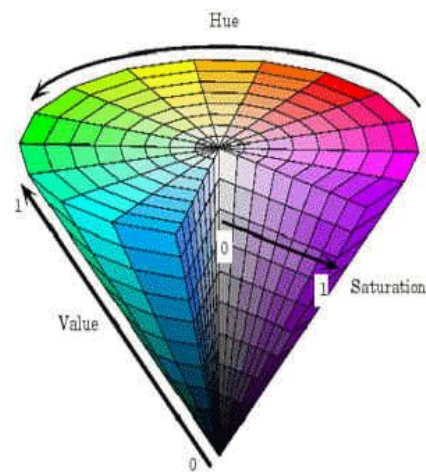
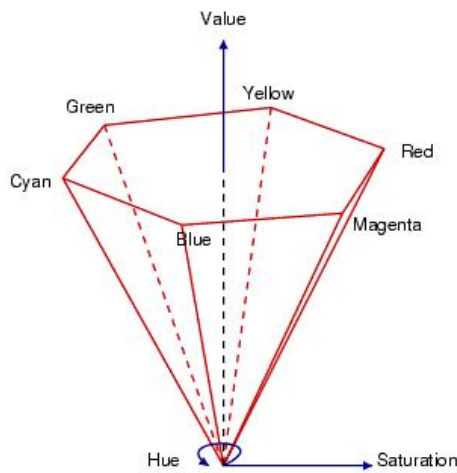
**Figure 1.9: RGB Coordinates System [30] Figure 1.10: RGB Color Model [30]**

### 1.4.2 HSV Color Model

The HSV stands for the Hue, Saturation, and Value based on the artists (Tint, Shade, and Tone). The coordinate system is a hexa-cone in Figure 1.11. And Figure 1.12 a

view of the HSV color model. The Value represents intensity of a color, which is decoupled from the color information in the represented image. The hue and saturation components are intimately related to the way human eye perceives color resulting in image processing algorithms with physiological basis [30].

As hue varies from 0 to 1.0, the corresponding colors vary from red, through yellow, green, cyan, blue, and magenta, back to red, so that there are actually red values both at 0 and 1.0. As saturation varies from 0 to 1.0, the corresponding colors (hues) vary from unsaturated (shades of gray) to fully saturated (no white component). As value, or brightness, varies from 0 to 1.0, the corresponding colors become increasingly brighter [30].



**Figure 1.11: HSV Coordinates System [30]    Figure 1.12: HSV Color Model [30]**

## 1.5 Image Processing

In electrical engineering and computer science, image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or, a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. It is motivated by two major applications:

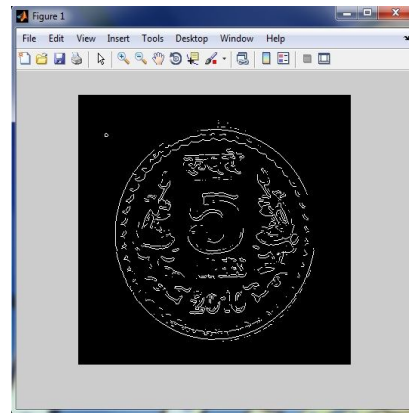
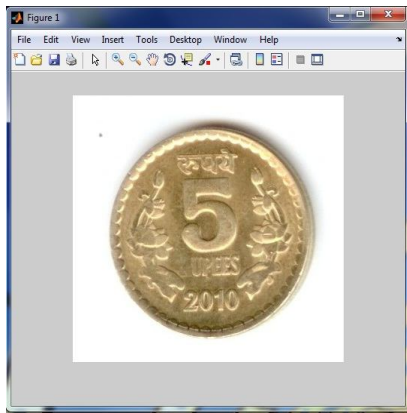
- **Image Enhancement:** Improvement of pictorial information for human perceptions means whatever image you get we want to enhance the quality of the image so that image will have better look.

- Efficient storage and transmission for example if we want to store the image on our computer the image will need certain amount of space in hard disk so to reduce this space requirement we apply image processing techniques.
- Image Quality Control: Digital images are captured, stored, transmitted, and displayed in different devices. Due to this the quality of images degrades. So image processing is used to maintain the image quality.

Image processing modifies pictures to improve them (enhancement, restoration), extract information (analysis, recognition), and change their structure (composition, image editing).

### 1.5.1 Sobel Edge Detection

In edge detection the aim is to mark the points in an image at which the intensity changes sharply. Sharp changes in image properties reflect important events which include: (i) Discontinuities in depth. (ii) Changes in material properties. (iii) Variations in scene illumination. Edge detection is used in the field of image processing and feature extraction. The Sobel operator is such an operator used in edge detection algorithms [26]. Figure 1.13 shows an original image and Figure 1.14 shows the same image after Sobel Edge Detection.



**Figure 1.13: Original Image      Figure 1.14: Image after Sobel Edge Detection**

### 1.5.2 Histogram

The histogram of an image shows us the distribution of Gray levels in image massively useful in image processing; especially in segmentation.

- The array “count” can be plotted to represent a “histogram” of the image as the number of pixels at particular gray level.

- The histogram can yield useful information about the nature of the image. An image may be too bright or too dark.
- It is useful to find the variations of gray levels in an image.

### **1.5.3 Segmentation**

Image segmentation is a process of partitioning the digital image into multiple regions that can be associated with the properties of one or more criterion. It is an initial and vital step in pattern recognition a series of processes aimed at overall image understanding. Properties like gray level, color, texture, and shape help to identify regions and similarity of such properties, is used to build groups of regions having a particular meaning.

- Segmentation divides an image into its constituent regions or objects.
- Segmentation of images is a difficult task in image processing. Still under research.
- Segmentation allows extracting objects in images.
- Segmentation is unsupervised learning.
- Model based object extraction, e.g., template matching, is supervised learning.

### **1.6 Thesis Outline**

Chapter 2 describes various approaches and techniques for coin recognition. A comparison between the approaches is also presented. Chapter 3 throws light on the gaps in the existing work and then briefly describe the problem statement. Chapter 4 demonstrates the step by step implementation of the ANN based Automated Coin Recognition System in the MATLAB. Chapter 5 shows the testing and validation results obtained during experiments. Then, in Chapter 6 we have concluded the thesis and also give some future directions for work.

## Chapter 2

### Literature Review

---

There are various approaches proposed by various researchers for image based coin recognition. In this chapter a brief description of these approaches and a comparison between them is given. We can broadly classify these approaches based on the coins on which they can be applied:

- Approaches for modern coins
- Approaches for ancient coins
- Approaches for both ancient and modern coins

#### 2.1 Approaches for Modern Coins

*In 1992 [1] Minoru Fukumi et al.* presented a rotational invariant neural pattern recognition system for coin recognition. They have used 500 yen coin and 500 won coin to perform the experiment. In this work they have created a multilayered neural network and a preprocessor consists of many slabs of neurons. This preprocessor was used to get a rotational invariant input for the multilayered neural network. For the weights of neurons in preprocessor, concept of circular array was used instead of square array. The results show that 25 slabs with 72 neurons in each slab give the best recognition.

*In 1993 [2] Minoru Fukumi et al.* tried to achieve 100% accuracy for coins. They have used 500 yen coin and 500 won coin. In this work they have used BP (Back Propagation) and GA (Genetic Algorithm) to design neural network for coin recognition. BP is used to train the network. Then after training, GA is used to reduce the size of network by varying the architecture to achieve 100% recognition accuracy rate.

*Paul Davidsson [3] in 1996* presented an approach for coin classification using learning characteristic decision trees by controlling the degree of generalization. Decision trees constructed by ID3-like algorithms were unable to detect instances of

categories not present in the set of training examples. Instead of being rejected, such instances get assigned to one of the classes actually present in the training set. To solve this problem the algorithm with learning characteristic, rather than discriminative, category descriptions was proposed. In addition, the ability to control the degree of generalization was identified as an essential property of such algorithms. Experiments were performed on Canadian and Hong-Kong coins and accuracy of 99.7% for Canadian and 98.3% for Hong-Kong coins was achieved.

*Michael Nolle et al. [4]* at the ARC Seibersdorf research centre in 2003 developed a coin recognition and sorting system called Dagobert. This system was designed for fast classification of large number of modern coins from 30 different countries. Coin classification was accomplished by correlating the edge image of the coin with a pre-selected subset of master coins and finding the master coin with lowest distance. Pre-selection of master coins was done based on three rotation-invariant features (edge angle distribution, edge distance distribution, occurrences of different rotation-invariant patterns on circles centered at edge pixels), coin diameter and thickness. Experiments on 12,949 coins were performed and 99.24% recognition rate was achieved.

A coin recognition system to recognize US coins using vector quantization and histogram modeling was presented by *Seth McNeill et al. [5]* in 2004. The system mainly focuses on the texture of various images imprinted on the coin tail. Based on different image texture the system differentiate between Bald eagle on the quarter, the Torch of liberty on the dime, Thomas Jefferson's house on the nickel, and the Lincoln Memorial on the penny. Experiments show that out of 200 coin images 188 were correctly classified. Thus, 94% recognition accuracy rate was achieved.

In 2005 a multistage approach for coin classification using Eigenspace and Bayesian fusion was presented by *Reinhold Huber et al. [6]*. In the first stage, a translational and rotational invariant description is computed. In a second stage, an illumination-invariant eigenspace is selected and probabilities for coin classes are derived for both sides of each coin. In the final stage, coin class probabilities for both coin sides are combined through Bayesian fusion including a rejection mechanism. Experiments

show that 93.23% of 11,949 coins from 30 different countries were correctly classified.

*In 2005 [7] R. Bremananth et al.* presented an approach using neural network for coin recognition. In this work they have concentrated on the recognition of the numerals on the coin rather than other images. For this they extract a sub image of numeral from coin then this sub image is used for character recognition. To achieve rotation invariance Gabor filters and Back Propagation neural network are used. Experiments are performed on 1-rupee, 2-rupee and 5-rupee coin. The experiments show 92.43% recognition accuracy rate.

*L.J.P. van der Maaten et al. [8] in 2006* developed a fast system for reliable coin classification called COIN-O-MATIC. In this system coin classification is done based on edge-based statistical features (edge angle distributions, edge distance distributions, edge angle-distance distributions). The system consists of four subsystems: (1) a segmentation subsystem, (2) a feature extraction subsystem, (3) a classification subsystem, and (4) a verification subsystem. Experiments were performed on MUSCLE-CIS dataset and 72% classification accuracy was achieved.

