Enhanced Banknote Recognizer for Sri Lankan Currency

E.M.A. Ekanayake

149209H

Faculty of Information Technology

University of Moratuwa

March 2018

Enhanced Banknote Recognizer for Sri Lankan Currency

E.M.A. Ekanayake

149209H

Dissertation submitted to the Faculty of Information Technology,

University of Moratuwa, Sri Lanka for the partial fulfillment of the

requirements of the Degree of Master of Science in

Information Technology.

March 2018

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education.

Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Name of Student: E.M.A. Ekanayake

Signature: Date:

Supervised by:

Name of Supervisor: Mr.Saminda Premaratne

Signature: Date:

Acknowledgement

I owe deep gratitude to my supervisor **Mr. Saminda Premaratne** for his guidance, meticulous attention and unstinted support given to me to drive my research towards success.

A big word of thanks goes to **Mr. Chaman Wijesiriwardana and Mr. B. H. Sudantha**, for being project coordinators, for giving valuable advices, for providing us with required knowledge and guidance in completing the research project.

My sincere thank goes to **Academic Staff** in **University of Moratuwa – Information Faculty** and **colleagues** of my Masters batch for their kind cooperation and friendship right throughout the course.

I wish to express my appreciation to my employer, **INOVA IT SYSTEMS (PVT) LTD** for having provided me a working atmosphere, conducive to proceed with my higher studies.

Finally, I am deeply indebted to my **Parents** and my **Husband** who have always been my most reliable source of inspiration.

Abstract

Automatic bank note recognition has been researched in many countries around the world in the recent past using many new technologies. It has various potential applications including electronic banking, currency monitoring applications, money exchange counters, super markets, rail way ticket counters, gas stations etc.

Many researches or developers have tried to come up with a global solution for this area of research. However, the accuracy of those solutions are tends to depend on the country's currency note characteristics and the identification depends mainly on the extraction of features. Hence, a generalized system is not applicable in every situation. Sri Lankan researches also have been tried implementing solution for Sri Lankan currency. Even though such systems are being used in banking sector in Sri Lanka, they have limitations and need improvements where researches have a space to give solutions. Therefore, the whole purpose of this research is to overcome the limitations and enhance the current automatic bank note recognition system. In addition, proposed system will perform a data analysis and give a recommendation to the Central Bank of Sri Lanka, about the usage of the currencies (money velocity).

Enhanced Banknote Recognizer for Sri Lankan Currency solution will be done for latest series of 'Development and Prosperity, and Sri Lankan Dancers' which released on 4th of February 2011. RBG value analysis, image processing, OCR and Data analysis will use to implement the proposed solution.

Table of Contents

List of Figures

List of Tables

Abbreviations

Abbreviation	Explanation
ANN	Artificial Neural Network
ATM	Automatic Teller Machine
CBSL	Central Bank of Sri Lanka
HSV	Hue, Saturation and Intensity
MODI	Microsoft Object Document Imaging
BR	Blind Recognition
MSDN	Microsoft Developer Network
RGB	Red Green Blue
OCR	Optical Character Recognition
URL	Unified Resource Locator

Chapter 1

Introduction

1.1 Introduction

Automatic bank note recognition has been researched in many countries around the world in the recent past using many new technologies. It has various potential applications including electronic banking, currency monitoring applications, money exchange counters, super markets, rail way ticket counters, gas stations etc. Many researches or developers have tried to come up with a global solution for this area of research it always tends to depend on the country's currency note characteristics and the success of the identification depends mainly on the extraction of features. Hence, a generalized system is not applicable in every situation. Sri Lankan researches also have been tried implementing solution for Sri Lankan currency. Even though such systems are being used in Sri Lanka, they have limitations and need improvements where researches have a space to give solutions.

1.2 Background & Motivation

Automated currency note counting and issuing machines (ATM) are available in the market and also widely used in day to day business activities and tasks [3], [4], mainly in banking sectors and several other financial institutions. These devices are very helpful for counting and issuing currency notes in large quantities, which makes people free from difficulties. Currency notes identification also has a huge impact on today’s modern world, currently most of the countries including Sri Lanka, have done many researches based on this area [5], [8], [9].

Even though the systems are available, implemented systems have drawbacks or limitations to overcome [16]. The currency note recognition systems in Sri Lanka which are used by the banks are not capable of identifying torn or worn currency notes.

Identifying currency notes including torn and worn notes (not damaged notes which are unable to use) would be very useful to automate as most of the activities involved in various places such as money exchange counters, super markets, rail way ticket counters, gas stations etc., in addition to banking sectors.

Below are some of the main concerns which motivates towards the study of bank note recognition.

- With the expansion of modern banking services, automatic means of currency note recognition is significant and vital to provide a better service to the customers of the bank.
  - Able to deposit money to banks independently without the interaction of a bank officer and also customer can use the deposited money at the same time.

- There are different other areas which involves with monetary transactions and require human intervention. (i.e. super markets, rail way ticket counters, gas stations). Proposed system can simply replace human interaction.

- Current transactions in businesses, money is open and visible to outside people. It can lead to many issues. If money can be kept in a secure location and people allow to ly the amount of money relevant to their transactions, it would be very helpful in this situations.

- The requirements for an automatic currency note recognition system offered many researches to build up robust and dependable techniques.

When it comes to the bank notes identification, 'money velocity' is plays an important role. If the bank note is discolored or used, we assume that the currency is having high money velocity. In Sri Lanka, Central Bank of Sri Lanka(CBSL) is responsible for issuing and replacing damage note. Currently CBSL uses manual process to identify the faded or damaged notes. This is good development area to include in the current bank not recognition systems where CBSL can speed up the manual task.

1.3 Problem Statement

Despite a decrease in the use of currency due to the recent growth in the use of electronic financial transactions, real money transactions remain very important in the market. When performing transactions with real money; counting notes by hand is still a common practice in daily life. Various types of automated machines (ATMs), counting machines should require to avoid the common practice in terms of safe and large-scale transactions.

Sri Lankan Currency note identification faces few limitations currently; identifying torn and worn notes is an impossible task in money deposit machines of Sri Lankan banks. Identifying torn and worn notes manual process need to speed up using automatic system. Currently high-experienced retired senior person can identify 20 bundles per day (a bundle includes 1000 notes). Main drawback is, if the above mentioned person is not available, the CBSL cannot complete the 20 bundles per day. Hence, the importance of a system is critical at this stage to speed up the manual process.

CBSL is issuing bank notes several times in a year as per the requirement of the country. Identified damaged notes can give a valid requirement to reprint the same note. If we can give a recommendation using data we have collected, it will be easy for them to decide whether reprint the identified note as a bulk at once or to use any other material to print the note or to introduce any other new note.

1.4 Aim and Objectives

Primary aim of this research is to develop an efficient and accurate Bank Note Recognition System which can identify Sri Lankan Currency Notes including torn and worn notes which will help to enhance the other co-related systems.

Objectives can be defined as below.

- Perform a Data analysis to identify mostly used denomination and
- Provide recommendation to CBSL for issuing/reprinting new notes
- Minimize the current drawbacks in identifying torn and worn notes at CBSL
- Speed up the manual process of identifying torn and worn notes with high accuracy

1.5 Approach of the Enhanced Banknote Recognizer for Sri Lankan Currency

1.5.1 Approach of identifying Bank notes including torn and worn notes

Enhanced Banknote Recognizer for Sri Lankan Currency solution would be done for latest series of 'Development and Prosperity, and Sri Lankan Dancers' which released on 4th of February 2011. RBG value analysis, image processing and OCR techniques will be used to implement the proposed solution.

1.5.2 Approach of Data Analysis

Perform a data analysis to automate manual process of identifying unusable denomination. This can be performed using RGB value analysis.

Steps:

- Evaluate the bank note by users (touch and feel) whether to Replace or Not. Save the Results.
- Scan the bank notes and get the RGB results of the scanned images, using the system.
- Evaluate and compare the above results. Identify a RGB range where the denomination is highly used.
- Using the identified range system can identify all the scanned images are (same denomination value) able to use more or in need of replace.

1.6 Structure of the Thesis

The rest of the Thesis is organized as follows. Chapter 2 reviews the literature on bank note recognition and Data analysis. It will help to identify the research problems. Chapter 3 is about the technology used to develop the proposed system. Chapter 4 presents our new approach to develop Enhanced Banknote Recognizer for Sri Lankan Currency solution. Chapter 5 and Chapter 6 describe the design and implementation of the system respectively. Chapter 7 is about evaluation and testing of the new solution. Lastly chapter 8 refers the conclusion of the overall system. Further References and Appendixes are described at the end of the Thesis.

Chapter 2

Developments and Drawbacks in Current Bank Note Recognition Systems

2.1 Introduction

Chapter 1 gave a comprehensive description of the overall project described in this thesis. This chapter provides a critical review of the literature review in relation to developments in similar researches and its drawbacks.

2.2 Literature Survey

Identification of significant monetary features from a currency note is a vital task when designing an accurate and effective currency note recognition system.

Feature extraction process and identification of size differences of currency notes can be considered as a straight forward method where you can easily determine the width and the height of bank note image in pixels and hence, allow you to distinguishably identify notes. Latest currency note series which issued in year 2011, can also give correct results as it contains unique widths [1], [2]. Edges and boundaries of notes get fold, torn or worn easily (Figure 2.1), due to high circulation of the note in public. Getting the Size of the note cannot be considered as the best solution; hence we need to consider few more features of the bank note.

Figure 2.1 - Torn & Worn out Bank Notes/Folded Notes

RGB color value analysis can also be used as another method to recognize bank notes, as 12th series of Sri Lankan Bank note has distinct colors for each denomination [1], [2]. Similar to the issues mentioned previously, colors also can be faded due to high circulation. Hence, it is not a good practice to identify the bank note, only using the RGB color values (Table 2.1).

Bank Note	**Predominant Color**
Rs. 20	Maroon
Rs. 50	Blue
Rs. 100	Orange
Rs. 500	Purple
Rs. 1000	Green
Rs. 5000	Gold

Table 2.1 - Predominant Color of each denomination

To read other features in the bank note, we may need additional technologies and methods. We can identify the edges of the features in the bank note, using Edge Detection method[7]. This is more challenging task as a bank note contains many number of designs, numbers and letters. Eg: Rs. 20 Note has a drawing of the view of the Port of Colombo as seen now and in history. The bird that appears on the right is the Sri Lanka Serendib Scops Owl (Otus thilohoffmanni). The butterfly that appears on the lower left corner is the Baronet (Symphaedra nais) (Figure 2.2).

Figure 2.2 - Features in Rs.20 bank note

In the edge detection process the dimensions of data is reduced. This is almost always necessary, due to the limitations in computational power and hardware resources. When trying to identify too many objects might increase the cost of the system and it might lower the system performance.

Hence it is an important task to select only important components which are easy to identify digitally.

Before continue with the System design, it is worth to study and understand specific features that can be seen in Sri Lankan Bank notes. The CBSL official web site describes current and new advanced security features that contains in new Currency Note Series issued in the year of 2011 by CBSL. Web Site explains important information regarding security features such as, Note Printing Paper, Watermark, Security thread, Intaglio Prints (Raised print area), See-through Features, Cornerstone, Extra small text, Blind recognition feature and Color Fibers etc. [6]

The purpose of adding different types of features into a bank note is to identify the note easily by the human eye. When a person sees a bank note, his eyes tends to find those features, color and the printed value altogether. In Sri Lankan currency notes, there are very complex images that include ancient objects, symbols, animals and constructions. An analysis has been carried out by D. A. K. S. Gunaratna, N. D. Kodikara and H. L. Premaratne for their research "ANN Based Currency Recognition System using Compressed Gray Scale and Application for Sri Lankan Currency Notes – SLCRec" to get an idea as to how Sri Lankan citizens identify notes. Educated people of undergraduates, graduates and research students have been participated for this survey and obtained results are shown the in the Table 2.2. According to their findings, most of the people recognize Sri Lankan currency notes by color and overall image composition [5]. This information would be very helpful when designing the currency note recognition system.

Attention	Preference	%
Only look at a part of the note	5	9
Look for hidden patterns	2	3
Look at the printed value	8	14
Look at the overall image composition	13	23
Identify by color	17	30
Look at the size of the currency note	6	10
Feel the currency note	5	9
Look for a specific image	1	2

Table 2.2 - Human Identification Pattern

Researchers have proposed various approaches and methodologies to recognize currency notes in different countries. Most of the research works have been carried out using single feed-forward neural network (NN) for the recognition [5], [8].