*Adnan Khashman et al. [9, 10]* presented an Intelligent Coin Identification System (ICIS) in 2006. ICIS uses neural network and pattern averaging for recognizing rotated coins at various degrees. ICIS consists of two phases. First is image processing in which coin images are mode converted, cropped, compressed, trimmed, pattern averaged etc. This is the preprocessing phase. In second phase a back propagation neural network is trained. Once neural network converges and learns then only one forward pass is used that yields the identification results. ICIS shows very encouraging results. It shows 96.3% correct identification *i.e.* 77 out of 80 variably rotated coin images were correctly identified. ICIS is very effective in reducing time costs because it reduces the processing data by preprocessing images.

*In 2006 [12] P.Thumwarin et al.* presented a robust method for coin recognition with rotation invariance. In this method the rotation invariance feature is represented by the absolute value of Fourier coefficients of polar image of coin on circles with different radii. In this paper the variations on surface of coin such as light reflection effect are

also considered. These effects can be reduced by Fourier approximation of the coin image. Finally coins are recognized by calculating distance between the absolute value of Fourier coefficients obtained from the reference coin and the coin to be recognized.

A fast and reliable coin recognition system based on registration approach was presented by *Marco Reisert et al. [13]* in 2007. For this gradient directions are used whereas gradient magnitude is completely neglected. Then classification is performed by simple nearest neighbor classifier scheme followed by several rejection criteria to meet the demand of low false positive rate. The system presented was also reliable to illumination and contrast changes. Experiments were performed on CIS benchmark dataset.

In 2009 [18] *Linlin Shen et al.* presented an image based approach for coin recognition. In this paper Gabor wavelets are used to extract features. Coin image is divided into number of small sections by concentric ring structures. Then Gabor coefficients from each section are concatenated to form feature vectors. Then these feature vectors are compared with the feature vectors of saved images in database. Identification is done using Euclidean distance and the nearest neighbor classifier. For experiments public MUSCLE database consisting of over 10,000 images is used. This algorithm shows the recognition accuracy rate of 74.27%.

*CHEN Cai-ming et al. [19]* in 2010 presented a coin recognition system with rotation invariance. In this system same approach presented in [12] is used but with BP neural network *i.e.* the feature vector which is the input to BP neural network is obtained by calculating the absolute value of Fourier coefficients of polar image of coin on circles with different radii. The experiments show the accuracy rate of 83%.

An efficient method for coin recognition using a statistical approach was presented in 2010 by *Hussein R. Al- Zoubi [20]*. In this paper statistical methods are used to recognize Jordanian coins. In this approach main focus is on area and color of coins. First of all coin image is converted to gray level image then the gray image is segmented into two regions coin and background based on the histogram of image. Then segmented image is cleaned and then four parameters *i.e.* area, average red,

average green and average blue are calculated. Then based on these parameters decision is made that to which category the coin belongs. This approach yields high recognition rate of 97% which is very encouraging.

*In 2010 [21] Huahua Chen* presented an approach for Chinese coin recognition based on unwrapped image and rotation invariant template matching. In this approach first of all coin segmentation is done using Hough transform then the segmented image is unwrapped. Unwrapping is done by transforming reference and specimen coin image from Cartesian coordinates to polar coordinates. After unwrapping, the template matching is done and on the basis of this recognition is done. Experiments were performed on 144 variably rotated coins images. Out of which 116 were correctly recognized. So, overall 80.6% correct recognition was achieved.

*In 2011 [22] Vaibhav Gupta et al.* presented an approach based on image subtraction technique for recognition of Indian coins. In this approach system performs 3 checks (radius, coarse and fine) on the input coin image. First of all radius is calculated of the input image. Then based on the radius a test image (rotated at certain fixed angle) from database is selected. Then coarse image subtraction between object and test image is done. Then, minima of the resultant image is checked if it is less than a specified threshold then fine image subtraction between object and test image is done otherwise new test image is selected. Then based on fine image subtraction, recognition takes place.

## **2.2 Approaches for Ancient Coins**

Very less work has been done on recognition of ancient coins. The main reason for this is that the ancient coins do not have symmetrical boundaries like modern coins because ancient coins were hammered or casted during manufacturing whereas modern coins are minted. Also ancient coins are generally found in poor conditions due to wear or fouling. So due to irregular shape and poor condition, the general approaches of coin recognition easily fail for recognition of ancient coins. A brief description about an approach for recognition of ancient coins is given in the following paragraph:

*Kaiping Wei et al. [14]* in 2007 presented a novel approach for classification of ancient coins based on image textual information. For extracting textual information Tree-Structured Wavelet Transform (TWT) and Ant Colony Optimization (ACO) algorithm is used. The multi-resolution character of the texture is extracted by TWT, and information can be accessed in various scale rather than low frequency. In addition, segmentation algorithm based on ACO is implemented before TWT to obtain textural information with the absence of noise. The results show that this hybrid approach provides very accurate recognition results for ancient coins.

### **2.3 Approaches for both Modern and Ancient Coins**

*In 2006 [11] Laurens J.P. van der Maaten et al.* presented algorithms for automatic coin classification. The algorithms take digital images of coins as input and generate a class as output *i.e.* to which class the coin belongs. There are two stages of automatic classification. First is feature extraction stage and second is classification stage. In first stage *i.e.* in feature extraction stage two types of features are extracted. First are Contour features and second are Texture features. For extracting Contour features first of all contour image is extracted from original image of coin and then this contour image is represented in statistical features using multi-scale edge angle histograms and multi-scale edge distance histograms. For extracting texture features two types of wavelet features *i.e.* Gabor wavelet features and Daubechies wavelet features are used. For experiments two datasets are used in this paper. The main dataset is the MUSCLE CIS dataset, which is used for evaluating the effectiveness of the feature types. The second dataset is the Merovingen coin dataset which is employed to evaluate to what extent our feature types are appropriate for ancient coin classification. The results revealed that a combination of Contour and Texture features yield the best performance.

*Abdolah Chalechale [15]* presented a novel approach for coin image recognition using image abstraction and spiral decomposition in 2007. The approach SDAI (Spiral Decomposition of Abstract Image) enables measuring the similarity between full color multi- component coin images and need no cost intensive image segmentation. Here an abstract image is derived from original image based on strong edges of the coin. Then spiral distribution of pixels in the abstract image is employed as the key

concept for feature extraction. Extracted features are scale, translation and rotation invariant. The images used for query set and test database are scanned, photographed or collected from web. The proposed approach is compared with three other approaches *i.e.* QVE, PFD (Polar Fourier Descriptor) and EHD (Edge Histogram Distribution). The results show that the proposed approach is much better than other three approaches because it shows significant improvement in recall ratio using proposed features.

In 2007 [16, 17] Martin Kampel *et al.* gives the overview and the preliminary results of EU project COINS (COmbatting Illicit Numismatic Sales). The project aims to substantially contribute to the fight against illegal trade and theft of coins which appears to be a major part of the illegal antiques market. In this paper recognition of both ancient and modern coins is considered. They have also discussed the existing approaches like Eigenspace approach, Contour based algorithms and Gradient based algorithms. They use two approaches for segmentation: Edge based segmentation and Generalized Hough Transform (GHT). Then they use edge based statistical distribution to extract the features. Then they just compare the features to recognize the coin by using K-nearest Neighbor algorithm. They have also compared the results from both Edge based segmentation and GHT and clearly the former performed better than the latter.

## 2.4 Comparison

In table 2.1, comparison of all the above techniques is given. In this table we have compared techniques based on dataset used and accuracy achieved by them.

**Table 2.1 Comparison of Various Existing Techniques**

Sr. No.	Year	Technique used	Dataset of coins used	Modern/Ancient coins		Accuracy achieved
				Moder n	Ancient	
1	1992 [1]	Neural network, circular array	500 yen (Japan) and 500 won (Korea) coin	√		
2	1993 [2]	Neural network using Genetic Algorithm	500 yen (Japan) and 500 won	√		

			(Korea) coin			
3	1996 [3]	Decision trees	Canadian and Hong Kong coins	√		99.7% - Canadian, 98.3% - Hong Kong coins
4	2003 [4]	Edge angle distribution, edge distance distribution	Coins from 30 countries	√		99.24%
5	2004 [5]	Vector Quantization and Histogram Modeling	US coins	√		94%
6	2005 [6]	Eigenspaces and Bayesian fusion	Coins from 30 countries	√		93.23%
7	2005 [7]	Neural network, Gabor filter, Statistical color threshold	Indian coins	√		92.43%
8	2006 [8]	Edge angle distribution, edge distance distribution, edge angle-distance distribution	MUSCLE CIS dataset	√		72%
9	2006 [9, 10]	Neural Network, Pattern Averaging	Turkish 1 Lira and 2 Euro coin	√		96.3%
10	2006 [11]	Multi-scale edge angle histograms, Multi-scale edge distance histograms, Gabor wavelet, Daubechies wavelet	MUSCLE CIS dataset and Merovingen coin dataset	√	√	76%
11	2006 [12]	Fourier approximation of polar image	Thai amulet and Thai baht coins	√		
12	2007 [14]	Tree structured wavelet transform, Ant colony optimization algorithm			√	
13	2007 [13]	Registration approach based on gradient directions	CIS Benchmark dataset	√		
14	2007 [15]	Image abstraction and spiral decomposition	COIN BANK	√	√	
15	2007 [16,	Edge based segmentation,	MUSCLE CIS dataset	√	√	76%

	17]	Generalized hough transform and K-nearest neighbor algorithm				
16	2009 [18]	Gabor wavelet, Euclidean distance and nearest neighbor classifier	MUSCLE dataset	√		74.27%
17	2010 [19]	Fourier approximation of polar image, neural network	Chinese coins	√		83%
18	2010 [20]	Statistical approach	Jordanian coins	√		97%
19	2010 [21]	Rotation invariant template matching	Chinese coins	√		80.6%
20	2011 [22]	Image subtraction technique	Indian coins	√		

## Chapter 3

### Problem Statement

---

In this chapter, gaps in the existing work, problem statement, objectives to achieve and methodology to achieve the objectives are discussed.