In another technique for paper currency recognition, the Markov chain concept [9] has been employed to model the texture of paper currencies as a random process. The method is used for recognizing paper currencies from different countries. In this method only intact examples of paper currencies from each denomination are used for training the system. The system is able to recognize 95% of data correctly from 100 denominations from different countries.

Trupti Pathrabe, Mrs. Swapnili Karmore have done a research on Indian Currency note recognition using the properties of the HSV (Hue, Saturation and Value) color space with emphasis on the visual perception of the variation in Hue, Saturation and Intensity values of an image pixel [10]. In this technique, Fitting tool of Neural Network is used for the purpose of paper currency verification and recognition. Crucial features from Indian banknotes were extracted by image processing and experimented on Neural Network classifier.

Another study describes an approach of using a simple statistical test for the verification step, where Gaussian distribution is employed [11]. The propose using the probability density formed by a multivariable Gaussian function, where the input data space is transferred to a lower dimensional subspace. Due to the structure of this model, the total processing system acts as a hybrid neural network. The method and the numerical experimental results are shown by using the real data and the recognition machine.

U.S.S. Perera and D.N Balasooriya have conduct a research for Sri Lankan Currency note recognizer for visually impaired people. They have tried to develop low cost, portable, hand held bank note recognizers for visually impaired target group only[18]. In Sri Lanka, one currency note recognition device has been designed and implemented as a final year undergraduate project and this system employs a light dependent resister array located at various points over the bank note's area [17].There are pre-determined reference intensity levels defined to detect each color in the bank notes.

2.3 Problem Definition

As listed above in detail, many researches have been conducted to identify currency notes in different countries. However, the research "ANN Based Currency Recognition System using Compressed Gray Scale and Application for Sri Lankan Currency Notes – SLCRec" by D. A. K. S. Gunaratna, N. D. Kodikara, and H. L. Premaratne [5], can be considered as the only successful research that have been carried out to recognize Sri Lankan currency notes. But none of the researchers were able to identify the torn and worn out notes. Similarly, the drawbacks in current manual system in CBSL are not visible to other parties or not documented, unless a responsible party will inform the current issues of the system. Perform a data analysis to identify the mostly used denomination, is having a significant requirement for the internal usage of the CBSL.

2.4 Summary

This chapter presented a comprehensive literature review on the Bank Note Recognition Systems globally and locally. Identified research problem have been clearly described in the section 2.3. Next chapter will discuss the technology and methods to be used in proposed solution.

Chapter 3

Technologies adopted in developing Enhanced Banknote Recognizer for Sri Lankan Currency System

3.1 Introduction

Chapter 2 gave a detailed view of the in relation to developments and its drawbacks in Current Bank Note Recognition Systems. This chapter provides technologies that are planned to adopt for the proposed system development.

3.2 Technology Adopted

Technologies adopt for the Enhanced Banknote Recognizer for Sri Lankan Currency System are being described in below table 3.1.

Objective 1	Enhanced bank note recognizer: Limitation of identifying torn and worn out notes in money deposit machines
Methodology	– Using RGB color – Using diagonal of each bank note (Rs.20/Rs.50/Rs.100/Rs.500/Rs.1000/Rs.5000) – Using Blind Recognition value – Using Serial Prefix – Using Artistic Impression – Using Template Matching to identify the numbers and images
Technology Used	Properties of an image: Pixel and RGB value Image processing techniques – Feature extraction – Edge detection – Filters

	OCR techniques
Objective 2	Identify and give recommendation of the mostly used denomination in the Sri Lankan society
Methodology	RGB value analysis
Technology Used	Properties of an image: Pixel and RGB value Image Processing
Objective 3	Speed up the manual process of identifying torn and worn out notes
Methodology	RGB value analysis
Technology Used	Properties of an image: Pixel and RGB value Image Processing

Table 3.1 - Objectives Vs Technology Adopted

A Pixel is the smallest addressable element of a digital image and users cannot see the actual pixels of a digital image. All the transformations and modification that are done in Images processing, would finally affect to pixel. Each pixel's color sample has three numerical **RGB components** (Red, Green, Blue) to represent the color of tiny pixel area. These three RGB components are three 8-bit numbers for each pixel. Three 8-bit bytes (one byte for each of RGB) are called 24 bit color. Each 8 bit RGB component can have 256 possible values, ranging from 0 to 255. For example, three values like (250, 165, 0), meaning (Red=250, Green=165, Blue=0) to denote an Orange pixel.

In **image processing**, feature extraction is a special form of dimensionality reduction. When the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant then the input data will be transformed into a reduced representation set of features. Feature Extraction is important in this research, since it is dealing with highly distorted currency images, where removing noise on them is very much required. Edge detection is a fundamental tool in image processing and computer vision, particularly in the areas of feature detection and feature extraction, which aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. There are many ways to perform edge detection. However, the most may be grouped into two categories, gradient and Laplacian. Since there are lots of objects and shapes on Sri Lankan currency notes, which are unique to each denomination of notes, use of edge detection techniques would immensely helpful for this research. Image processing filter is a method that used to suppress either the high frequencies in an image (i.e. smoothing the image) or the low frequencies (i.e. enhancing or detecting edges) in an image [19]. Image processing Filters would greatly helpful for Feature Extraction process [20].

3.3 Summary

As explained the 3.2 section, this chapter described what are the technologies are being used in this system and why it has to use the same technology. Next chapter describes the overall design of the proposed system.

Chapter 4

Approach of Enhanced Banknote Recognizing for Sri Lankan Currency

4.1 Introduction

In this chapter, we describe our approach to develop optimized bank note identification. This system improves the accuracy and efficiency of the bank note identification. Similarly this chapter describes the approach of obtaining other objectives mentioned in the chapter one. The approach is further elaborated in sections: input to the system, output of the system, process which converts input to the output, users of system and features of the system.

A reliable and accurate currency recognition system is important for the automation of tasks in different sectors such as banking services, shopping malls, vending machines, rail way ticket counters, filling stations and currency exchange services etc.

4.2 Input to the System

Input to the system would be scanned image of a bank note which depicts the latest series of Development, Prosperity and Sri Lanka Dancers (Rs.20/Rs.50/Rs.100/Rs.500/Rs.1000/Rs.5000).

For Data analysis, system should be able to accept input data of RBG value analysis. Data should be preprocessed before entered and should identify the valid data range for a particular denomination which depicts the high usage.

4.3 Output of the system

Result of identifying bank notes including torn and worn out notes. Similarly output of the overall product should include the results of a mostly used denomination of the latest bank note series.

4.4 Process

The currency notes issued by Central Bank of Sri Lanka have been segmented mainly into five categories.

1. Clean notes of new currency note series issued in February 2011, which is the Twelfth series of Sri Lankan Currency Notes [3].
2. Old, torn and worn notes of new currency note series issued in year 2011(Figure 4 .1)
3. Clean notes of previous release.
4. Old, torn and worn notes of previous release.
5. Commemorative notes with limited distribution. (i.e. Commemorative note issued by the CBSL to mark the ushering of peace and prosperity to Sri Lanka (Figure 4.2)).

Figure 4.1 - Old, Torn and Worn Currency Note

Figure 4.2 - Commemorative Note

This research and implementation work would be categorized into several steps as below.

Step 1 – Recognize valid currency notes belongs to 1^{st} category mentioned above. Here, I will assume side and the direction consider are same for all notes. (It will recognize notes from a pre-defined side and direction)

Step 2 – Recognize valid currency notes belongs to 2^{nd} category mentioned above. As in the Step 1 it will also recognize notes from a pre-defined side and direction.

Step 3 – Recognize valid currency notes belongs to 1^{st} and 2^{nd} category mentioned above from either direction but same side. (Figure 4.3)

Step 4 – Recognize valid currency notes belongs to 1^{st} and 2^{nd} category mentioned above from both side. (Figure 4.4)

Step 5 – Recognize valid currency notes belongs to 3^{rd} category mentioned above from either direction and both side.

Step 6 – Recognize valid currency notes belongs to 4^{th} category mentioned above from either direction and both side.

Step 7 – Recognize valid currency notes belongs to 5^{th} category mentioned above from either direction and both side.

In this research I will mainly focus on implementing currency notes recognition system which full fill the specification mentioned in 'Step 1' above. However, one of the major concerns in Sri Lankan currency notes is the exposure to high rate of distortion when in circulation (Figure 4.1). Hence, it is important if proposed system could give accurate results, by identifying, torn or worn bank notes to an important precision. I will give a same level of weightage and attention to cover step 2 as well. I would give less attention for Step 3 to 7, as they are in previous version releases and will not print again by CBSL.

Figure 4.3 - 100 Rupee Note

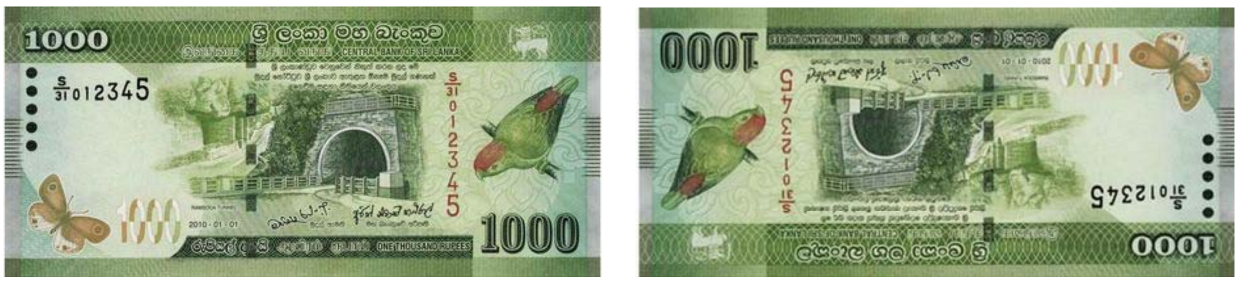

Figure 4.4 - 1000 Rupee Note

It needs to be noted that the proposed method may not be able to differentiate genuine currency notes from counterfeits and it is beyond the scope of the research. The techniques which utilize infrared or ultraviolet spectra may be applied for discriminating between genuine and counterfeit banknotes.

Enhanced Banknote Recognizer for Sri Lankan Currency solution would be done for latest series of 'Development and Prosperity, and Sri Lankan Dancers' which released on 4th of February 2011. RBG value analysis, image processing and OCR techniques will be used to implement the proposed solution.

Data that have been collected from the system are subjected for preprocessing and are loaded into the MS Office Excel and perform comprehensive data analysis. Following are the steps of data analysis.

Data Selection & Integration: the data are taken from the system using RBG value analysis

In the Data Analysis, initially entering scanned notes to the system, should identify the data range of which the soiled notes are grouped.by using that identified range we can add any number of notes of same value and decide percentage of usage of the bank note. Follow the same process for all the denominations, then analyze which denomination has been mostly used.

4.5 Users of the System

System can be used by any person who is aware of handling software systems and who needs to identify the bank note including torn and worn out notes.

- Users of banking sector (mainly the CBSL)
- Users of any other financial sectors
- Users who use Money exchange machines
- Users in shopping malls, vending machines, rail way ticket counters, filling stations

4.6 Features of the System

Features of the Enhanced banknote recognition system are described under 4.6.1 and 4.6.2.

4.6.1 Functional Requirements

- Recognize valid currency notes issued by Central Bank of Sri Lanka.
- Recognize torn or worn currency notes to an important precision.
- Obtain a good accuracy percentage and success rate for currency note recognition.
- Data Analysis for CBSL money issuing
- Speed up the manual process of identifying torn and worn notes in CBSL

4.6.2 Non-functional Requirements

- Performance / response time

  Algorithm should be able to recognize reasonable amount of currency notes within a given time with a fair usage computational power and other necessary resources.

- Modifiability

  Algorithm or the application should be able to use to recognize any series of currency notes, which would be released in future, with fewer amounts of modifications.

- Reliability

  Algorithm should give accurate and correct results for an important precision.

- Security

  Since this system deals with money, necessary actions should be taken to protect the algorithm and stored data against danger, damage, loss, and crime.

4.7 Summary

This chapter describes our approach to develop the proposed system. Here we have mention input to the system, output from the system, processes, users and features of the system. Next chapter describes the design of our solution.

Chapter 5

Design of the Proposed Solution

5.1 Introduction

This chapter describes the design of the solution for the process presented in the approach chapter. We have designed the solution as two-tier architecture; a client–server software architecture with a backend database. Here we describe the role of each component in the architecture.

5.2 Top Level Architecture of the System

The top-level architecture of the proposed system comprises of user interface module and a database. Figure 5.1 illustrates the Top Level Architecture.

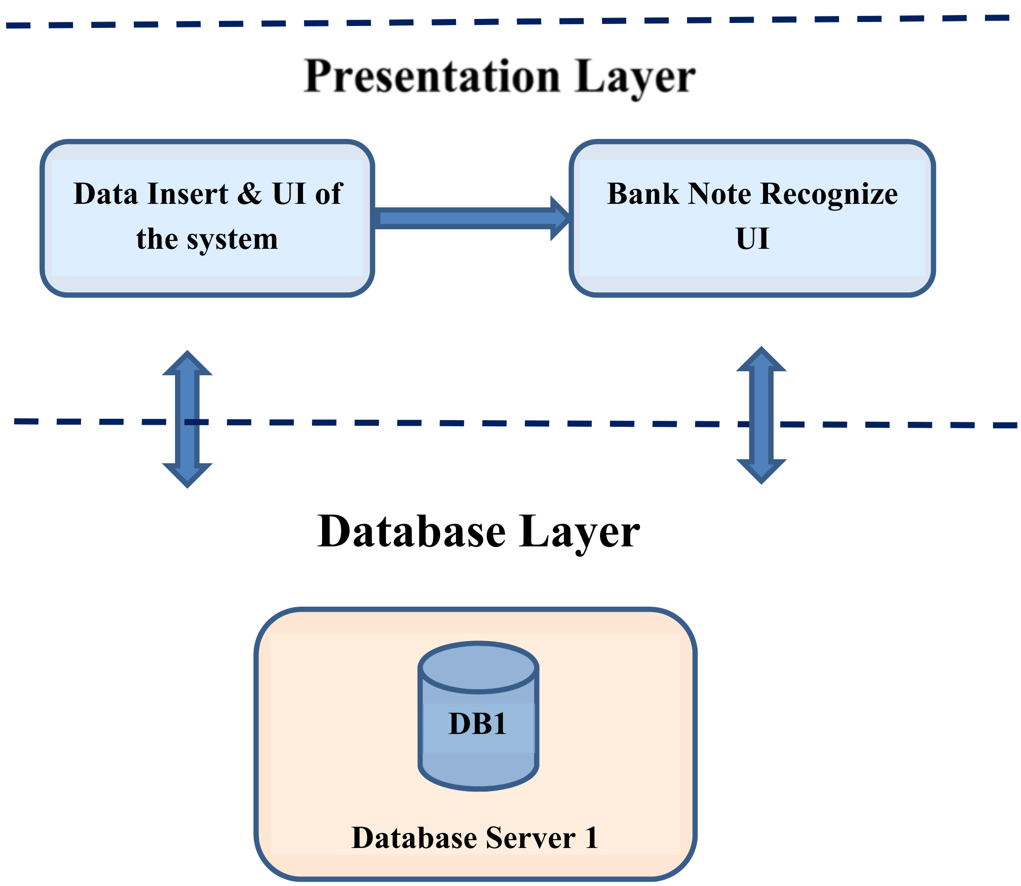

Figure 5.1 - Illustrates the Top Level Architecture

5.3 High-Level Design Flow of the Enhanced Bank Note Recognizing System

Implementation of the proposed system would be carried out based on steps depict in following flow diagram (Figure 5.2).

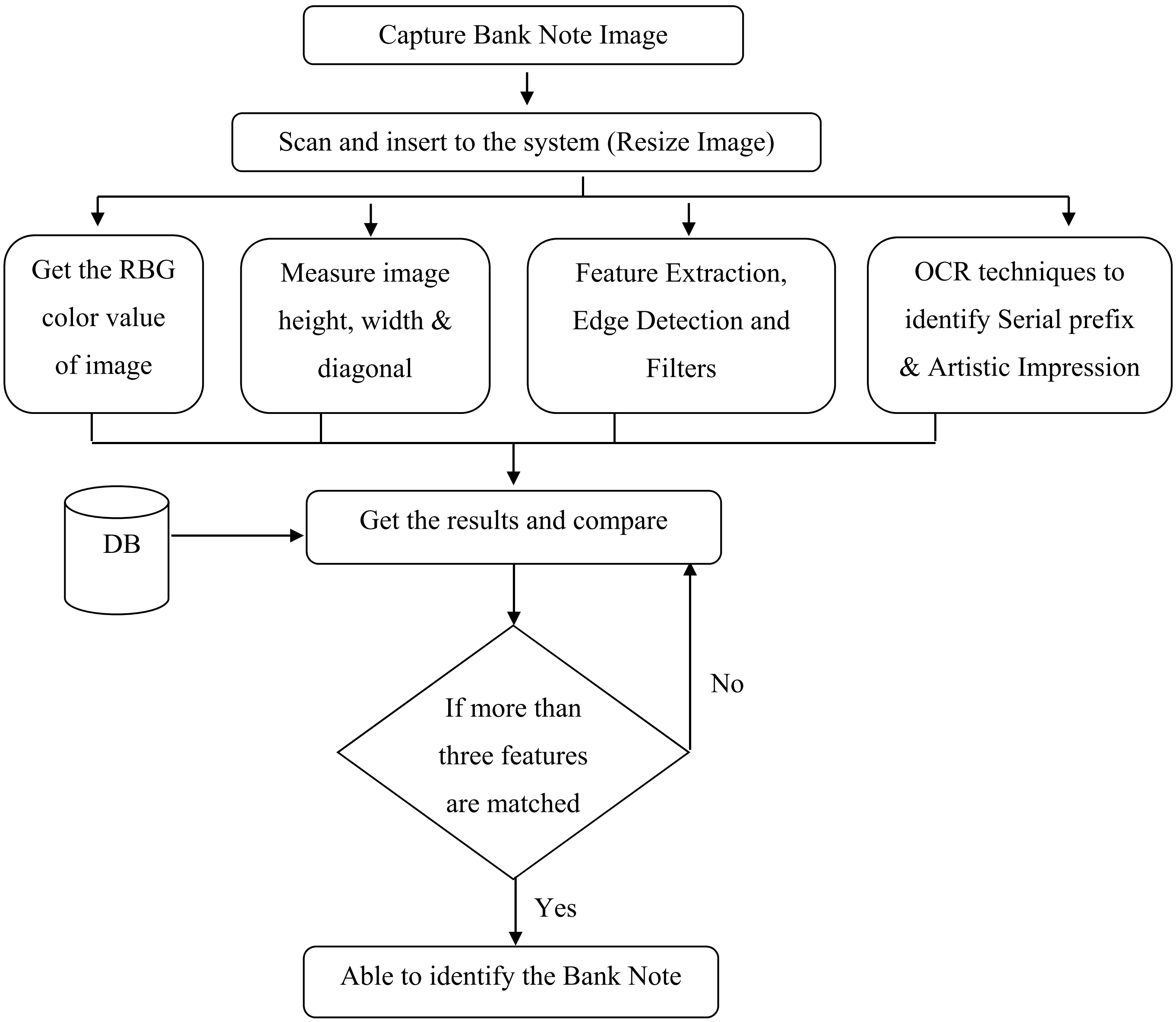

Figure 5.2 - High-level Design Flow of Proposed System

5.3.1 Capture Bank Note Image

A color scanning machine would be used to capture tangible currency notes and digitally store image of the currency in computer system. However, it needs to be noted that, capturing of notes using the scanner and transferred to computer system is just a manual process, which means that scanning machine or the scanning process is not a part of the proposed system. Therefore, pre-captured currency notes would use only for demonstration purpose.

5.3.2 Resize Image

Due to limitation of computational power and also to obtain better performance, size of the captured currency image would be reduced to certain level while maintain visual quality of the image. When reducing the size number of pixels would be reduced (in other words resolution, or sharpness, of the image would be reduced).

5.3.3 Compare using color value and image size

In the proposed system, color variation and size differences of captured currency note would be verified as the first step to distinguishably identify notes. This method would give fairly accurate results for new currency note series issued in year 2011 by CBSL, since they have distinct widths. But this may not be accurate enough to recognize fold, torn or worn currency notes. Due to this reason, system will first verify color and size of the currency note and would determine whether that note can clearly and accurately differentiate from other note types. If the system couldn't clearly identify the class of the note type which it belongs to, it will use advanced methods and algorithms to recognize notes.

5.3.4 Identified Techniques

5.3.4.1 Feature Extraction

Feature extraction of images is challenging work in digital image processing. But, extracting sufficient monetary characteristics from the currency image is essential for accuracy and robustness of the automated system. Extracting too many features will increase the cost and it will lower the system performance. Therefore, only the critical features that are easy to extract but difficult to imitate would be chosen.

For example (1) Currency value of the note (numeric values 5000 in 5000 Rupee note), (2) Blind recognition feature (six vertically ordered heavily printed dots appears on the left hand side of the 5000 Rupee note), (3) Serial Prefix which is unique to each note (letter ‘R’ in 5000 Rupee note) and Artistic impression (4) on Bank note image can be considered (Figure 5.3).

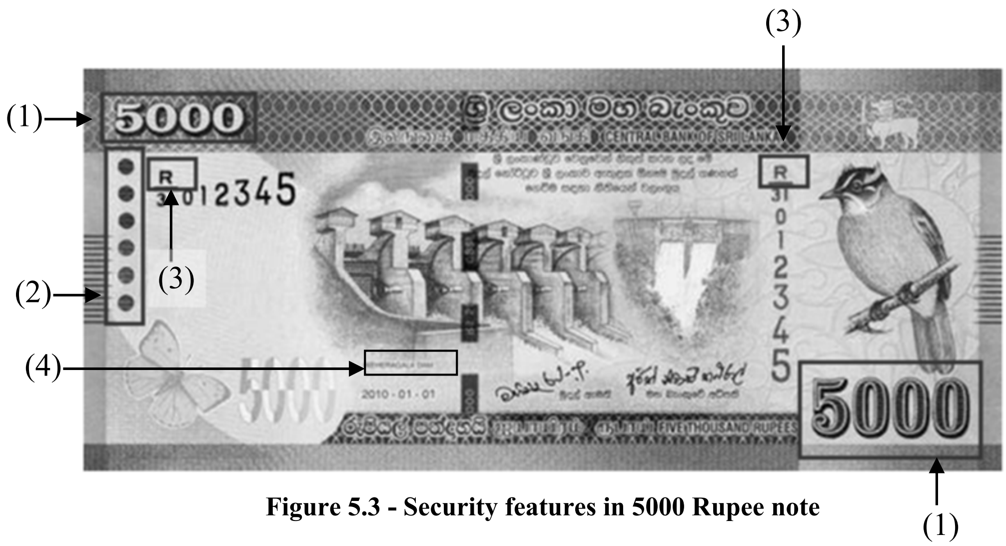

Figure 5.3 - Security features in 5000 Rupee note

5.3.4.2 Edge Detection

Because of circulation, edges of the currency notes are usually worn and torn when they get old. Therefore, to overcome the problem of recognizing noisy currency notes, need to apply a suitable pre-processing step. Edge detection is a fundamental tool in image processing and computer vision, particularly in the areas of feature detection and feature extraction, which aim at identifying points of a digital image. Edge detection reflects sharp intensity changes in colors of the image.There are many ways to perform edge detection. Gradient and Laplacian are the mostly used methods. Since there are lots of objects and shapes on Sri Lankan Bank notes, which are unique to each denomination, the use of edge detection techniques will be helpful to success the research in terms of accuracy.

5.3.4.3 Image Processing Filters

Image processing filter is a method that used to suppress either the high frequencies in an image (i.e. smoothing the image) or the low frequencies (i.e. enhancing or detecting edges) in an image. Image processing Filters would greatly helpful for Feature Extraction process. Invert, Smooth, EdgeDetectDifference and EdgeEnhance filters can be used to purpose.

5.3.4.4 OCR Techniques

OCR (Optical Character Recognition) is the mechanical or electronic conversion of scanned images of handwritten, typewritten or printed text into machine-encoded text. OCR involves scanning and comparison techniques to identify printed text or numerical data.

Reading text and words which is unique to each denomination can be used as another parameter to differentiate and recognize currency notes.Serial Prefix (3) and text which describes the Artistic impression (4) on image (Figure 5.3) are some of the unique characters in twelfth series of Sri Lankan bank notes. A proper OCR tool can be used to read text on currency image and MODI will be used in this case.

5.3.5 Final Result

The features extracted in the each steps mentioned above will be used in the system for comparing and recognizing bank notes.

5.4 Summary

The design has been described with main focus on major techniques that are going use in the system. The illustrated design has been used for the implementation, which is described in next chapter.

Chapter 6

Implementation of Enhanced Bank Note System

6.1 Introduction

This chapter describes the implementation of the proposed system. General design concept has been described in the previous chapter, including the system and software design will be implemented in this chapter.

6.2 Overall Solution

6.2.1 Software and Hardware Requirement for the Implementation

A system has many Software and Hardware resources, and following mentioned have been used to develop the Enhanced Bank Note Recognition System.