#### 3.1 Gap Analysis

In the previous chapter of Literature Survey, various techniques are discussed for the recognition of coins based on image processing. Following are the gaps in the existing work:

- 1) Existing techniques require too much time for processing. So these can be improved to process the images in real-time.
- 2) There is very less work done for the recognition of ancient coins because mostly ancient coins are found in poor condition and do not possess regular boundaries.
- 3) Existing systems sometimes fail to recognize those coins which wear out because of excessive use.
- 4) Most of the systems available recognize the coins by taking physical properties like radius, thickness etc into consideration. Due to which these systems can be fooled easily.

#### 3.2 Problem Statement

There are various techniques already implemented for recognition of coins based on features like thickness, weight and diameter etc. But these techniques or systems can be easily fooled by fake coins having same physical properties as the original coin. Therefore to remove such discrepancies features such as the drawings and numerals printed on the coin could be used as the patterns for which neural networks can be trained so that more accurate recognition results can be obtained.

#### 3.3 Objectives

- 1) To study the existing techniques for coin recognition.

- 2) To propose a technique for coin recognition using Neural Network.
- 3) To implement the proposed technique.
- 4) To test and validate the implemented technique.

### **3.4 Methodology**

- 1) Literature Survey
- 2) Purpose the technique using concepts of image processing and neural networks.
- 3) Implementation process is as follow:
  - 3.1) Preprocess the images of coins through Hough Transformation, Pattern Averaging etc.
  - 3.2) Create a Neural Network using Neural Network Toolbox in MATLAB.
  - 3.3) Train the network using some of preprocessed images.
- 4) Test the network for other preprocessed images that are not used in training and observe the results.

## Chapter 4

### Implementation Details

---

This chapter includes the implementation details for the automated coin recognition system. MATLAB 7.11 R2010b is used for implementation. It integrates computation, visualization and programming in easy to use environment. As discussed in literature survey, there are various approaches/techniques available for coin recognition, each with its own merits and demerits. In our implementation we have used neural networks for coin recognition. This section gives a brief introduction to the MATLAB and presents a neural network-based coin recognition system.

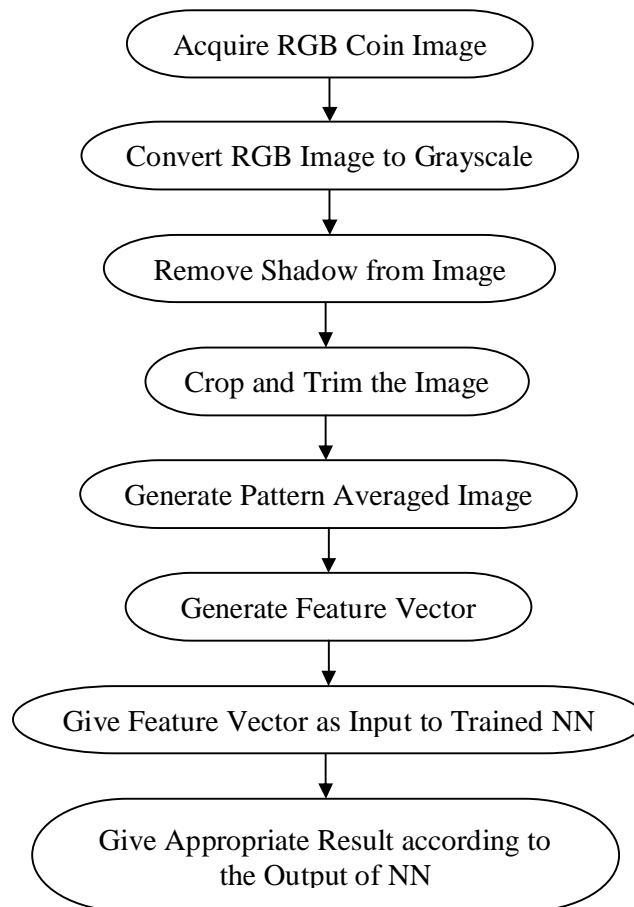
#### 4.1 What is MATLAB?

The name MATLAB stands for MATrix LABoratory. Dr. Cleve Moler, chief scientist at Math Works Inc. originally wrote MATLAB to provide easy access to matrix software developed in the LINPACK and EISPACK projects. The first version was written in the late 1970s for the use in courses in matrix theory, linear algebra, and numerical analysis. MATLAB is therefore built upon a foundation of sophisticated matrix software, in which the basic data element is a matrix that does not require pre-dimensioning. MATLAB is a product of the Math Works, Inc. and is an advanced software package specially designed for scientific and engineering computation. MATLAB is a high-performance language for technical computing [31]. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or FORTRAN. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB provides various types of toolboxes like Neural Network Toolbox, Image Processing Toolbox and Signal Processing Toolbox etc. Toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems.

## 4.2 Architecture for ANN based Automated Coin Recognition System

Coin recognition process is divided into eight steps. The architecture of ANN based automated coin recognition system is shown in Figure 4.1.



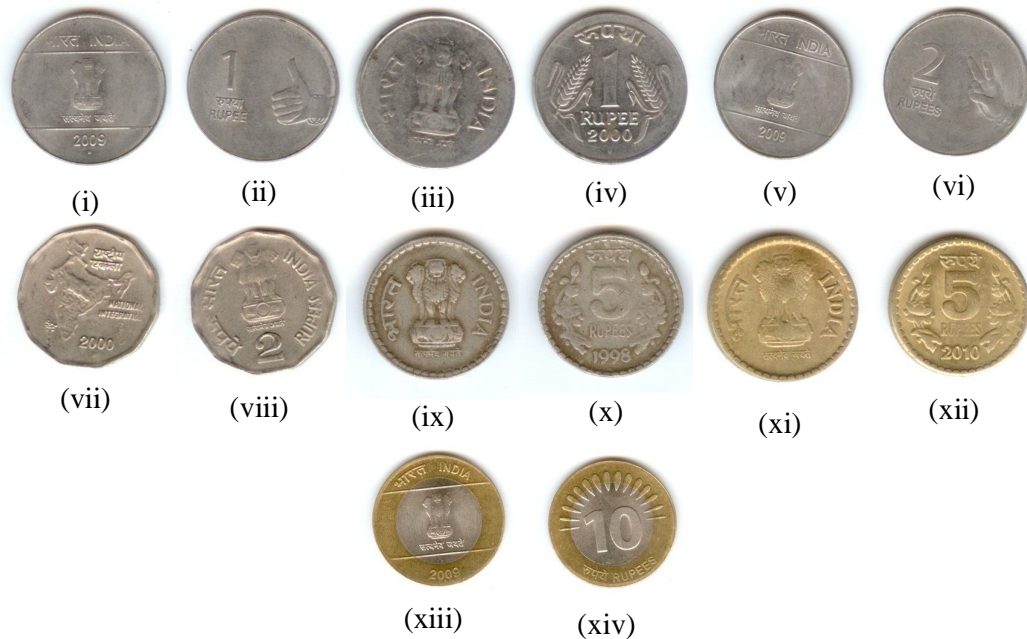
**Figure 4.1: Architecture of ANN based Automated Coin Recognition System**

Now let's discuss each step of the above architecture in detail.

### 4.2.1 Acquire RGB Coin Image

This is the first step of coin recognition process. In this step the RGB coin image is acquired. Indian coins of denominations `1, `2, `5 and `10 were scanned from both

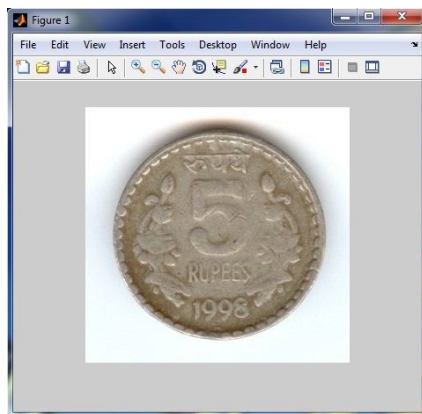
sides at 300 dpi (dots per inch) using color scanner as shown in Figure 4.2. Five coins of each denomination were scanned.



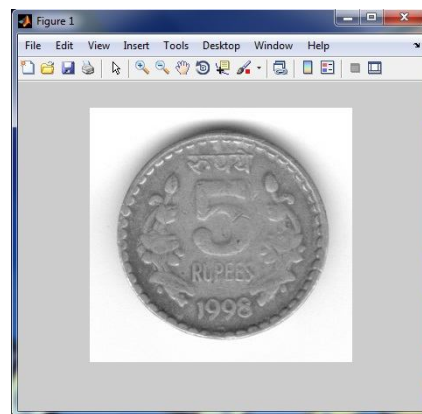
**Figure 4.2: Indian Coins of different Denominations; (i) Head of ₹1 coin (1<sup>st</sup> type), (ii) Tail of ₹1 coin (1<sup>st</sup> type), (iii) Head of ₹1 coin (2<sup>nd</sup> type), (iv) Tail of ₹1 coin (2<sup>nd</sup> type), (v) Head of ₹2 coin (1<sup>st</sup> type), (vi) Tail of ₹2 coin (1<sup>st</sup> type), (vii) Head of ₹2 coin (2<sup>nd</sup> type), (viii) Tail of ₹2 coin (2<sup>nd</sup> type), (ix) Head of ₹5 coin (1<sup>st</sup> type), (x) Tail of ₹5 coin (1<sup>st</sup> type), (xi) Head of ₹5 coin (2<sup>nd</sup> type), (xii) Tail of ₹5 coin (2<sup>nd</sup> type), (xiii) Head of ₹10 coin, (xiv) Tail of ₹10 coin**

#### 4.2.2 Convert RGB Coin Image to Grayscale

From the first step the image we got is a 24-bit RGB image. Image processing of colored images takes more time than the grayscale images. So, to reduce the time required for processing of images in further steps it is good to convert the 24-bit RGB image to 8-bit Grayscale image.



**Figure 4.3: 24-bit RGB Coin Image**



**Figure 4.4: 8-bit Grayscale Coin Image**

### 4.2.3 Remove Shadow of Coin from Image

In this step, shadow of the coin from the Grayscale image is removed. As all the coins have circular boundary. So, for removing shadow Hough Transform for Circle Detection [32] is used. For this first of all edge of the coin is detected using Sobel Edge Detection. Following is the pseudo code for Hough Transform:

- Step 1. Define a 3-dimensional Hough Matrix of  $(M \times N \times R)$ , where M, N is the height and width of the Grayscale image and R is the no. of radii for which we want to search.
- Step 2. For each edge pixel  $(x, y)$  and for particular radius  $r$ , search circle center coordinates  $(u, v)$  that satisfy the equation  $(x-u)^2+(y-v)^2=r^2$  and increase count in Hough Matrix at  $(u, v, r)$  by 1.
- Step 3. Repeat step 2 for other radii.
- Step 4. Find the maximum value from the Hough Matrix. The corresponding indices give the center coordinates and radius of coin.