- **Microsoft Visual Studio 2012 (C#.NET)**

MS Visual Studio 2012 used as it contains enhancing features, such as IntelliSense technology, auto-colorization, auto-completion, method lookup, syntax and type checking, code re-factoring, and many more. It provides an integrated set of tools which facilitated the requirements of proposed system like image processing and OCR operations.

- **Microsoft Office Document Imaging (MODI)**

MODI is a Microsoft Office application that allows users to scan images, view and convert scanned images to text under program control, using its built-in OCR engine. It was first introduced in Microsoft Office XP and is included in later Office versions including Office 2007. It is no longer available in Office 2010, although it is possible to install MODI from a previous version of Microsoft Office along with Office 2010 .

- **SharePoint Designer 2007**

Microsoft Office Document Imaging (MODI) has been removed in Microsoft Office 2010. As a solution, we can install SharePoint Designer 2007, since it provides methods to adapt MODI on the computer. However, other alternative methods are also there which helps to regain the functionalities of certain MODI features.

- **Microsoft SQL Server 2014**

SQL Server 2014 is able to support older computer hardware, and works well with newer hardware designs. It facilitates faster queries and less wait time for user when search information from an application.

- **Microsoft Office Excel 2016**

MS Office Excel enabled to turn data into information with powerful tools to analyze. This has been used with the data analyzing part.

- **Hardware Resources**

Canon Pixma E500, multifunction Inkjet Printer has used as currency notes scanning machine and a computer have been used mainly as the hardware resources. Intel Core i5 – CPU 2.6 GHz or higher processor (64 bit Operating System), 4 GB RAM or higher , 2 MB Cache memory,4 GB of available Hard Disk space, VGA (1280 x 800) or higher-resolution are some of the highly required components in the used PC.

6.2.2 Implementation flow of the Currency Recognize Feature

- **Get the Scanned Currency Image**

Scanning the currency note is the first step in the Enhanced currency note system, color scanning machine has been used to capture actual currency notes. Above mentioned scanning machine, has been used for this purpose. Capturing of notes using the color scanning machine and transferred to computer system is a manual process in this research. Scanning machine or its process is not a part of the developed system. However, users have to browse the saved currency notes in order to use them in the system.

- **Get the Height/Width of an Image**

Currency notes issued by CBSL as Twelfth series, have distinct widths, calculation of diagonal values (Refer below formula) in pixels (Refer Chapter 3, Section 3.2) can be used as another parameter to identify notes specifically. Due to high circulation of currency notes, boundaries of notes get fold or damaged easily and size of the note will get change. Hence, considering only the width and height of the notes, are not enough to identify currency notes.

$$D = \sqrt{w^2 + h^2}$$

where:
D is the Diagonal of the currency note
w is the width of the currency note
h is the height of the currency note

- **Resizing Image as required**

To obtain better performance, size of the captured currency image is resized to certain level while maintain visual quality of the image. When reducing the size, number of pixels would be resized (in other words resolution, or sharpness, of the image would be reduced).

- **Get the R,G,B values of an Image**

Every pixel has a red component, a green component and a blue component which is named as RGB value (Refer Chapter3, Section 3.2). Therefore, when given an image, we can calculate the mean value of Red, Blue and Green according to a formula, which gives very closer values for each denomination of currency notes. Color variation of captured currency note would give fairly accurate results for new currency notes. Old currency notes, due to high circulation and over usage, colors of some denomination of notes tend to overlap with each other. Hence use of only the RGB values to differentiate or identify notes will not be enough.

- **Get the Blind Recognition(BR) value**

Above mentioned features will not always give accurate results for used currencies. Hence additional methods need to be used to identify currency notes. One solution to identified issue is to get the Blind Recognition value (Figure 6.1). Image processing methods like Feature extraction and Edge Detection (Refer Chapter 3, Section 3.2) have been used for this purpose. Similarly, several image processing filters has been used without effecting the basic features of an image.

Figure 6.1 - Blind Recognition Circles

- **Get the Read Serial Prefix and Artistic impression**

Another parameter that we can use is reading text and words which is unique to each denomination of currency note. Twelfth series of Sri Lankan currency notes, Serial Prefix (1) and text which describes the Artistic impression (2) on currency image (Figure 6.2) are some of the unique characters that can be used. OCR tool with MODI has been used to read text on currency note image.

Figure 6.2 - Serial Prefix (1) and Artistic Impression (2)

- **Finally Read Currency Note**

Using above mentioned parameters can be used in Enhanced Currency Notes Recognition system and the extracted features would be used in an intelligent system for comparing and recognizing currency notes. Those parameters are mean value of Red, Green, Blue color values, diagonal of the note, Blind Recognition Value, Serial Prefix and Artistic Impression. In this comparison process, mean RGB values and the diagonal of the note will be compared with the values stored in the Database.

As the system output, it will show the recognized currency notes with its above mentioned parameter values.

6.3 Implementation of the System Functionality

This section depicts the functional behavior of the system with the used algorithms etc.

There are four main forms, Base interface, Identify Basic Data interface, System Learning interface and Currency Note recognition interface. Functionality of the each form is described in the below section.

6.3.1 Base Interface

This interface has navigation to main functionalities of the system. It contains Bank Note Setup button to navigate to Bank Note Setup interface to add currency with the Serial prefix and value.

Currency Recognizer button to navigate to Currency Recognizer interface. Close and Help buttons are also available in the main Base interface (Figure 6.3).

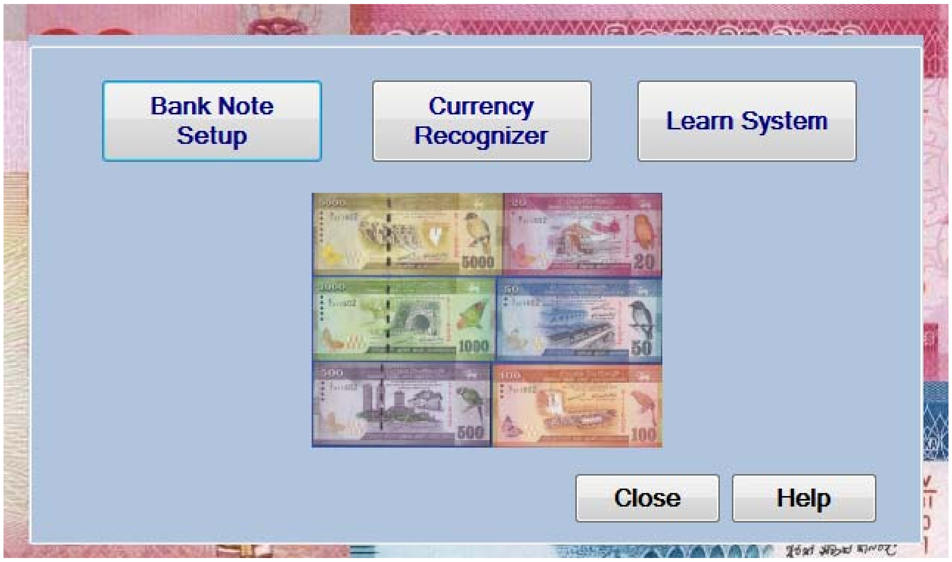

Figure 6.3 - Main Base interface

6.3.2 Identify Bank Note Set up interface

Adding a Record

Add Currency value from the dropdown and add serial prefix from a dropdown, add description if required. Click 'Add' button to save the data.

Relevant data will be stored in the system database and also displayed in the data-grid.

Deleting a Record/Clearing the Selected Record

Double click on the relevant record (to be cleared/deleted) in the data-grid. Then the information of the note would be mapped to relevant data fields. (Figure 6.4)

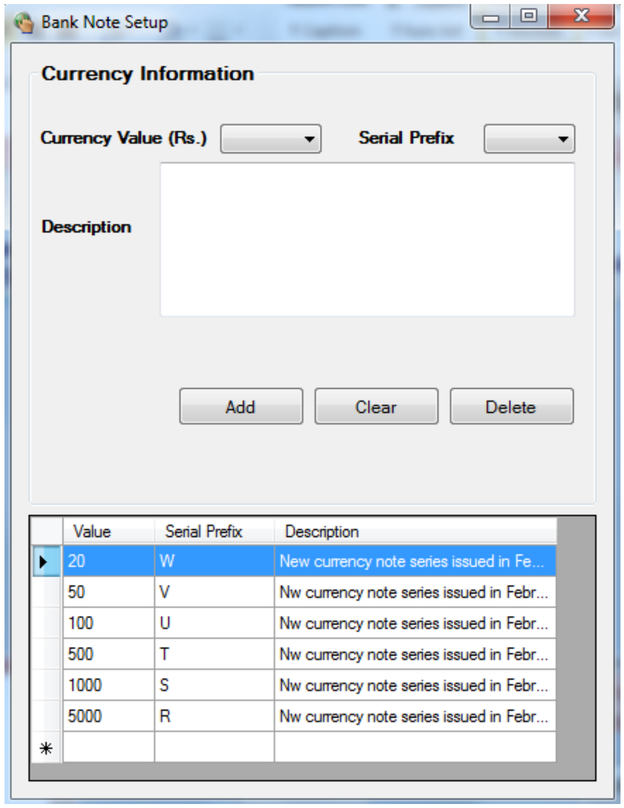

Figure 6.4 - Bank Note Set up interface

6.3.3 System Learning

Learn the system for a given denomination of note

- Select the relevant denomination of currency note from the drop-down list. If the value does not exist in the dropdown, user can add as a new currency using 'New Currency' button.

- Click on 'Browse Currency' and select multiple notes (user can select a single note at a time also. However, it is recommended to select multiple notes) belongs to same denomination(Figure 6.5).

- Statistics of each selected note (Diagonal and RGB values) would be listed in the data-gird. Calculated mean diagonal and mean RGB values of the selected set of notes would be mapped in to relevant data fields. Double click on relevant record on the data-grid or use track-bar to navigate through the records of each currency note.

- Click on 'Save Data' button to train the system. System will process the data by going through an algorithm (Refer Chapter 6, Section 6.4.2).
- Information messages would be given if current data is overlapped with previously trained data. User has the options to take necessary action, whether to add the data to the system without considering overlap data or not.
- A record of the saved denomination of note would be added to data-grid 'Statistics of Saved Currency Notes' and Result Log will be updated.
- User can clear the data saved in Data grid using 'Delete Data' and ' Clear Records' buttons.

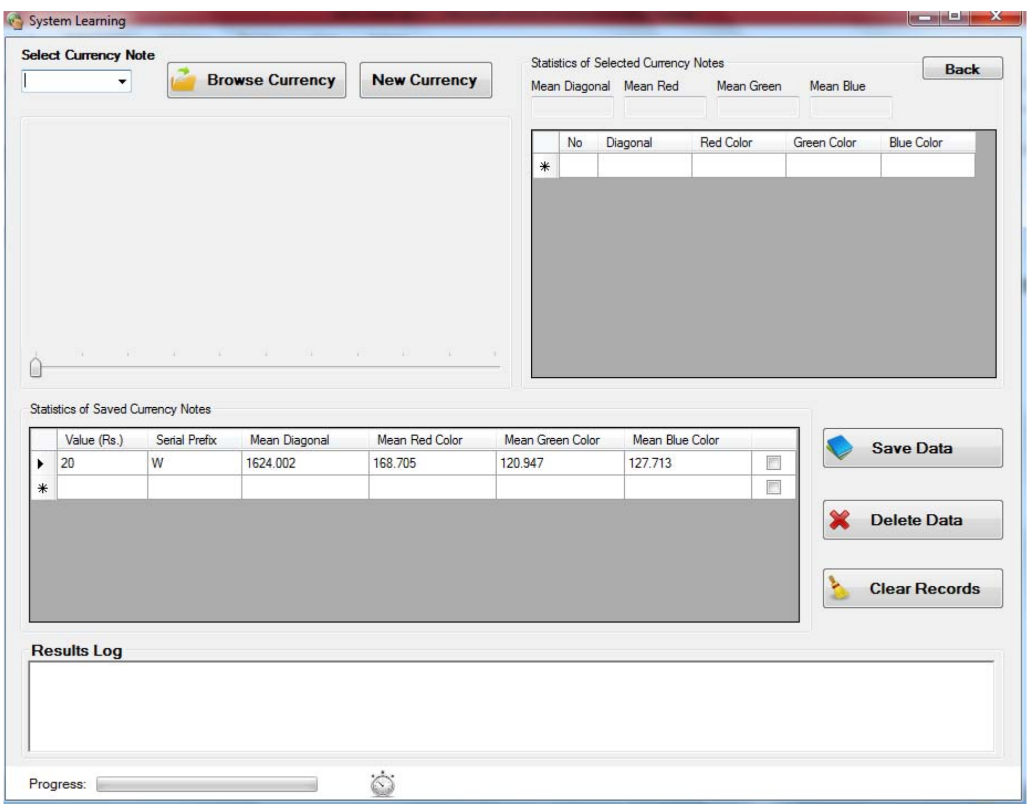

Figure 6.5 - System Learning interface

6.3.4 Recognize Currency Notes

Main operation is to recognize any given currency note(s). If the notes are trained for diagonal and RGB values, system will give more accurate results. Hence, it is recommended to train notes before recognizing notes. This operation is facilitated by Recognize Currency window.