Now based on the center coordinates and radius, the coin is extracted from the background. So, in this way the shadow of the coin is removed. Figure 4.5 shows a coin with shadow and Figure 4.6 shows the coin without shadow after applying Hough Transform.

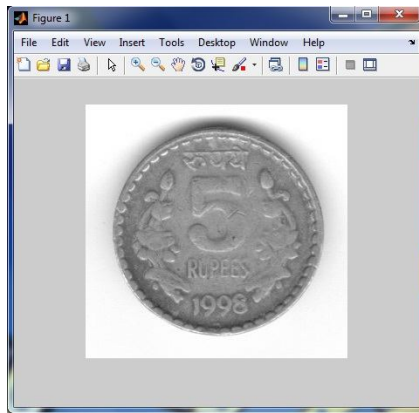


Figure 4.5: Grayscale Coin Image

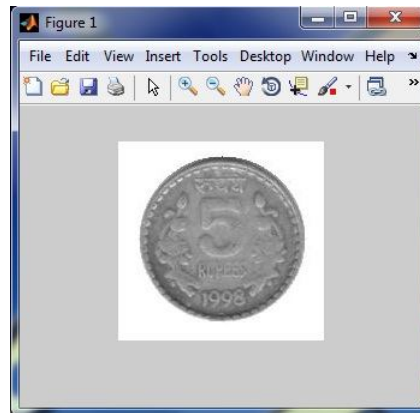
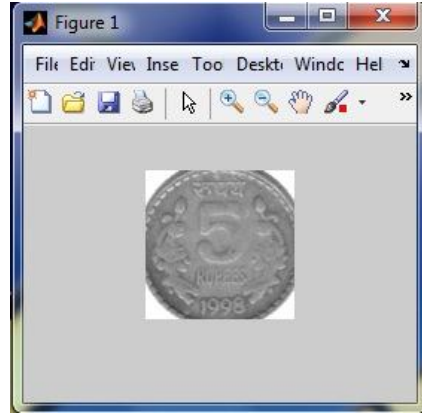
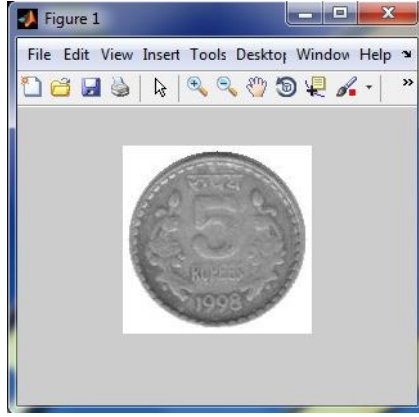


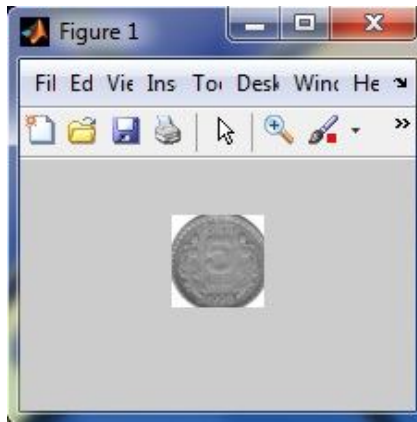
Figure 4.6: Coin Image without Shadow

### 4.2.4 Crop and Trim the Image

After shadow removal the image is cropped so that we just have the coin in the image as shown in Figure 4.7. Then after cropping, coin image is trimmed to make it of equal dimension of  $100 \times 100$  or  $50 \times 50$  as shown in Figure 4.8 and Figure 4.9.



**Figure 4.7: Cropped Coin Image    Figure 4.8: 100×100 Trimmed Coin Image**



**Figure 4.9: 50×50 Trimmed Coin Image**

#### 4.2.5 Generate Pattern Averaged Image

The 100×100 or 50×50 trimmed coin images become the input for the trained neural network. But to reduce the computation and complexity in the neural network these images are further reduced to size 20×20 or 10×10 by segmenting the image using segments of size 5×5 pixels, and then taking the average of pixel values within the segment. This can be represented by mathematical equations, as shown in (4.1) and (4.2):

$$Sum_i = \sum_{j=1}^5 \sum_{k=1}^5 P_{ijk} \quad \dots(4.1)$$

$$SegAvg_i = Sum_i / 25 \quad \dots(4.2)$$

where  $i, j, k$  is the segment no., row no. and column no. of a particular segment respectively,  $Sum_i$  is the sum of the pixel values  $P_{ijk}$  of the segment  $i$ ,  $SegAvg_i$  is the average of pixel values of segment  $i$ .

#### **4.2.6 Generate Feature Vector**

In this step, a feature vector is generated from the pattern averaged coin image. The  $20 \times 20$  image generates a feature vector of dimension  $400 \times 1$  *i.e.* all the pixel values are put into a vector of 1 column. Similarly, from  $10 \times 10$  pattern averaged coin image feature vector of  $100 \times 1$  gets generated.

#### **4.2.7 Give Feature Vector as Input to Trained NN**

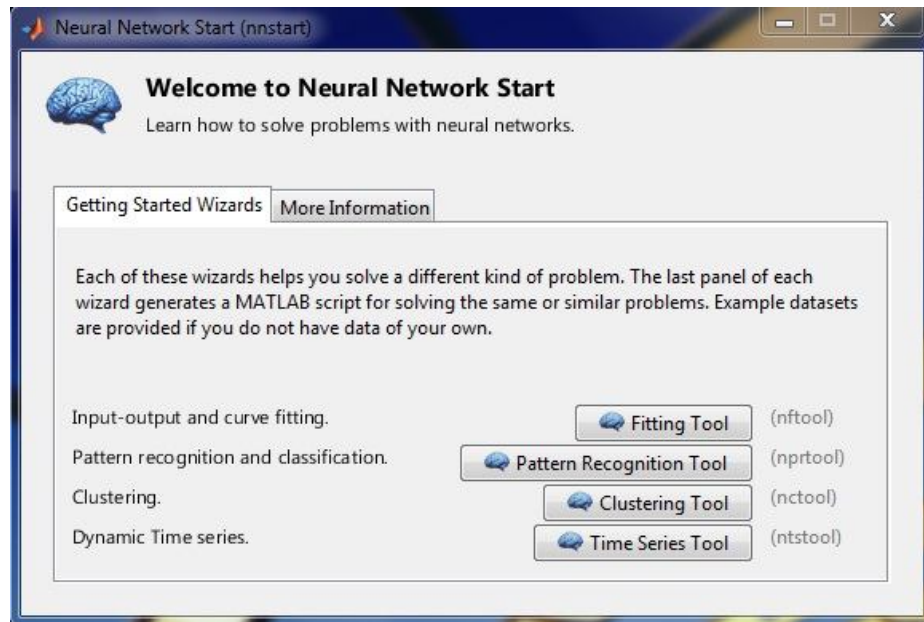
The feature vector from the above step is then passed as input to a trained neural network. This trained neural network classifies the coin into appropriate class based on which the output will be generated. MATLAB provides a Neural Network Toolbox with the help of which Neural Network for pattern recognition can be easily created.

##### **4.2.7.1 Introduction to the GUI**

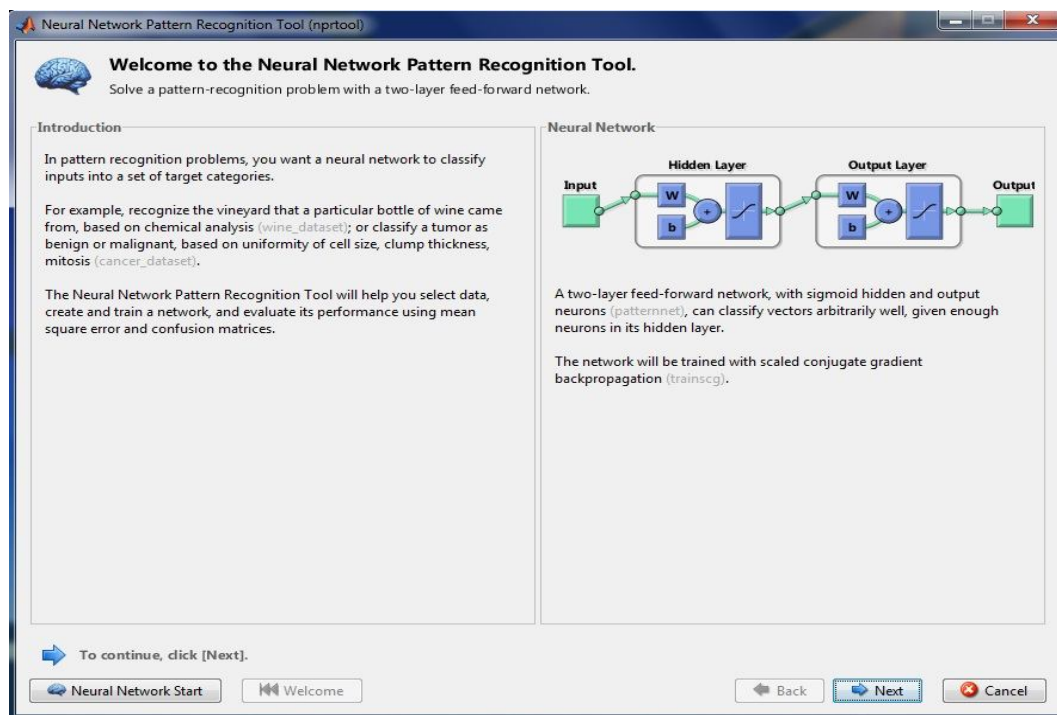
The Neural Network Toolbox makes it easier to use neural networks in MATLAB. The toolbox consists of a set of functions and structures that handle neural networks, so we do not need to write code for all activation functions, training algorithms, etc. that we want to use. The graphical user interface (GUI) is designed to be simple and user friendly. Once the Neural Network Start (nnstart) window is up and running, you can create a network, view it, train it, simulate it, and export the final results to the Workspace. Similarly, you can import data from the workspace for use in the GUI. It goes through all the steps of creating a network and shows what you might expect to see as you go along.

##### **4.2.7.2 Selecting the Wizard according to the Problem Type**

To start creating a Neural Network first of all type the command nnstart in the command window. The window shown in Figure 4.10 will appear. In this window MATLAB provides four different wizards (Fitting Tool, Pattern Recognition Tool, Clustering Tool and Time Series Tool) to solve different type of problems. Click on 'Pattern Recognition Tool' button to create Neural Network for pattern recognition. It will open a wizard window (Neural Network Pattern Recognition Tool) as shown in Figure 4.11.



**Figure 4.10: Neural Network Start Window**

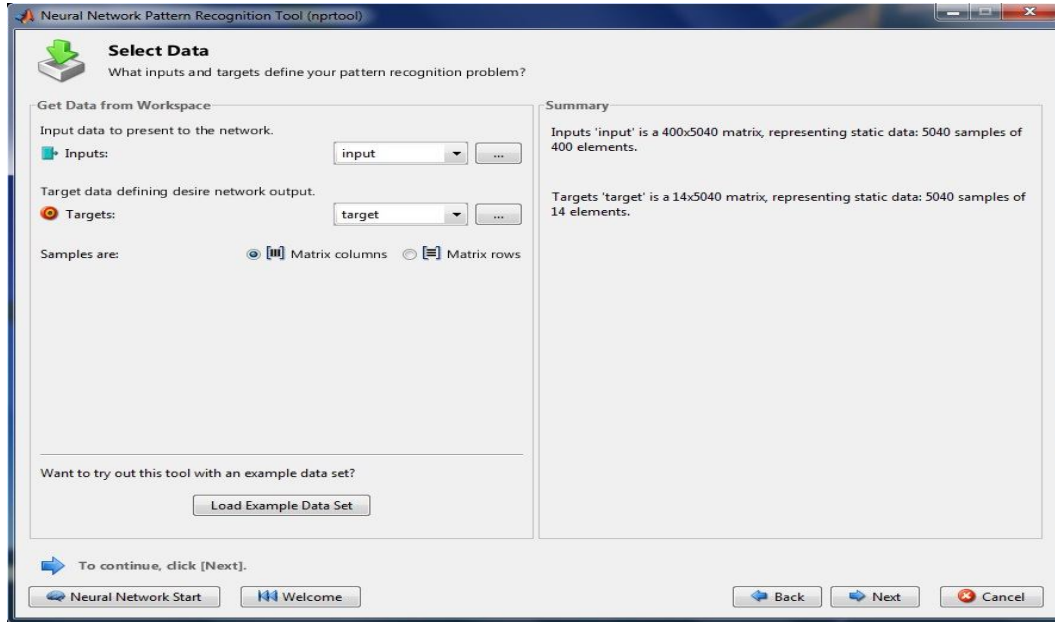


**Figure 4.11: Neural Network Pattern Recognition Tool Window**

### 4.2.7.3 Select Input and Target

Click 'Next' on the Figure 4.11 to open a window as shown in Figure 4.12. In this window we can select data for input and target. From the drop down menu the data can be loaded from the workspace. Also data can be loaded by browsing the data files

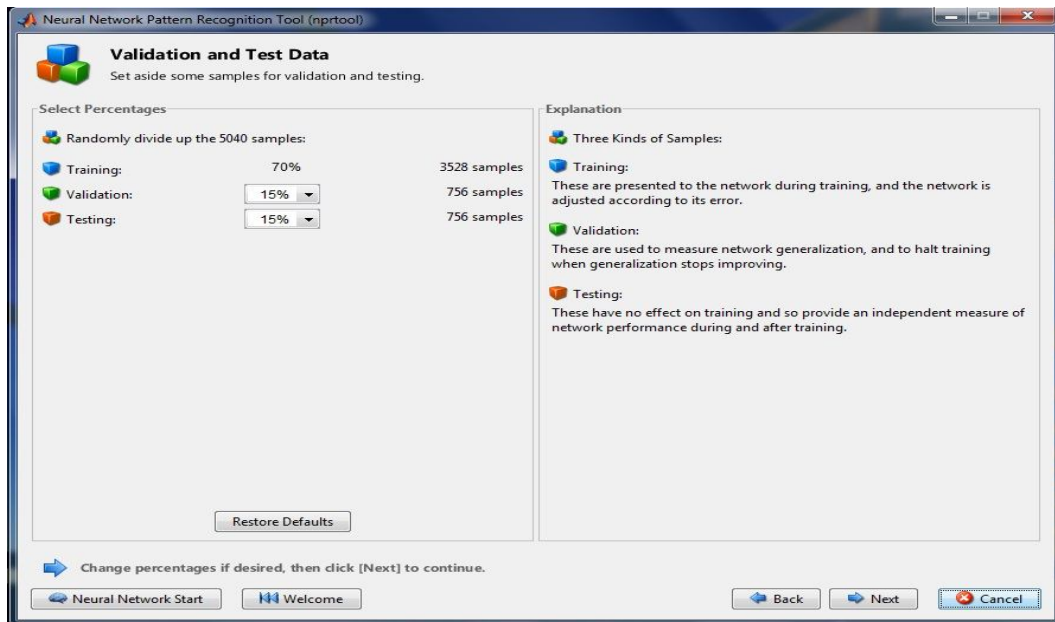
recognized by MATLAB. MATLAB also provides some example datasets that can be loaded directly.



**Figure 4.12: Input / Target Data Window**

#### 4.2.7.4 Select Validation and Test Data

After selecting input / target data click 'Next' to open a window as shown in Figure 4.13. In this window specify how much amount of data to be used for validation and testing. By default both are set to 15%.



**Figure 4.13: Validation and Test Data Window**

#### 4.2.7.5 Select the Number of Hidden Neurons

Click on 'Next' after specifying percentage of samples to be used for validation and testing in Figure 4.13 to open next window as shown in Figure 4.14. In this window specify the number of hidden neurons to be used in Hidden layer. In the second half of this window network diagram is also available.

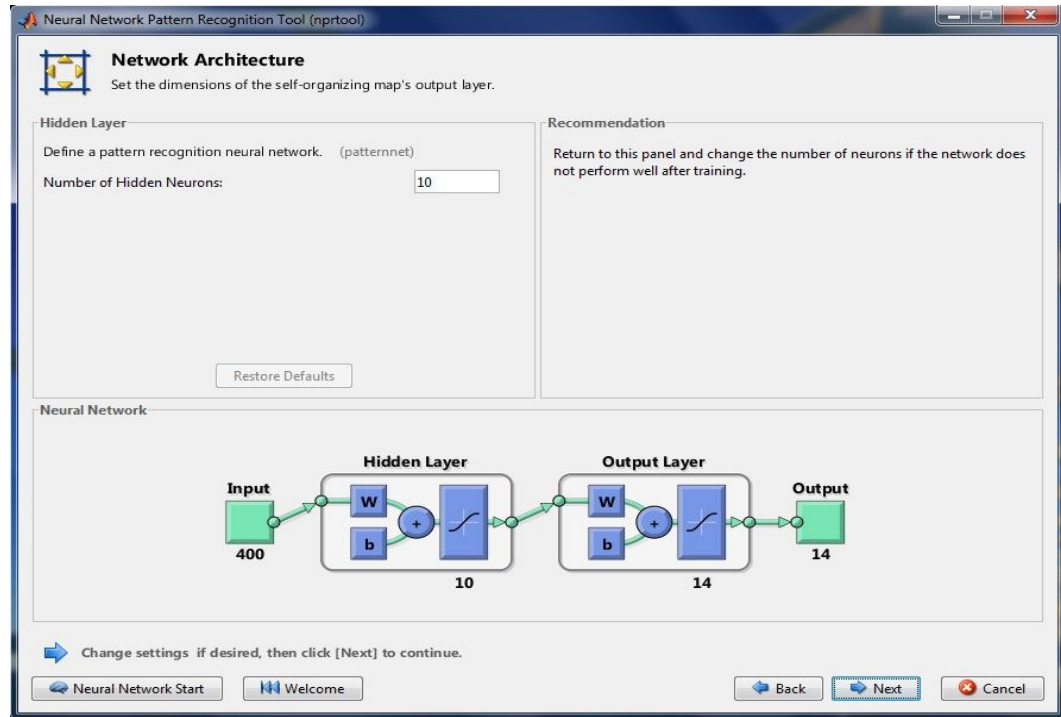


Figure 4.14: Network Architecture Window

#### 4.2.7.6 Train the Network

Now the architecture of network is ready. So now training of the network can be done. Click 'Next' on the Network architecture window to open a window as shown in Figure 4.15. In this window training of the network is done. Click on the 'Train' button to start training, the window shown in Figure 4.16 will appear. When the training gets completed the window shown in Figure 4.15 will get refreshed and on the right side the values for MSE (Mean Square Error) and %E for Training, Validation and Testing will appear. Smaller the values for MSE and %E the better will be the Network. If the values for MSE and %E are not satisfactory then retraining of the network can be done by clicking on the 'Retrain'. Training network multiple times will generate different results due to different initial conditions and

sampling.