- Check or Uncheck the ‘Save Blind Recognition Image’ check box as needed. (If user select the check box it will display blind circle outlined image of the note, upon selecting and recognizing currency notes) (Figure 6.6)
- Click on ‘Browse’ and select multiple notes (user can select a single note at a time also).
- Statistics of each selected note (Serial Prefix, Diagonal, RGB values, BR Value and Artistic Impression) would be recognized by the system and listed in the data-gird along value of the note. If the system could not identify any parameter (i.e. Blind Recognition values, Serial Prefix or Artistic Impression) it would be denoted by a hyphen ‘-’ in the relevant cell of the data-grid.
- Time it took to recognize given set of notes would be displayed using the timer in the bottom of the form window.
- Double click on relevant record on the data-grid or use track-bar to navigate through the records of each currency note.

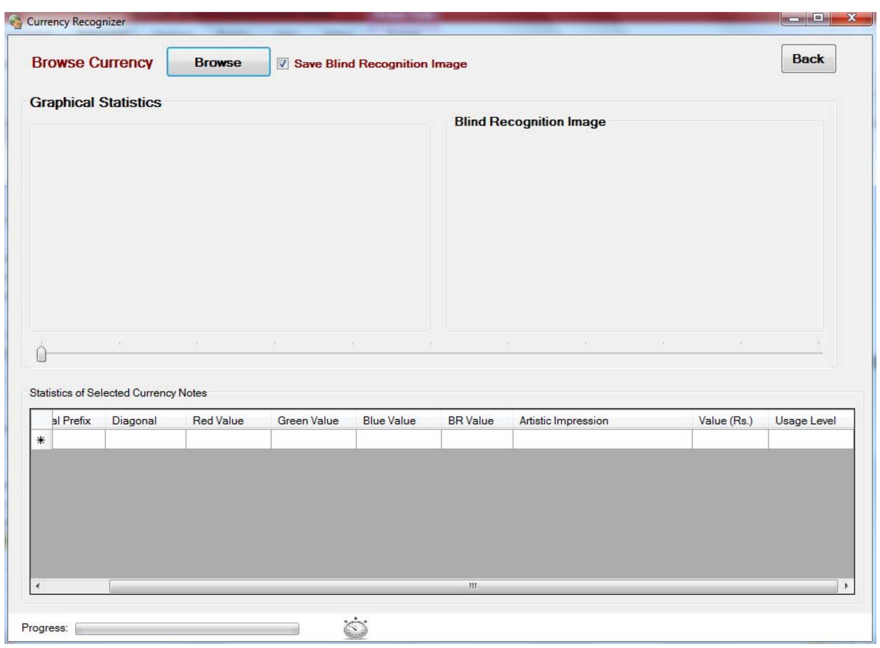

Figure 6.6 - Currency Note Recognizer interface

6.4 Implementation of the Code

6.4.1 Class Definition

Enhanced Bank Note Recognize system contains 8 major classes which performs unique operations in the system.

Class	Description
cDatabase	cDatabase is the class which directly interacts with the backend database. It manipulates select, insert, delete and update operations against the database.
cTrain4Size	cTrain4Size class is used to train a given denomination of currency notes for their Diagonal values and it goes through a complex and an intelligent algorithm.
cTrain4Color	cTrain4Color class is used to train a given denomination of currency notes for their RGB values and it goes through a complex and an intelligent algorithm.
cColorImageStatistics	cColorImageStatistics class is used to retrieve the mean RGB value of a given currency note. This class uses the libraries of AForge.net (Refer Appendix A – AForge) for this purpose.
cFilters	cFilters class contains the implementation of image filtering methods [17], [18] such as Invert, EdgeDetectDifference, EdgeEnhance and Smooth. (Refer Section 4.3.2.3)
cObjectsDetector	cObjectsDetector class contains the implementation of objects or shape checking algorithms which helps to identify Blind Recognition Value of the note. This class uses the

	libraries of AForge.net (Refer Appendix A – AForge) for this purpose.
cGetBlindRecognitionValue	This is a co-coordinating class which applies necessary filters on a given currency image and invoke cObjectsDetector class in order to identify BlindRecognitionValue of the note.
cPerformOCR	cPerformOCR class contains the necessary implementation to read text on a given currency image. This class uses libraries of MODI object model for this purpose

Table 6.1 - Class Definition

6.4.2 Class Implementation

Moving into the implementation code, it has several classes and this section will explain implementation details of major classes and methods.

cTrain4Size and cTrain4Color Class

cTrain4Size and cTrain4Color classes have almost similar implementation structure. Following algorithm describes the way it works.

Step 1 - Get Currency Key (Value and Serial Number pair)

Step 2 - Get Diagonal/RGB value of selected currency image

Step 3- **if** previously trained currency data of selected of currency type exists = true then

3.1 MinD = Minimum of Diagonal of specified Currency type

3.2 MaxD = Maximum of Diagonal of specified Currency type

3.3 **if** MinD < Diagonal of selected currency < MaxD

3.3.1 **Mark as valid data**

3.4 **else**

3.4.1 count = 0

3.4.2 while count < Number of trained notes for other denominations

3.4.2.1 MinOD = Minimum of Diagonal of other Currency type

3.4.2.2 MaxOD = Maximum of Diagonal of other Currency type

3.4.2.3 **if** MinOD < Diagonal of selected currency < MaxOD

3.4.2.3.1 **Display an Information message**

3.4.2.4 **else**

3.4.2.4.1 **Mark as valid data**

3.4.2.5 count = count + 1

Step 4 - **else**

4.1 Go back to 3.4.1

Above algorithms ensure that trained data does not contain any outliers or invalid data. However, user has the full control to add the outliers to the system or not.

cGetBlindRecognitionValue Class

This is a co-coordinating class which applies Invert, Smooth, EdgeDetectDifference and EdgeEnhance filters on a given currency image and invoke cObjectsDetector class in order to identify BlindRecognitionValue of the note (Figure 6.7).

```
class cGetBlindRecognitionValue
{
    public string getBlindRecognitionValue(Bitmap bmpImage, int Note)
    {
        cFilters.Invert(bmpImage);
        bmpImage.Save("Temp Images\\Filters\\" + Note + "-1.bmp");
        cFilters.Smooth(bmpImage, 1);
        bmpImage.Save("Temp Images\\Filters\\" + Note + "-2.bmp");
        cFilters.EdgeDetectDifference(bmpImage, 55);
        bmpImage.Save("Temp Images\\Filters\\" + Note + "-3.bmp");
        cFilters.EdgeEnhance(bmpImage, 2);
        bmpImage.Save("Temp Images\\Filters\\" + Note + "-4.bmp");

        string sResult = cObjectsDetector.getCircleCount(bmpImage);

        if (frmRecognizeCurrency.chkSaveSteps == true)
        {
            bmpImage.Save("Temp Images\\BR Value\\" + Note + ".bmp");
        }
        return sResult;
    }
}
```

Figure 6.7- Implementation of cGetBlindRecognitionValue Class

Morphology filters and numeric values (i.e. 1, 55, 2) that have been passed as parameters to the static methods of the cFilter class were decided after a series of researches, by taking a sample of more than 100 currency images, belongs to different denomination of 12th series of Sri Lankan currency notes. These parameters can be considered as the most suitable set of values to apply the morphology filter on the image before performs the Edge Detection, since they gave the best results.

However, selection of these filters and suitable parameters can be further automated and can be suggested as a future improvement to the system.

After applying Morphology Filters on currency image it will be sent to identify the number or count of the blind recognition value. Relevant count would be returned by static method getCircleCount along with the coordinates of X axis of the blind circle(s) as a concatenated string value. Value of the X axis of the blind circle(s) would be used as a feed forward data for OCR process.

cFilters Class

cFilters class contains the implementation of image filtering methods Invert, Smooth, EdgeDetectDifference and EdgeEnhance [25].

- **Invert(Bitmap Image) method**

Invert method makes the colors on the selected pixel of the given currency image become it's opposites on the color scale. For example, black will become white, and vice versa. Dark gray will become Light Gray, and vice versa. However, when there is a shape which is a shade of gray exactly 50% black and 50% white, in which case there will be no color change.

- **Smooth(Bitmap Image, int Weight) method**

Smooth method helps to reduce details of the currency image or in other words image would be filtered for noise. This method will smooth the corners of the shapes without losing edges and this contemporary effect is great for hiding flaw in extracted edges on an image. In this case, it will sharpen the edges of the blind circles of the currency image.

- **EdgeDetectDifference(Bitmap Image, byte Threshold) method**

EdgeDetectDifference method employed a more refined second-order edge detection approach which helps to detects edges with sub pixel accuracy which uses differential approach of detecting zero-crossings of the second-order directional derivative. This method executes by working out the greatest difference between a pixel and its eight neighbors. The threshold allows softer edges to be forced down to black and use zero to negate its effect.

- **EdgeEnhance (Bitmap Image, byte Threshold) method**

EdgeEnhance method is operates after the image has been filtered for noise from Smooth method and after applying second-order edge detection from EdgeDetectDifference method. The threshold allows softer edges to be forced down to black and use 0 to negate its effect. The lower the threshold, the more edges will be detected, and the result will be increasingly susceptible to noise and detecting edges of irrelevant features in the image. Conversely a high threshold may miss subtle edges, or result in fragmented edges. If threshold value selected carefully, this would results in one pixel thick edge elements.

After applying the above four filters to currency image, it can be noticed that all the edges and corners of the shapes on the currency image are highlighted and extracted. This filtered image will be sent to identify objects on it.

Implementation: cObjectsDetector Class

cObjectsDetector class used to identify the Blind Recognition circle(s) of the filtered currency image. This class uses AForge.Imaging, AForge.Imaging.Filters, and AForge.Math.Geometry namespaces of AForge library. This method executes according to following two steps [24].

- **Locating objects in currency and check for blind circle and highlight**

Assigned values for 'blobcounter.MinHeight' and 'blobcounter.MinWidth' in pixels are the approximate minimum height and width of a blind circle on currency image. All the other objects less than the given height and width will be discarded. Rest of the objects would be identified using GetObjectsInformation method and located in the Blob array.

Below code segment get the edge point of each identified object from the blob array. Subsequently edge points will be passed to shapeCheker.IsCircle method to detect for circles and it will return the radius and center point coordination. Out line of the detected circles will be colored yellow to identify them easily. Variable iCount, used to count the number of circles detected in the given currency image(Figure 6.8).

```
public static string getCircleCount(Bitmap bitmap)
        {
            int iCount = 0;

            BitmapData bitmapData = bitmap.LockBits(new Rectangle(0, 0, bitmap.Width, bitmap.Height),
ImageLockMode.ReadWrite, bitmap.PixelFormat);

            BlobCounter blobCounter = new BlobCounter();
            blobCounter.FilterBlobs = true;
            blobCounter.MinHeight = 15;
            blobCounter.MinWidth = 15;

            blobCounter.ProcessImage(bitmapData);
            Blob[] blobs = blobCounter.GetObjectsInformation();
            bitmap.UnlockBits(bitmapData);

            SimpleShapeChecker shapeChecker = new SimpleShapeChecker();
            Graphics g = Graphics.FromImage(bitmap);
            Pen yellowPen = new Pen(Color.Yellow, 2);
            int iLocation = 0;

            for (int i = 0, n = blobs.Length; i < n; i++)
            {
                List<IntPoint> edgePoints = blobCounter.GetBlobsEdgePoints(blobs[i]);
                DoublePoint center;
                double radius;

                if (shapeChecker.IsCircle(edgePoints, out center, out radius))
                {
                    g.DrawEllipse(yellowPen, (float)(center.X - radius), (float)(center.Y - radius),
                                            (float)(radius * 2), (float)(radius * 2));
                    iCount++;
                    iLocation = Convert.ToInt32(center.X);
                }
            }
            yellowPen.Dispose();
            g.Dispose();

            return iCount + ":" + iLocation;
        }
```

Figure 6.8 - Implementation of Locating Objects and Checking for Blind Circle

cPerformOCR Class

cPerformOCR class contains the necessary methods to read Serial Prefix and Artistic Impression text on a given currency image. This class uses libraries of MODI object model for this purpose [22]. It should be noted that the given currency image should be in correct direction (text should not be in upside down), in order to correctly read the Serial Prefix and Artistic Impression. Therefore, a method should be employed to determine whether the currency image is in the correct direction. For that purpose, value of the X axis of the blind circle(s) would be used as feed forward information for OCR process, where it can give a clue about the direction of the note (Figure 6.9).

```
cPerformOCR PerformOCR = new cPerformOCR();

                    int BRXValue = Convert.ToInt32(arrBRnX[1]);
                    int BlindCount = Convert.ToInt32(arrBRnX[0]);
                    if (((BRXValue < 50) || (BRXValue > 150)) & (BlindCount!=0))
                    {
                        RotateCurrencyNote(new     Bitmap(sArrImageSource[note]),
180).Save("Temp Images\\Rotate\\" + note + ".jpg");
                        dgvRecognizeNoteStatistics["dgcRNSSerialPrefix",
note].Value = PerformOCR.getSerialPrefix(new Bitmap("Temp Images\\Rotate\\" + note
+ ".jpg"), note);
                        dgvRecognizeNoteStatistics["dgcRNSArtisticImpression",
note].Value = PerformOCR.getArtisticImpression(new Bitmap("Temp Images\\Rotate\\"
+ note + ".jpg"), note);
                    }
                    else
                    {
                        dgvRecognizeNoteStatistics["dgcRNSSerialPrefix",
note].Value   =   PerformOCR.getSerialPrefix(new   Bitmap(sArrImageSource[note]),
note);
                        dgvRecognizeNoteStatistics["dgcRNSArtisticImpression",
note].Value = PerformOCR.getArtisticImpression(new Bitmap(sArrImageSource[note]),
note);
                    }
```

Figure 6.9 - Direction of the note from BR value recognition

If the note is in the wrong direction where text on the image appears upside down, a rotation (Figure 6.10) should be done prior to send the image for OCR process.

```
private static Bitmap RotateCurrencyNote(System.Drawing.Image image, float angle)
{
    Bitmap rotatedBmp = new Bitmap(image.Width, image.Height);
    rotatedBmp.SetResolution(image.HorizontalResolution, image.VerticalResolution);

    Graphics g = Graphics.FromImage(rotatedBmp);
    g.TranslateTransform((float)image.Width, (float)image.Height);
    g.RotateTransform(angle);
    g.DrawImage(image, new PointF(0, 0));

    return rotatedBmp;
}
```

Figure 6.10 - Implementation of Rotating an Image

After rotating (if necessary), the image would be sent for OCR process. OCR process operates in following three steps.

- **Crop the currency image**

Before perform actual OCR operation, currency image would be cropped to limit the area which require performing OCR [23]. This helps to avoid generating unnecessary text from the OCR process and also it would be more efficient and will give more accurate results. Best parameters for Crop Area were decided after series of attempts.

However, selecting the necessary parameters for cropping can be further automated and can be suggested as a future enhancement to the system (Figure 6.11).

```
switch (i)
{
    case 0:
        CropArea = new Rectangle(105, 165, 200, 95);
        break;
    case 1:
        CropArea = new Rectangle(105, 165, 250, 95);
        break;
    case 2:
        CropArea = new Rectangle(105, 165, 210, 95);
        break;

    case 3:
        CropArea = new Rectangle(1250, 175, 75, 200)
        break;
}
```

Figure 6.11 - Implementation of Selecting Crop Area

- **Perform OCR on cropped currency image**

A colne of the cropped currency image would be sent to MODI OCR engine to identify English text on it [22] (Figure 6.12).

```
Bitmap objBitmapClone = bmpImage.Clone(CropArea, bmpImage.PixelFormat);

int iNote = Note + 1;
objBitmapClone.Save("Temp Images\\Serial Prefix\\" + iNote + ".jpg");

DocumentClass doc = new DocumentClass();
doc.Create("Temp Images\\Serial Prefix\\" + iNote + ".jpg");
doc.OCR(MiLANGUAGES.miLANG_ENGLISH, true, true);
```

Figure 6.12 - Implementation of Sending Data for OCR

- **Compare the read text with Serial Prefix/Artistic Impression text**

Compare the text recognized from MODI OCR engine with relevant text of Serial Prefix/Artistic Impression text (Figure 6.13).

```
foreach (MODI.Image image in doc.Images)
{
    string sResult = image.Layout.Text;
    if (sResult.StartsWith("W", true, null))
    {
        return "W";
    }
    else if (sResult.StartsWith("V", true, null))
    {
        return "V";
```

Figure 6.13 - Implementation of Comparing OCR Result with Relevant Text

6.5 Implementation of the Data analysis

RGB value analysis has been conducted to automate manual process of identifying unusable denomination. With the help of few users, get the results of the scanned images. Identify a RGB range where the denomination is highly used. Using that range system can identify all the scanned images are (same denomination value) able to use more or in need of replace (Figure 6.14).

#	Note name	Human eye recognition 1_anushi	Human eye recognition 2_Manjula	Human eye recognition 3_Devinda	20-System(RGB Range)			
					R	G	B	
1	20-W2	1	0	0	173.815	131.245	141.197	R > 160-175
2	20-W4	1	1	1	164.372	114.576	127.008	G > 110-125
3	20-W5	1	1	1	166.201	119.217	127.73	B > 120-135
4	20-W9	1	1	1	167.812	124.024	127.311	
5	20-W15	1	1	1	176.355	136.704	146.293	if in the range replace '1'
6	20-W19	1	0	0	177.179	136.895	145.979	
7	20-W20	1	1	1	166.981	117.508	125.207	1 - Replace
8	20-W22	1	0	0	180.811	140.749	150.415	0 - Do not repalce
9	20-W23	1	0	0	177.514	138.769	150.005	
10	20-W234	1	1	0	170.048	123.875	134.39	
11	20-W30	0	0	0	172.99	126.445	140.149	
12	20-W33	0	0	0	184.264	145.393	158.955	
13	20-W37	0	0	0	177.65	134.527	150.232	
14	20-W40	0	0	0	186.906	146.866	157.59	
15	20-W41	0	0	0	174.521	130.419	144.763	
16	20-W44	1	0	0	187.084	140.804	157.231	
17	20-W49	1	0	0	187.793	152.162	167.778	
18	20-W54	0	0	0	188.615	144.492	162.317	
19	20-W55	0	0	0	189.274	153.114	169.192	
20	20-W59	0	0	0	177.583	134.383	150.592	

Figure 6.14 - RGB Value Analysis

Upon identifying whether to Replace the currency note or Not, it can be viewed from the System. This helps user to identify unusable denomination via automated system, which saves time and money.

```
if (dgvRecognizeNoteStatistics["dgcRNSValue", iNote].Value == "Rs.100")
{
    double x = (double)dgvRecognizeNoteStatistics["dgcRNSRedMean", iNote].Value;
    double y = (double)dgvRecognizeNoteStatistics["dgcRNSGreenMean", iNote].Value;
    double z = (double)dgvRecognizeNoteStatistics["dgcRNSBlueMean", iNote].Value;

    int Replace = 0;

    if (x > 193 && x < 208)
    {
        Replace++;
    }
    if (y > 126 && y < 142)
    {
        Replace++;
    }
    if (z > 109 && z < 126)
    {
        Replace++;
    }

    if (Replace >=2)
    {
        dgvRecognizeNoteStatistics["dgcRNSUsageLevel", iNote].Value = "Replace";
    }
    else
    {
        dgvRecognizeNoteStatistics["dgcRNSUsageLevel", iNote].Value = "Do not Replace";
    }
}
```

Figure 6.15 - RGB Value Analysis for Rs.100

6.6 Summary

The chapter describes the implementation of the proposed solution and next chapter will describe the Evaluation of the developed system. Test cases, test results can be added in next chapter.

Chapter 7

Evaluation

7.1 Introduction

This chapter discusses the evaluation of the designed and developed system. An evaluation was done for the system by measuring the accuracy and reliability of currency note recognition and performance (speed) of the system. Similarly accuracy of the RGB value analysis also considered in this section.

A sample notes of each denomination which includes distorted currency notes as well as clean currency notes have been used for this purpose.

7.2 Accuracy and Reliability

As discussed in above chapters, system extracts features based on seven parameters (i.e. mean Red, Green, Blue values, diagonal of the note, Blind Recognition Value, Serial Prefix and Artistic Impression.) from a given currency note and those extracted features are used in an intelligent system to recognize currency note. Accuracy and reliability of the system has been measured by individually considering the results obtained for Serial Prefix, Blind Recognition Value, and Artistic Impression.

Rs.	Serial Prefix		BR Value		Art. Impression		Overall	
	Torn Ratio	Good Ratio	Torn Ratio	Good Ratio	Torn Ratio	Good Ratio	Torn Ratio %	Good Ratio %
20	4/5	5/5	4/5	5/5	3/5	4/5	**73**	**93**
50	3/5	5/5	4/5	5/5	4/5	3/5	**73**	**86**
100	4/5	5/5	4/5	5/5	4/5	5/5	**80**	**100**
500	3/5	5/5	4/5	5/5	3/5	5/5	**67**	**100**
1000	3/5	5/5	4/5	5/5	3/5	4/5	**67**	**93**
5000	3/5	4/5	4/5	5/5	3/5	4/5	**67**	**86**

Table 7.1 - Statistics for Accuracy and Reliability

- **Serial Prefix**

According to the results in the table (Refer Table 7.1), system has correctly read the serial prefix of 9 out of 10 notes of each denomination (maximum value), where the maximum success rate has value of 90%.

- **Blind Recognition Value**

According to the results in the table (Refer Table 7.1), system has correctly identified the blind recognition value of 9 out of 10 notes of each denomination, where the maximum success rate has a value of 90%.

- **Artistic Impression**

According to the results in the table (Refer Table 7.1), system has correctly read the serial prefix of 7 out of 10 notes of each denomination, where the minimum success rate has a value of 70%.

However, success ratio and the rate are very low, when reading the Artistic impression of 20 Rupee note. Main reason for this is that 20 Rupee notes are circulated much more than the other denomination of notes and most of them are highly distorted.

- **Overall Accuracy of the System**

According to the results in the table (Refer Table 7.1), minimum success rate has a ratio of 67% where torn and worn notes are considered. Other than that the system gives average success rate as 80% and above.

- **Accuracy of the Data Analysis**

Accuracy and reliability of the Data Analysis has been measured by individually, considering each denomination. Table 7.2 depicts the test results of the Data Analysis which performed using RGB values. Expected results show the count of 'Replace' notes and 'Do Not Replace' notes. This is a manual task done by the user. Actual results received from system are captured in next column. Difference of the expected and actual results are calculated and displayed as a percentage in the last columns.

Eg: Rs. 100 bank note, hundred notes have been used for this analysis. Human eye has recognized 13 notes to replace and 87 not to replace. Out of 13 replace notes, system has recognized 11 notes to replace. Similarly, out of 87 not to replace notes, system has recognized 80 only. Difference of the 'Replace' ratio calculated as 84% ((11/13)*100) and 'Not to Replace' ratio as 92%.

Value of the Note (Rs.)	No of Notes Used	Expected results		Actual Results from system		Difference	
		Replace	Do Not Replace	Replace	Do Not Replace	Replace %	Do Not Replace %
20	50	15	35	13	35	86	100
50	30	9	21	9	15	100	71
100	100	13	87	11	80	84	92
500	40	15	25	12	15	80	60
1000	64	24	40	14	31	58	77.5

Table 7.2 - Data Analysis Accuracy Rate

Denomination 5000 has not considered in this data analysis, as most of the 5000 rupee has not circulated in public compared to other bank notes. Hence, there is no much difference whether to Replace or Not to Replace 5000 Bank note.

According to the results in the table (Refer Table 7.2), overall accuracy rate of the data analysis can be identified as 70% and above (maximum rate).

7.3 Performance

Performance of the system is important, when it is going to be used in real world. Following table represents the average time it took to recognize 25 notes in each denomination and average time to recognize a single note and the note recognition rate as notes per minute (Table 7.3). Also it demonstrates the overall performance of the system by taking 25 notes from different denominations as test data.

Rs.	No. of Notes	Average Time	Avg. Time to recognize 1 note	Rate (Notes/min)
20	25	2min 01s	4840ms	12
50	25	1min 09s	2760ms	22
100	25	1min 23s	3320ms	18
500	25	1min 25s	3400ms	18
1000	25	1min 14s	2960ms	20
5000	10	38 secs	3800ms	16

Table 7.3 - Statistics for Performance

According to obtained results, system correctly recognizes 25 currency notes(Rs.50 value) in 1minute and 09seconds. Therefore, it takes an average time of 2760miliseconds to identify a given note. However, when the number of distorted or old notes is high, it takes more time to recognize.