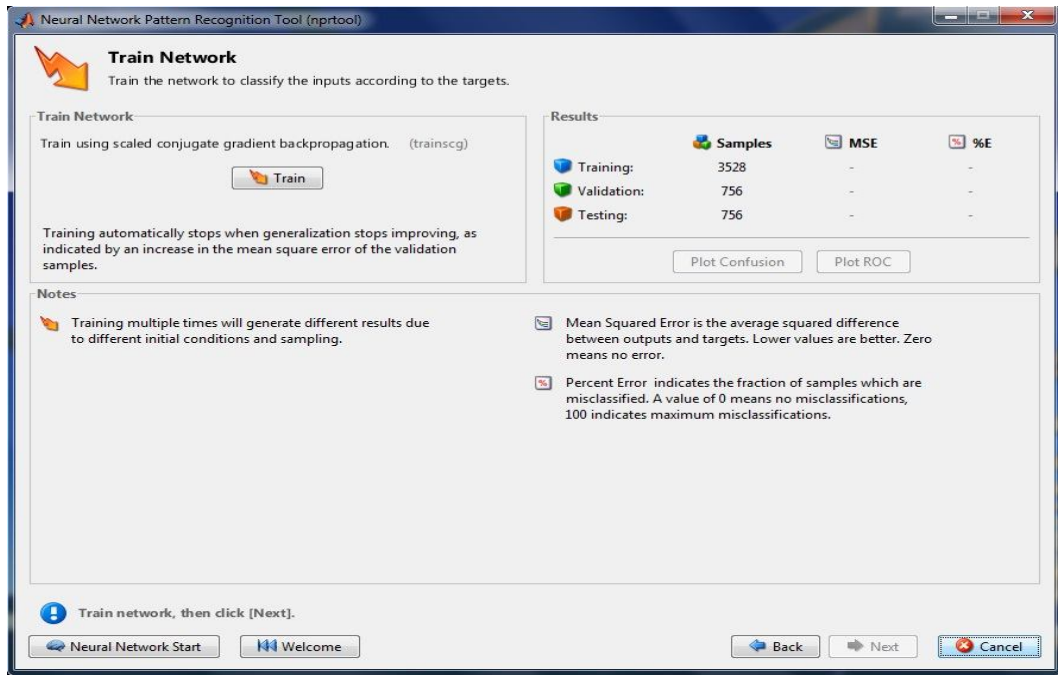


Figure 4.15: Train Network Window

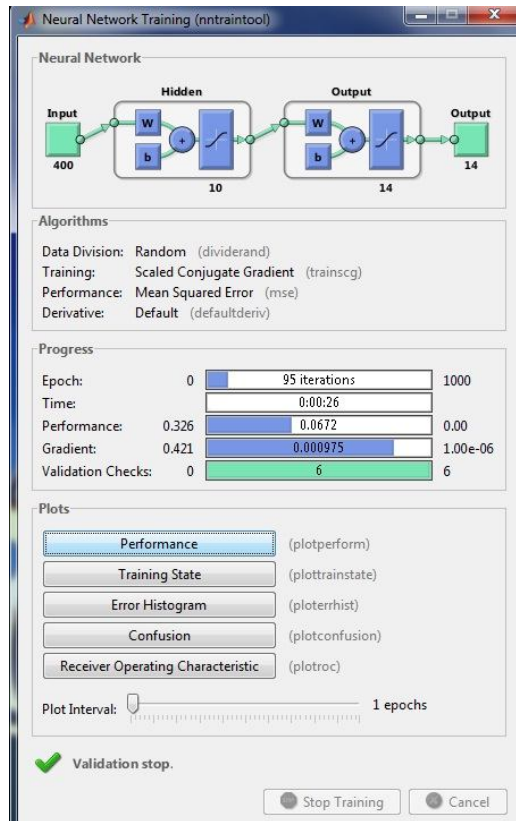


Figure 4.16: Neural Network Training Window

### 4.2.7.7 Evaluate Network

Additional tests can also be performed on the network. For this, click on the ‘Next’ button in the window shown in Figure 4.16 to open a window shown in Figure 4.17. In this window network can be retrained, network size can be adjusted and larger dataset can be imported to train the network.

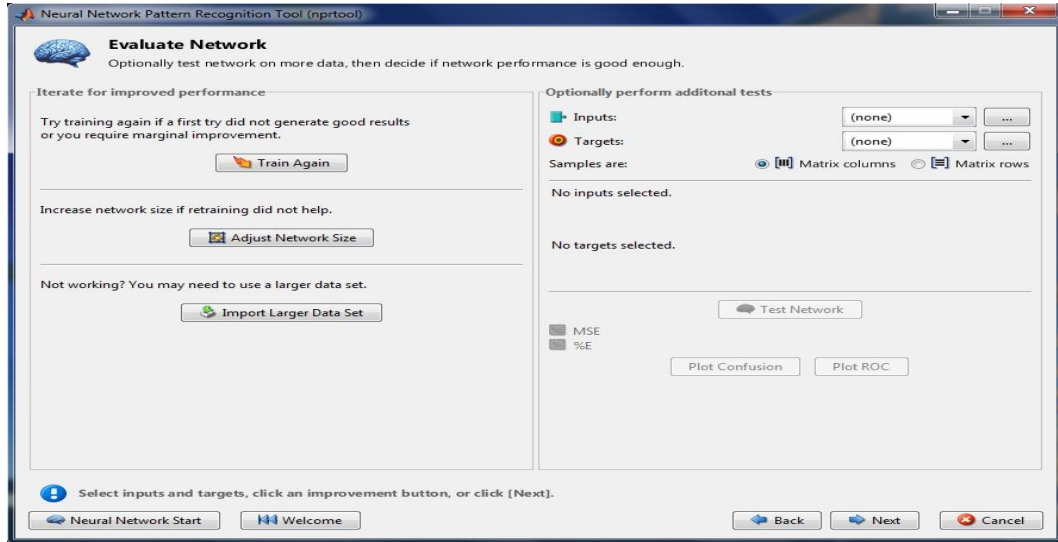


Figure 4.17: Evaluate Network Window

### 4.2.7.8 Save Results

Click ‘Next’ on the Evaluate Network window, the window shown in Figure 4.18 will appear. In this window the results obtained can be exported or saved to the MATLAB workspace. Also Simple or Advanced script for the network can be generated.

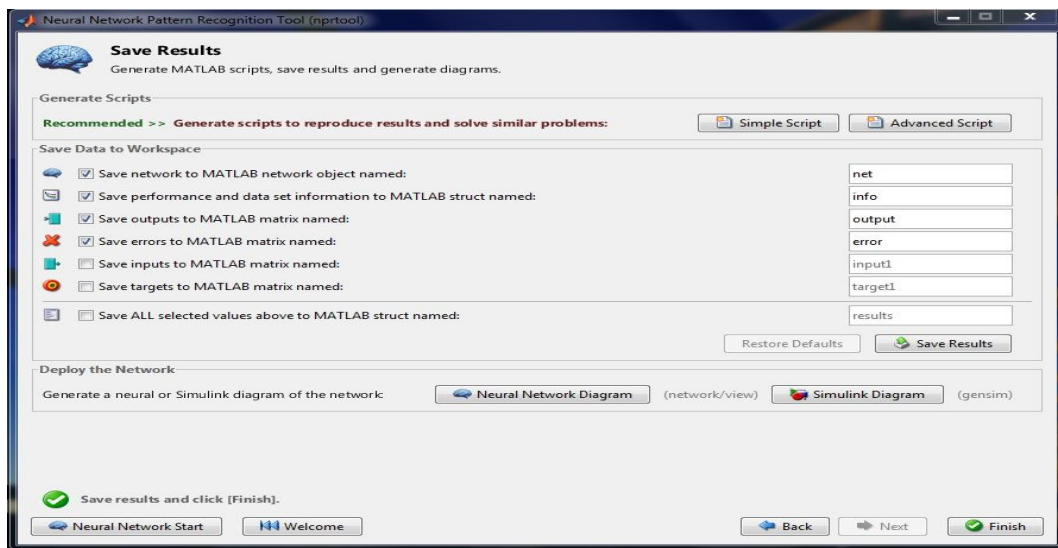


Figure 4.18: Save Results Window

#### 4.2.8 Give Appropriate Result according to the Output of Neural Network

In this thesis coins are classified into 14 categories as shown in Figure 4.2. The neural network classifies the given coin image into one of these classes and based on the classification the results get generated that to which denomination the given coin belongs.

#### 4.3 ANN based Automated Coin Recognition System

In Figure 4.19 interface of ANN based Automated Coin Recognition System is shown. In this interface first of all user have to browse a coin image to be recognized. Then click on the 'Load Image' button to load the image. Then click on 'Recognize' button to start the recognition process. In Figure 4.19 an example is shown in which recognition of 10 rupee coin is done.



**Figure 4.19: ANN based Automated Coin Recognition System**

## Chapter 5

### Experimental Results

---

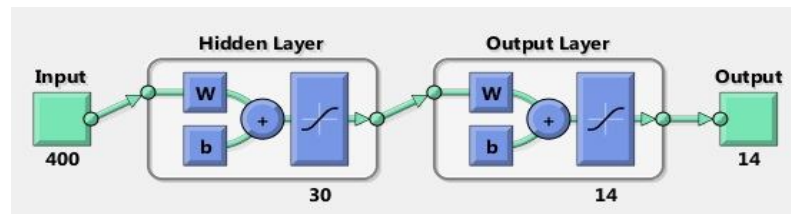
Two neural networks have been created to recognize the coins based on the number of features used for recognition. One neural network takes 400 features so let's call it as coin\_neural\_400 and the other one takes 100 features so let's name it as coin\_neural\_100. For training and testing of both the networks following data has been used.

#### 5.1 Training and Testing Data

Five samples of each denomination of Indian coins are scanned from both sides as shown in Figure 4.2. So, it results to 10 images for each coin. But for `1, `2 and `5 two types of coins are used. So for each of these denominations there are 20 images from which 10 (5 for head and 5 for tail) are of 1<sup>st</sup> type and other 10 (5 for head and 5 for tail) are of 2<sup>nd</sup> type of that particular denomination. Then after preprocessing when we get images of 100×100 and 50×50 then these images were rotated to  $5^0, 10^0, 15^0, \dots, 355^0$  i.e. total 72 rotated images get generated for each image. So there are  $20 \times 72 = 1440$  images for `1, `2 and `5 but  $10 \times 72 = 720$  images for `10. So there are total  $1440 \times 3 + 720 = 5040$  images for each network.

#### 5.2 Neural Network Architecture

In the first type of neural network i.e. coin\_neural\_400, three layers (input layer, hidden layer and output layer) are used. Network diagram for coin\_neural\_400 is shown in Figure 5.1.



**Figure 5.1: Network Diagram for 400 Features (coin\_neural\_400)**

Similar to the coin\_neural\_400 three layers are used for coin\_neural\_100 as shown in Figure 5.2.

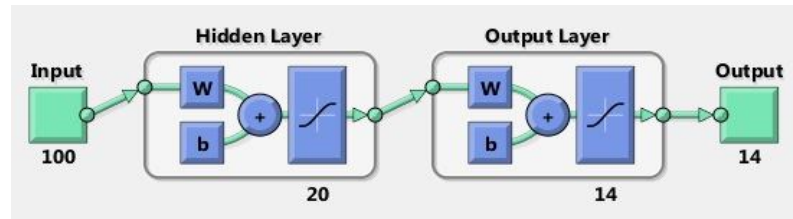


Figure 5.2: Network Diagram for 100 Features (coin\_neural\_100)

### 5.3 Results for coin\_neural\_400

Random samples get selected from 5040 sample images for training, testing and validation. 90% sample images are used for training, 5% for testing and 5% for validation. Figure 5.3 shows the values for MSE (Mean Square Error) and %E for training, testing and validation.

	Samples	MSE	%E
Training:	4536	1.65692e-3	2.00617e-0
Validation:	252	3.81927e-3	3.57142e-0
Testing:	252	6.43451e-3	5.55555e-0

Figure 5.3 Results after Training, Testing and Validation of coin\_neural\_400

Training of the network takes 148 epochs in total. Figure 5.4 shows the performance of network for each training, testing and validation. The best validation performance is achieved at epoch 142.

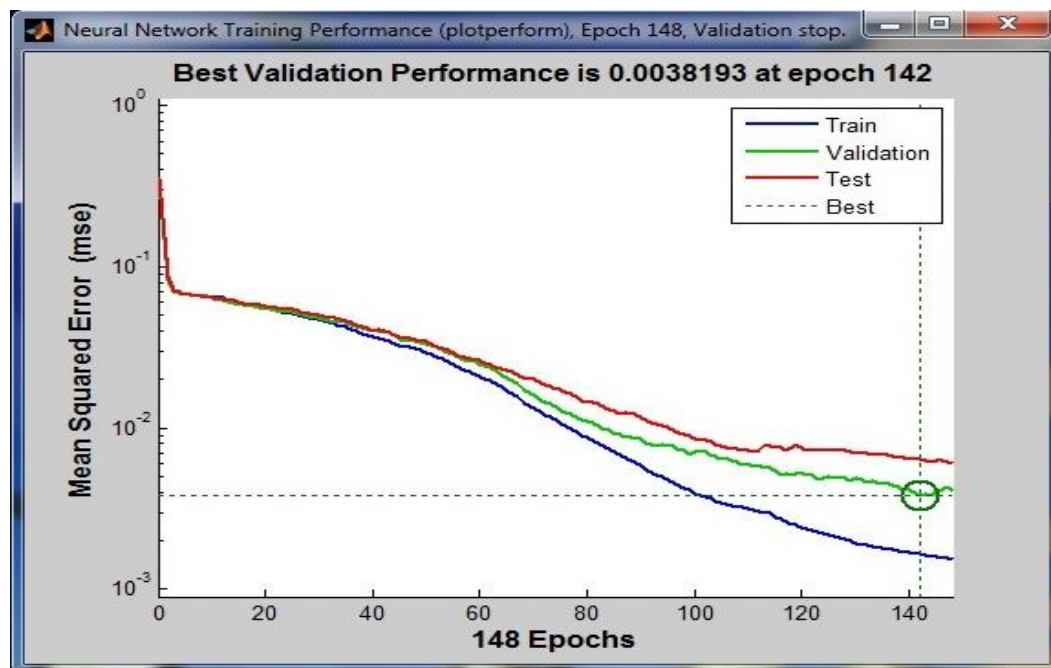


Figure 5.4: Performance of Network coin\_neural\_400

The Figure 5.5 shows the confusion matrix for coin\_neural\_400. In confusion matrix Target classes are the classes to which the coin actually belongs and Output classes are the classes in which the coins get classified by trained NN. It is clear from the figure that 97.74% correct recognition has been achieved which is quite encouraging. So, there is 2.26% misclassification only.

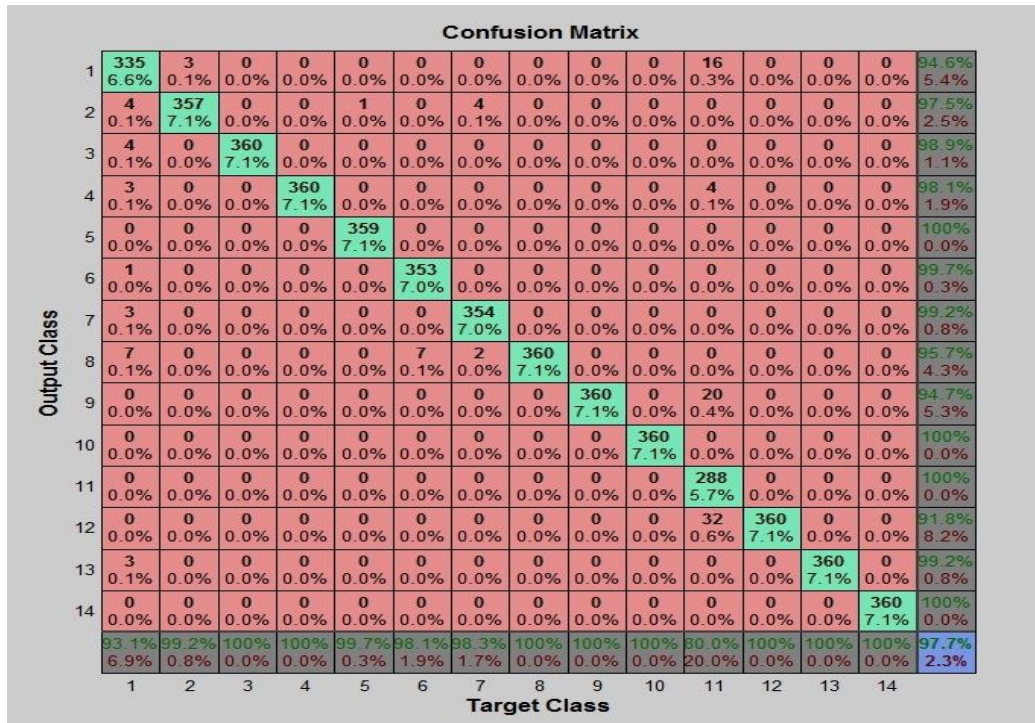


Figure 5.5: Confusion Matrix for coin\_neural\_400

## 5.4 Results for coin\_neural\_100

Similarly for coin\_neural\_100 random samples get selected from 5040 sample images for training, testing and validation. 90% sample images are used for training, 5% for testing and 5% for validation. Figure 5.6 shows the values for MSE (Mean Square Error) and %E for training, testing and validation.

	Samples	MSE	%E
Training:	4536	1.51706e-2	18.71693e-0
Validation:	252	1.79188e-2	20.63492e-0
Testing:	252	1.47992e-2	15.07936e-0

Figure 5.6 Results after Training, Testing and Validation of coin\_neural\_100

Training of the network takes 164 epochs in total. Figure 5.7 shows the performance of network for each training, testing and validation. The best validation performance is achieved at epoch 158.

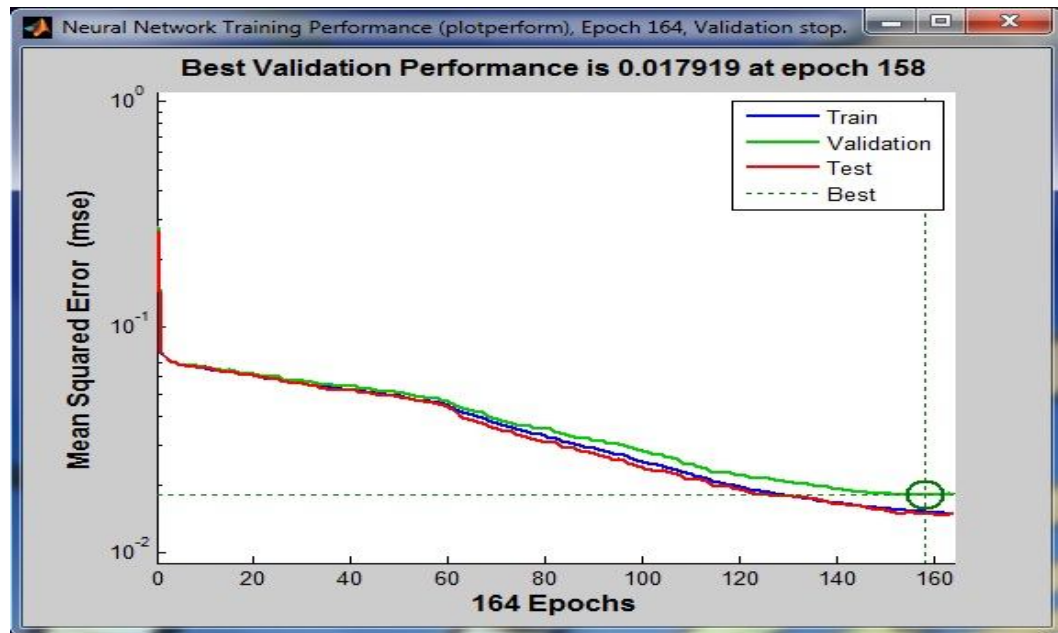


Figure 5.7: Performance of Network coin\_neural\_100

The Figure 5.8 shows the confusion matrix for coin\_neural\_100. It is clear from the figure that 81.4% correct recognition has been achieved. So, there is 18.6% misclassification.

Confusion Matrix															
Output Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	298 5.9%	4 0.1%	0 0.0%	2 0.0%	3 0.1%	68 1.3%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	6 0.1%	53 1.1%	0 0.0%	0 0.0%	68.6%
2	0 0.0%	216 4.3%	0 0.0%	5 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	97.7%
3	2 0.0%	6 0.1%	360 7.1%	37 0.7%	1 0.0%	46 0.9%	7 0.1%	1 0.0%	5 0.1%	0 0.0%	1 0.0%	6 0.1%	0 0.0%	0 0.0%	76.3%
4	8 0.2%	54 1.1%	0 0.0%	306 6.1%	34 0.7%	18 0.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	15 0.3%	26 0.5%	0 0.0%	0 0.0%	66.4%
5	1 0.0%	3 0.1%	0 0.0%	1 0.0%	255 5.1%	17 0.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	9 0.2%	0 0.0%	0 0.0%	89.2%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	144 2.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	18 0.4%	0 0.0%	0 0.0%	88.9%
7	31 0.6%	56 1.1%	0 0.0%	0 0.0%	11 0.2%	46 0.9%	350 6.9%	0 0.0%	10 0.2%	0 0.0%	37 0.7%	2 0.0%	0 0.0%	0 0.0%	64.5%
8	7 0.1%	0 0.0%	0 0.0%	1 0.0%	2 0.0%	3 0.1%	1 0.0%	352 7.0%	21 0.4%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	90.7%
9	2 0.0%	16 0.3%	0 0.0%	0 0.0%	29 0.6%	6 0.1%	0 0.0%	0 0.0%	288 5.7%	0 0.0%	1 0.0%	2 0.0%	0 0.0%	0 0.0%	83.7%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	30 0.6%	360 7.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	92.3%
11	11 0.2%	5 0.1%	0 0.0%	8 0.2%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	300 6.0%	28 0.6%	33 0.7%	0 0.0%	77.7%
12	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	6 0.1%	0 0.0%	0 0.0%	216 4.3%	10 0.2%	0 0.0%	92.7%
13	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 0.2%	4 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	296 5.9%	0 0.0%	95.5%
14	0 0.0%	0 0.0%	0 0.0%	0 0.0%	15 0.3%	7 0.1%	0 0.0%	7 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 0.4%	360 7.1%	88.0%
	82.8%	60.0%	100%	85.0%	70.8%	40.0%	97.2%	97.8%	80.0%	100%	83.3%	50.0%	82.2%	100%	81.4%
	17.2%	40.0%	0.0%	15.0%	29.2%	60.0%	2.8%	2.2%	20.0%	0.0%	16.7%	40.0%	17.8%	0.0%	18.6%
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	

Figure 5.8: Confusion Matrix for coin\_neural\_100

## 5.5 Comparison

Table 5.1 shows the comparison of both the networks. It is clear from the data that the network coin\_neural\_400 performs better and provides correct recognition of 97.74%.