7.4 Summary

The chapter describes the evaluation of the developed solution and next chapter will describe the conclusion of the overall system. Future work also can be added in next chapter.

Chapter 8

Conclusion

8.1 Introduction

The Main objective of the research was to develop an efficient and accurate system to identify torn and worn out currency notes which identifies twelfth series of Sri Lankan currency notes, issued by Central Bank of Sri Lanka. Recognizing both clean currency notes and old, noisy currency notes from either direction were captured the scope of the project. Automating the manual process of identifying unusable denomination via system is also another sub objective of this developed system.

8.2 Future Work

There are many future works that can be done for this research to improve.

- **Identify the note by both sides**

In current system, there is no any mechanism to recognize a given note in both sides and it identifies the note by extracting the features in front of the note. However, this limitation can be overcome by including minor adjustments to currency note scanning machine where it scans both sides of the note, hence system can always recognizes the note using the features in the front side, since it has images of both sides. As another option, certain feature such as RGB values, Diagonal value, numeric and texts which represent the values of notes that are in the backside can be used to differentiate notes.

- **Recognize currencies of other series**

Objective of this research was to develop a system which can recognize twelfth series of Sri Lanka currency notes. However, the algorithm can be modified and enhanced to recognize a denomination of currency notes belongs to series of previous releases.

- **Improve the performance**

Performance or the currency recognition speed can be improved to a very god level by using Threads in the code as well as using fast and accurate OCR engine.

- **Identify Counterfeit Currency notes**

Even though the system objective is not to identify Counterfeit Currency notes, it can be achieved by a counting plug-in or using black-light. Counting machines can identify forged notes by checking notes' standard density [21]. Since, Sri Lankan currency note series issued to date have fluorescent symbols on them, black-light (UV) based detector can also be used to identify forge notes.

- **Improve the system to identify the hand written bank notes**

With the latest government rules noticed 2017 December, to have a hand written bank notes with a person or any other business is a useless action. As it has no value and any ban will not take such money to the bank. Hence, banking systems like ATM machines should be able to identify hand written bank notes effectively.

8.3 Summary

As shown in Chapters 7 - Evaluation, system successfully recognizes 80% (above) of notes including torn or worn bank notes. It also proved that accuracy and reliability of the system is in a very high level and also the performance (Notes per Second) is in a satisfactory level.

Therefore considering these facts, it can be concluded that the achievements of this research in this area are well enough and findings of this research can be used as a base for other researches.

References

[1] Central Bank of Sri Lanka (2010). Currency Management, New Currency Note Series of Central Bank of Sri Lanka. [Online]. Available: http://www.cbsl.gov.lk/htm/english/06_cm/c_4.html

[2] 2010 - Sri Lanka - Currency note Development, Prosperity and Sri Lanka Dancers. [Online]. Available: http://notes.lakdiva.org/2010/

[3] Wikipedia, the free encyclopedia (2011, August.). Banknote counter. [Online]. Available: http://en.wikipedia.org/wiki/Banknote_counter

[4] Global Market, Certified Manufactures Online. Currency Counter. [Online]. Available: http://www.globalmarket.com/products/machinery-equipments/business-equipments/currency-counter-4007.html?gclid=COWm05jrs7ACFU966wods0jsWQ

[5] D.A.K.S. Gunaratna, N.D.Kodikara and H.L.Premaratne, "ANN Based Currency Recognition System using Compressed Gray Scale and Application for Sri Lankan Currency Notes-SLCRec", Proceedings of world academy of science, engineering and technology, vol. 35, ISSN 2070-3740, pp. 235-240, Nov 2008.

[6] Central Bank of Sri Lanka (2010). Currency Management, New Currency Note Series of Central Bank of Sri Lanka. [Online]. Avalable: http://www.cbsl.gov.lk/htm/english/06_cm/security_featurs.html

[7] Chandra Sekhar Panda, Srikanta Patnaik, "Filtering Corrupted Image and Edge Detection in Restored Grayscale Image Using Derivative Filters", International Journal of Image Processing, (IJIP) Volume (3) : Issue (3)

[8] M. Gori, A. Frosini and P. Priami. "A neural network based model for paper currency recognition and verification", IEEE Trans. Neural Networks, Nov.1996, pp. 1482-1490.

[9] H. Hassanpour and E. Hallajian, "Using Hidden Markov Models for Feature Extraction in Paper Currency Recognition", Expert Systems with Applications, Vol. 36, No. 6, pp. 10105-10111, 2009.

[10] Ms. Trupti Pathrabe, Mrs.Swapnili Karmore, A Novel Approach of Embedded System for Indian Paper Currency Recognition. International Journal of Computer Trends and Technology. May to June Issue 2011, ISSN: 2231-2803.

[11] M. Tanaka, F. Takeda, K. Ohkouchi, Y. Michiyuk. Recognition of Paper Currencies by Hybrid Neural Network, IEEE Transactions on Neural Networks. 0-7803-4859-1/98, 1998.

[12] The Code Project. How To: Use Office 2007 OCR Using C#. (2009, August.) [Online]. Available: http://www.codeproject.com/Articles/41709/How-To-Use-Office-2007-OCR-Using-C

[13] stackoverflow [Online]. NET OCRing an Image. [Online]. Available: http://stackoverflow.com/questions/1130473/net-ocring-an-image

[14] Bitmap.GetPixel Method. (2012). [Online]. Available: http://msdn.microsoft.com/en-us/library/wbey6xyz(vs.71).aspx

[15] Muhammad Sarfraz , "An intelligent paper currency recognition system", Department of Information Science, College of Computing Sciences & Engineering, Kuwait University, Safat 13060,Kuwait, 2015.09.12

[16] Ji Woo Lee, Hyung Gil Hong, Ki Wan Kim, Kang Ryoung Park, A Survey on Banknote Recognition Methods by Various Sensors, February 2017.

[17] Wickramasinghe, K. and De Silva, D., (2013). Bank Notes Recognition Device for Sri Lankan Vision Impaired Community. In Proc. 8th International Conference on Computer Science & Education (ICCSE 2013): 609-612.

[18] U.S.S. Perera, D.N. Balasuriya, Sri Lankan Currency note recognizer for visually impaired people, Department of Electrical & Computer Engineering, The Open University of Sri Lanka.

[19] Image Processing for Dummies with C# and GDI+ Part 3 - Edge Detection Filters. (2002) [Online]. Available: http://www.codeproject.com/Articles/2056/Image-Processing-for-Dummies-with-C-and-GDI-Part-3

[20] Chandra Sekhar Panda, Srikanta Patnaik, "Filtering Corrupted Image and Edge Detection in Restored Grayscale Image Using Derivative Filters", International Journal of Image Processing, (IJIP) Volume (3) : Issue (3)

[21] Declan McAleese. "Counterfeit Currency Detection Techniques", December 2012.

[22] How To: Use Office 2007 OCR Using C# - Code Project (2009) [Online]. Available: http://www.codeproject.com/Articles/41709/How-To-Use-Office-2007-OCR-Using-C

[23] Cropping Images - Code Project (2008) [Online]. Available: http://www.codeproject.com/Articles/30725/Cropping-Images

[24] AForge.NET :: FrameWork Samples - Image Processing (2012) [Online]. Available: http://www.aforgenet.com/framework/samples/image_processing.html

[25] Image Processing for Dummies with C# and GDI+ Part 2 - Convolution Filters (2005) [Online]. Available: http://www.codeproject.com/Articles/2008/Image-Processing-for-Dummies-with-C-and-GDI-Part-2

Appendix A

Source Code

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

using System.Drawing.Imaging;
using System.IO;
using MODI;
using System.Diagnostics;
using System.Data.SqlClient;

//using IronOcr;
//using System.Data.SqlClient;
//using AForge.Imaging.Filters;

namespace EBRS
{
    public partial class frmRecognizeCurrency : Form
    {
        public frmRecognizeCurrency()
        {
            InitializeComponent();
        }

        public static bool chkSaveSteps;
        private string[] sArrImageSource = null;
        Bitmap objBitmap = null;

        private void btnBrowseNote_Click(object sender, EventArgs e)
        {

            cMyClass mc = new cMyClass();
            sArrImageSource = mc.BrowseImage("Browse Image");

            DateTime startDateTime = DateTime.Now;
```

```
            if (sArrImageSource != null)
            {
                chkSaveSteps = chkSaveIntermidiateOperaions.Checked;
                dgvRecognizeNoteStatistics.Rows.Clear();
                imgNote.Load(sArrImageSource[0]);
                trkNotes.Maximum = sArrImageSource.Length - 1;

                ProgressBar.Maximum = sArrImageSource.Length;

                for (int note = 0; note < sArrImageSource.Length; note++)
                {
                    dgvRecognizeNoteStatistics.Rows.Add();
                    dgvRecognizeNoteStatistics["dgcRNSNo", note].Value = note + 1;


                    //Diagonal
                    objBitmap = new Bitmap(sArrImageSource[note]);
                    dgvRecognizeNoteStatistics["dgcRNSDiagonal", note].Value =
Math.Round(Math.Sqrt((Math.Pow(objBitmap.Width, 2)) + (Math.Pow(objBitmap.Height, 2))),
3);


                    //RGB
                    if (objBitmap.PixelFormat == PixelFormat.Format24bppRgb)
                    {
                        cColorImageStatistics cis = new cColorImageStatistics(objBitmap);
                        dgvRecognizeNoteStatistics["dgcRNSRedMean", note].Value =
cis.RedMeanValue;
                        dgvRecognizeNoteStatistics["dgcRNSGreenMean", note].Value =
cis.GreeenMeanValue;
                        dgvRecognizeNoteStatistics["dgcRNSBlueMean", note].Value =
cis.BlueMeanValue;


                    }


                    //BR
                    cGetBlindRecognitionValue getBRV = new cGetBlindRecognitionValue();
                    string[] arrBRnX = getBRV.getBlindRecognitionValue(objBitmap,
note).Split(':');
                    dgvRecognizeNoteStatistics["dgcRNSBRValue", note].Value =
arrBRnX[0].ToString();
                    if (chkSaveSteps == true)
                    {
                        imgBRNote.Load("Temp Images\\BR Value\\0.bmp");
                    }


                    //SerialPrefix and ArtisticImpression using OCR

                    cPerformOCR PerformOCR = new cPerformOCR();

                    int BRXValue = Convert.ToInt32(arrBRnX[1]);
                    int BlindCount = Convert.ToInt32(arrBRnX[0]);
                    if (((BRXValue < 50) || (BRXValue > 150)) & (BlindCount!=0))
                    {
```

```
                        RotateCurrencyNote(new Bitmap(sArrImageSource[note]),
180).Save("Temp Images\\Rotate\\" + note + ".jpg");
                        dgvRecognizeNoteStatistics["dgcRNSSerialPrefix", note].Value =
PerformOCR.getSerialPrefix(new Bitmap("Temp Images\\Rotate\\" + note + ".jpg"), note);
                        dgvRecognizeNoteStatistics["dgcRNSArtisticImpression",
note].Value = PerformOCR.getArtisticImpression(new Bitmap("Temp Images\\Rotate\\" + note
+ ".jpg"), note);
                    }
                    else
                    {
                        dgvRecognizeNoteStatistics["dgcRNSSerialPrefix", note].Value =
PerformOCR.getSerialPrefix(new Bitmap(sArrImageSource[note]), note);
                        dgvRecognizeNoteStatistics["dgcRNSArtisticImpression",
note].Value = PerformOCR.getArtisticImpression(new Bitmap(sArrImageSource[note]), note);
                    }

                    //Usage Level
                    dgvRecognizeNoteStatistics["dgcRNSUsageLevel", note].Value =
"Replace";

                    recognizeCurrencyValue(note);
                    ProgressBar.Value = note;
                }

                ProgressBar.Value = sArrImageSource.Length;
                DateTime endDateTime = DateTime.Now;
                TimeSpan duration = endDateTime - startDateTime;

                // Show duration
                lblTime.Text = (string.Format("{0:00} : {1:00} : {2:00} : {3:000}",
duration.Hours, duration.Minutes, duration.Seconds, duration.Milliseconds));