**Table 5.1: Comparison of Networks**

Sr. No.	Coin Type	coin_neural_400		coin_neural_100	
		Images correctly recognized / Total no. of images	Recognition Rate (in %age)	Images correctly recognized / Total no. of images	Recognition Rate (in %age)
1	`1	1412/1440	98.05	1180/1440	81.94
2	`2	1426/1440	99.03	1101/1440	76.46
3	`5	1368/1440	95	1164/1440	80.83
4	`10	720/720	100	656/720	91.11
<b>Total</b>		4926/5040	97.74	4101/5040	81.37

#### 6.1 Conclusion

In this thesis an ANN based automated coin recognition system has been developed using MATLAB. In this system, firstly preprocessing of the images is done and then these preprocessed images are fed to the trained neural network. Neural network has been trained, tested and validated using 5040 sample images of denominations `1, `2, `5 and `10 rotated at  $5^{\circ}$ ,  $10^{\circ}$ ,  $15^{\circ}$ .....,  $355^{\circ}$ . In this system images of coins from both sides (Head & Tail) have been used. So, this system is capable of recognizing coins from both sides. We have created two Neural Networks: One takes 400 features as input, we call it as coin\_neural\_400, and other takes 100 features as input, we call it as coin\_neural\_100. Recognition rate of 81.37% has been achieved during the experiments with coin\_neural\_100 whereas coin\_neural\_400 shows drastic improvement and gives recognition rate of 97.74% *i.e.* there is only 2.26% miss-recognition.

#### 6.2 Future Scope

There are so many approaches available for recognition of modern coins. But for ancient coins very less work has been done till now because mostly ancient coins found in poor conditions and also they do not have proper boundary. So, new approaches / techniques can be developed for recognition of ancient coins. Also existing approaches can be extended to increase the accuracy or recognition rate and to make the recognition process real-time.

## REFERENCES

---

- [1] Fukumi M. and Omatu S., "Rotation-Invariant Neural Pattern Recognition System with Application to Coin Recognition", IEEE Trans. Neural Networks, Vol.3, No. 2, pp. 272-279, March, 1992.
- [2] Fukumi M. and Omatu S., "Designing A Neural Network For Coin Recognition By A Genetic Algorithm", Proceedings of 1993 International Joint Conference on Neural Networks, Vol. 3, pp. 2109-2112, Oct, 1993.
- [3] P. Davidsson, "Coin classification using a novel technique for learning characteristic decision trees by controlling the degree of generalization", Ninth International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems, pp. 403-412, 1996.
- [4] M. Nolle, H. Penz, M. Rubik, K. Mayer, I. Hollander and R. Granec, "Dagobert – A New Coin Recognition and Sorting System", Proceedings of the 7th International Conference on Digital Image Computing - Techniques and Applications (DICTA'03), pp. 329-338.
- [5] McNeill S., Schipper J., Sellers T. and Nechyba M.C.. "Coin Recognition using Vector Quantization and Histogram Modeling". 2004 Florida Conference on Recent Advances in Robotics (FCRAR)
- [6] Reinhold Huber, Herbert Ramoser, Konrad Mayer, Harald Penz, Michael Rubik, "Classification of coins using an eigenspace approach", Pattern Recognition Letters (2005), Vol.26, No.1, 61-75
- [7] R. Bremananth, B. Balaji, M. Sankari and A. Chitra, "A new approach to coin recognition using neural pattern analysis" IEEE Indicon 2005 Conference, Chennai, India, 11-13 Dec. 2005, pp. 366-370.
- [8] L.J.P. van der Maaten, P.J. Boon, "COIN-O-MATIC: A fast system for reliable coin classification", MUSCLE CIS Coin Recognition Competition Workshop 2006
- [9] Khashman A., Sekeroglu B. and Dimililer K., "Intelligent Coin Identification System", Proceedings of the IEEE International Symposium on Intelligent Control ( ISIC'06 ), Munich, Germany, 4-6 October 2006, pp. 1226-1230.

- [10] Khashman A., Sekeroglu B. and Dimililer K., "ICIS: A Novel Coin Identification System", Lecture Notes in Control and Information Sciences, Vol. 345, Springer-Verlag, September 2006, Vol. 345/2006, pp. 913-918.
- [11] L.J.P. van der Maaten and E.O. Postma, "Towards automatic coin classification", Proceedings of the EVA-Vienna 2006, Vienna, Austria, 2006, pp. 19-26.
- [12] Thumwarin, P., Malila, S., Janthawong, P. and Pibulweij, W., "A Robust Coin Recognition Method with Rotation Invariance", 2006 International Conference on Communications, Circuits and Systems Proceedings, 2006, pp. 520-523.
- [13] Reisert M., Ronneberger O. and Burkhardt H., "A Fast and Reliable Coin Recognition System", in Proceedings of the 29th DAGM conference on Pattern recognition, 2007
- [14] Kaiping Wei, Bin He, Fang Wang, Tao Zhang and Quanjun Ding, "A Novel Method for Classification of Ancient Coins Based on Image Textures," dmamh, Second Workshop on Digital Media and its Application in Museum & Heritage (DMAMH 2007), pp.63-66.
- [15] Chalechale, A. "Coin Recognition Using Image Abstraction And Spiral Decomposition", 9th International Symposium on Signal Processing and Its Applications, 2007. ISSPA 2007, pp. 1-4.
- [16] Zaharieva, M., Kampel, M. and Zambanini, S., "Image based recognition of coins – An Overview of the COINS project.", 31st AAPR/OAGM Workshop, Krumbach, Austria, 2007.
- [17] Kampel, M. and Zambanini, S., "Optical Recognition of Modern and Roman Coins", Layers of Perception- CAA 2007.
- [18] Shen, L., Jia, S., Ji, Z. and Chen, W.S., "Statistics of Gabor features for coin recognition", IEEE International Workshop on Imaging Systems and Techniques, 2009, pp. 295 - 298.
- [19] Cai-ming Chen, Shi-qing Zhang, Yue-fen Chen, "A Coin Recognition System with Rotation Invariance," 2010 International Conference on Machine Vision and Human-machine Interface, 2010, pp. 755-757.
- [20] Al-Zoubi, H.R., "Efficient coin recognition using a statistical approach", 2010 IEEE International Conference on Electro/Information Technology (EIT), 2010, pp. 1-5.

- [21] Chen, H. "Chinese Coin Recognition Based on Unwrapped Image and Rotation Invariant Template Matching", Third International Conference on Intelligent Networks and Intelligent Systems, 2010, pp. 5-7.
- [22] Gupta, V., Puri, R., Verma, M., "Prompt Indian Coin Recognition with Rotation Invariance using Image Subtraction Technique", International Conference on Devices and Communications (ICDeCom), 2011, pp. 1-5.
- [23] Stergiou, C., and Siganos, D., "Neural Networks" [ONLINE] Available: "[http://www.doc.ic.ac.uk/~nd/surprise\\_96/journal/vol4/cs11/report.html](http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html)", Accessed on Jan 2011.
- [24] B. Yegnanarayana, "Artificial Neural Networks", Eight Printing, Prentice-Hall of India Limited, pp. 306-322, August 2003.
- [25] E. Sackinger, B.E. Boser, J. Bromley, Y. Lecun, and L. D. jacket, "Application of the ANNA neural network chip to high speed character recognition", IEEE transaction on neural networks, vol.3.3 pp. 498-505, 1992.
- [26] William, K. Pratt, "Digital Image Processing", Fourth Edition, A John Wiley & Sons Inc. Publication, pp.465-529, 2007.
- [27] Cornell University Library/Research Department, "Moving Theory into Practice - Digital Imaging Tutorial", [ONLINE] Available: "<http://www.library.cornell.edu/preservation/tutorial/intro/intro-01.html>" Accessed on Nov, 2010.
- [28] Adobe Systems Incorporated, "Photoshop Elements Help and Support - Key concept: Grayscale" [ONLINE] Available: "[http://www.adobe.com/designcenter/photoshopelements/articles/concept\\_grayscale.html](http://www.adobe.com/designcenter/photoshopelements/articles/concept_grayscale.html)" Accessed on Nov, 2010
- [29] Sean McHugh, "Bit Depth Tutorial", [ONLINE] Available: "<http://www.cambridgeincolour.com/tutorials/bit-depth.htm>" Accessed on Nov, 2010.
- [30] Jeong, S., "Histogram-Based Color Image Retrieval", Psych221/EE362 Project Report, March 2001.
- [31] S. N. Sivanandam, S. Sumathi, and S. N. Deepa, "Introduction to Neural Networks using MATLAB 6.0", Vol. 1, Tata McGraw-Hill 2006.
- [32] Roushdy, M., "Detecting Coins with Different Radii based on Hough Transform in Noisy and Deformed Image", In the proceedings of GVIP Journal, Volume 7, Issue 1, April, 2007.

## List of Papers Communicated

---

- [1] Shatrughan Modi and Seema Bawa, “Automated Coin Recognition System using ANN”, Communicated at International Journal of Computer Application (IJCA), New York, USA, Vol. 26, July 11.
- [2] Shatrughan Modi, Seema Bawa, Parminder Singh Reel, “Automated Coin Recognition- Approaches and Techniques”, Communicated at International Conference on Issues and Challenges in Networking, Intelligence and Computing Technologies (ICNICT, 2011), KIET, Ghaziabad (UP), India, 2-3rd September, 2011.