            }
        }

        private void recognizeCurrencyValue(int Note)
        {
            int iCurrentRow = 0;
            int iNote = Note;
            while (dgvRecognizeNoteStatistics.RowCount-1 > iCurrentRow)
            {
                int[] arrPoints = new int[6];

                string sSerialPrefix = (dgvRecognizeNoteStatistics["dgcRNSSerialPrefix",
iCurrentRow].Value).ToString();
                if (sSerialPrefix == "W")
                {
                    arrPoints[0] = 1;
                }
                else if (sSerialPrefix == "V")
                {
                    arrPoints[1] = 1;
                }
                else if (sSerialPrefix == "U")
                {
                    arrPoints[2] = 1;
                }
                else if (sSerialPrefix == "T")
```

```
                {
                    arrPoints[3] = 1;
                }
                else if (sSerialPrefix == "S")
                {
                    arrPoints[4] = 1;
                }
                else if (sSerialPrefix == "R")
                {
                    arrPoints[5] = 1;
                }


                int iBlindRecognitionValue =
Convert.ToInt32(dgvRecognizeNoteStatistics["dgcRNSBRValue", iCurrentRow].Value);
                if (iBlindRecognitionValue == 1)
                {
                    arrPoints[0] = arrPoints[0] + 2;
                }
                else if (iBlindRecognitionValue == 2)
                {
                    arrPoints[1] = arrPoints[1] + 2;
                }
                else if (iBlindRecognitionValue == 3)
                {
                    arrPoints[2] = arrPoints[2] + 2;
                }
                else if (iBlindRecognitionValue == 4)
                {
                    arrPoints[3] = arrPoints[3] + 2;
                }
                else if (iBlindRecognitionValue == 5)
                {
                    arrPoints[4] = arrPoints[4] + 2;
                }
                else if (iBlindRecognitionValue == 6)
                {
                    arrPoints[5] = arrPoints[5] + 2;
                }


                string sArtisticImpression =
(dgvRecognizeNoteStatistics["dgcRNSArtisticImpression", iCurrentRow].Value).ToString();
                if (sArtisticImpression == "Colombo Port")
                {
                    arrPoints[0] = arrPoints[0] + 1;
                }
                else if (sArtisticImpression == "Manampitiya Bridge")
                {
                    arrPoints[1] = arrPoints[1] + 1;
                }
                else if (sArtisticImpression == "Norochcholai Coal Power Plant")
                {
                    arrPoints[2] = arrPoints[2] + 1;
                }
                else if (sArtisticImpression == "World Trade Center and Bank of Cylon
Headquarters")
                {
```

```
                    arrPoints[3] = arrPoints[3] + 1;
                }
                else if (sArtisticImpression == "Ramboda Tunnel")
                {
                    arrPoints[4] = arrPoints[4] + 1;
                }
                else if (sArtisticImpression == "Weheragala Dam")
                {
                    arrPoints[5] = arrPoints[5] + 1;
                }

                double dDiagonal = 
Convert.ToDouble(dgvRecognizeNoteStatistics["dgcRNSDiagonal", iCurrentRow].Value);

                cDatabase db = new cDatabase();
                SqlDataReader dr1 = db.Select("SELECT CurrencyKey FROM CurrencySizeTab 
WHERE (ABS(MeanDiagonal - " + dDiagonal + " )) = (SELECT MIN(ABS(MeanDiagonal - " + 
dDiagonal + " )) FROM CurrencySizeTab)");

                string sCurrencyKey = "";

                if (dr1.HasRows)
                {
                    while (dr1.Read())
                    {
                        sCurrencyKey = dr1.GetString(0);
                    }
                }


                if (sCurrencyKey == "W^20")
                {
                    arrPoints[0] = arrPoints[0] + 1;
                }
                else if (sCurrencyKey == "V^50")
                {
                    arrPoints[1] = arrPoints[1] + 1;
                }
                else if (sCurrencyKey == "U^100")
                {
                    arrPoints[2] = arrPoints[2] + 1;
                }
                else if (sCurrencyKey == "T^500")
                {
                    arrPoints[3] = arrPoints[3] + 1;
                }
                else if (sCurrencyKey == "S^1000")
                {
                    arrPoints[4] = arrPoints[4] + 1;
                }
                else if (sCurrencyKey == "R^5000")
                {
                    arrPoints[5] = arrPoints[5] + 1;
                }


                int icount = 0;
                int iMaxValue = 0;
```

```
int iMaxIndex = 0;
while (arrPoints.Length > icount+1)
{
    iMaxValue = (arrPoints[iMaxIndex]);

    if ((iMaxValue) < (arrPoints[icount+1]))
    {
        iMaxValue = arrPoints[icount + 1];
        iMaxIndex = icount + 1;
    }

    icount++;
}

if (iMaxIndex == 0)
{
    dgvRecognizeNoteStatistics["dgcRNSValue", iNote].Value = "Rs.20";
}
else if (iMaxIndex == 1)
{
    dgvRecognizeNoteStatistics["dgcRNSValue", iNote].Value = "Rs.50";
}
else if (iMaxIndex == 2)
{
    dgvRecognizeNoteStatistics["dgcRNSValue", iNote].Value = "Rs.100";
}
else if (iMaxIndex == 3)
{
    dgvRecognizeNoteStatistics["dgcRNSValue", iNote].Value = "Rs.500";
}
else if (iMaxIndex == 4)
{
    dgvRecognizeNoteStatistics["dgcRNSValue", iNote].Value = "Rs.1000";
}
else if (iMaxIndex == 5)
{
    dgvRecognizeNoteStatistics["dgcRNSValue", iNote].Value = "Rs.5000";
}

if (dgvRecognizeNoteStatistics["dgcRNSValue", iNote].Value == "Rs.20")
{
    double x = (double)dgvRecognizeNoteStatistics["dgcRNSRedMean",
iNote].Value;
    double y = (double)dgvRecognizeNoteStatistics["dgcRNSGreenMean",
iNote].Value;
    double z = (double)dgvRecognizeNoteStatistics["dgcRNSBlueMean",
iNote].Value;

    int Replace = 0;

    if (x > 160 && x < 175)
    {
        Replace++;
    }
    if (y > 110 && y < 125)
    {
        Replace++;
    }
```

```
                    if (z > 120 && z < 135)
                    {
                        Replace++;
                    }

                    if (Replace >= 2)
                    {
                        dgvRecognizeNoteStatistics["dgcRNSUsageLevel", iNote].Value =
"Replace";
                    }
                    else
                    {
                        dgvRecognizeNoteStatistics["dgcRNSUsageLevel", iNote].Value = "Do
not Replace";
                    }
                }

                if (dgvRecognizeNoteStatistics["dgcRNSValue", iNote].Value == "Rs.50")
                {
                    double x = (double)dgvRecognizeNoteStatistics["dgcRNSRedMean",
iNote].Value;
                    double y = (double)dgvRecognizeNoteStatistics["dgcRNSGreenMean",
iNote].Value;
                    double z = (double)dgvRecognizeNoteStatistics["dgcRNSBlueMean",
iNote].Value;

                    int Replace = 0;

                    if (x > 141 && x < 153)
                    {
                        Replace++;
                    }
                    if (y > 141 && y < 157)
                    {
                        Replace++;
                    }
                    if (z > 142 && z < 167)
                    {
                        Replace++;
                    }

                    if (Replace >= 2)
                    {
                        dgvRecognizeNoteStatistics["dgcRNSUsageLevel", iNote].Value =
"Replace";
                    }
                    else
                    {
                        dgvRecognizeNoteStatistics["dgcRNSUsageLevel", iNote].Value = "Do
not Replace";
                    }
                }

                if (dgvRecognizeNoteStatistics["dgcRNSValue", iNote].Value == "Rs.100")
                {
                    double x = (double)dgvRecognizeNoteStatistics["dgcRNSRedMean",
iNote].Value;
```

```
                    double y = (double)dgvRecognizeNoteStatistics["dgcRNSGreenMean",
iNote].Value;
                    double z = (double)dgvRecognizeNoteStatistics["dgcRNSBlueMean",
iNote].Value;

                    int Replace = 0;

                    if (x > 193 && x < 208)
                    {
                        Replace++;
                    }
                    if (y > 126 && y < 142)
                    {
                        Replace++;
                    }
                    if (z > 109 && z < 126)
                    {
                        Replace++;
                    }

                    if (Replace >=2)
                    {
                        dgvRecognizeNoteStatistics["dgcRNSUsageLevel", iNote].Value =
"Replace";
                    }
                    else
                    {
                        dgvRecognizeNoteStatistics["dgcRNSUsageLevel", iNote].Value = "Do
not Replace";
                    }
                }

                if (dgvRecognizeNoteStatistics["dgcRNSValue", iNote].Value == "Rs.500")
                {
                    double x = (double)dgvRecognizeNoteStatistics["dgcRNSRedMean",
iNote].Value;
                    double y = (double)dgvRecognizeNoteStatistics["dgcRNSGreenMean",
iNote].Value;
                    double z = (double)dgvRecognizeNoteStatistics["dgcRNSBlueMean",
iNote].Value;

                    int Replace = 0;

                    if (x > 146 && x < 166)
                    {
                        Replace++;
                    }
                    if (y > 123 && y < 148)
                    {
                        Replace++;
                    }
                    if (z > 132 && z < 158)
                    {
                        Replace++;
                    }

                    if (Replace >= 2)
                    {
```

```
                        dgvRecognizeNoteStatistics["dgcRNSUsageLevel", iNote].Value =
"Replace";
                    }
                    else
                    {
                        dgvRecognizeNoteStatistics["dgcRNSUsageLevel", iNote].Value = "Do
not Replace";
                    }
                }

                if (dgvRecognizeNoteStatistics["dgcRNSValue", iNote].Value == "Rs.1000")
                {
                    double x = (double)dgvRecognizeNoteStatistics["dgcRNSRedMean",
iNote].Value;
                    double y = (double)dgvRecognizeNoteStatistics["dgcRNSGreenMean",
iNote].Value;
                    double z = (double)dgvRecognizeNoteStatistics["dgcRNSBlueMean",
iNote].Value;

                    int Replace = 0;

                    if (x > 160 && x < 166)
                    {
                        Replace++;
                    }
                    if (y > 166 && y < 172)
                    {
                        Replace++;
                    }
                    if (z > 118 && z < 136)
                    {
                        Replace++;
                    }

                    if (Replace >= 2)
                    {
                        dgvRecognizeNoteStatistics["dgcRNSUsageLevel", iNote].Value =
"Replace";
                    }
                    else
                    {
                        dgvRecognizeNoteStatistics["dgcRNSUsageLevel", iNote].Value = "Do
not Replace";
                    }
                }

                iCurrentRow++;
            }
        }

        private void trkNotes_ValueChanged(object sender, EventArgs e)
        {
            if (sArrImageSource != null)
            {
                if (trkNotes.Value != 0)
                {
                    dgvRecognizeNoteStatistics.Rows[trkNotes.Value - 1].Selected = false;
                }
```

```
                dgvRecognizeNoteStatistics.Rows[trkNotes.Value + 1].Selected = false;

                imgNote.Load(sArrImageSource[trkNotes.Value]);
                if (chkSaveSteps == true)
                {
                    imgBRNote.Load("Temp Images\\BR Value\\" + trkNotes.Value + ".bmp");
                }

                dgvRecognizeNoteStatistics.Rows[trkNotes.Value].Selected = true;
            }
        }

        private void dgvRecognizeNoteStatistics_CellDoubleClick(object sender,
DataGridViewCellEventArgs e)
        {

dgvRecognizeNoteStatistics.Rows[dgvRecognizeNoteStatistics.CurrentCell.RowIndex].Selected
= true;
            trkNotes.Value = dgvRecognizeNoteStatistics.CurrentCell.RowIndex;
        }


        private static Bitmap RotateCurrencyNote(System.Drawing.Image image, float angle)
        {
            //create a new empty bitmap to hold rotated image
            Bitmap rotatedBmp = new Bitmap(image.Width, image.Height);
            rotatedBmp.SetResolution(image.HorizontalResolution,
image.VerticalResolution);

            Graphics g = Graphics.FromImage(rotatedBmp);    //make a graphics object from
the empty bitmap
            g.TranslateTransform((float)image.Width, (float)image.Height);  //Put the
rotation point in the center of the image
            g.RotateTransform(angle);   //rotate the image
            g.DrawImage(image, new PointF(0, 0));   //draw passed in image onto graphics
object

            return rotatedBmp;
        }

        private void imgNote_DoubleClick(object sender, EventArgs e)
        {
            if (imgNote.Image != null)
            {
                try
                {
                    string ImagePath = imgNote.ImageLocation;
                    System.Diagnostics.Process.Start(@ImagePath);
                }
                catch (Exception) { }
            }
        }

        private void dgvRecognizeNoteStatistics_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {

        }
```

```
        private void button1_Click(object sender, EventArgs e)
        {

            this.Close();


        }


    }
}
```