

**PIN NUMBER DETECTION FROM MOBILE SCRATCH CARD USING
OCR ON ANDROID PLATFORM AND BUILD AN APPLICATION
FOR BALANCE RECHARGE**

BY

MD NAZRUL HASAN KHAN (ID - 10101003)

FAIZAN SIDDIQUI (ID - 11101073)

AMIT DAS (ID -10101020)

SUPERVISOR

PROF. MOHAMMAD ZAHIDUR RAHMAN

CO-SUPERVISOR

MD. ZAHANGIR ALOM



SCHOOL OF ENGINEERING AND COMPUTER SCIENCE

BRAC UNIVERSITY, BANGLADESH

A Thesis submitted in partial fulfillment of the
Requirements for the degree of Bachelor of Science
In
Computer Science and Engineering
Of
BRAC University

By
Md Nazrul Hasan Khan (ID - 10101003)
Faizan Siddiqui (ID - 11101073)
Amit Das (ID -10101020)

Supervisor
Prof. Mohammad Zahidur Rahman
Co-Supervisor
Md. Zahangir Alom

APRIL, 2014

Declaration

This is to certify that the thesis entitled “Pin number detection from mobile phone scratch card using OCR on android platform and build an application for balance recharge ”, which is submitted by Md Nazrul Hasan Khan (ID - 10101003), Faizan Siddiqui (ID - 11101073) , Amit Das (ID -10101020), in partial fulfillment of the requirement for the award of degree of Bachelor of Science in Computer Science & Engineering to the Department of Computer Science & Engineering, BRAC University, 66 Mohakhali C/A, Dhaka, 1212, comprises only their original work and due acknowledgement has been made in the text to all other material used. The result of the thesis has not been submitted to any other University or Institute for the award of any degree or diploma.

Signature of Authors:

Approved By:

Supervisor: Prof. Mohammad Zahidur Rahman

Co-Supervisor: Md. Zahangir Alom

Amit Das

MD Nazrul Hasan Khan

Faizan Siddiqui

Acknowledgement

We would humbly like to thank everyone who has helped in completion of this thesis work, for their advice, suggestion and help. We cordially thank our supervisor Prof. Mohammad Zahidur Rahman sir, our co-supervisor Zahangir Alom sir, for their endless support. We specially thank to Md. Abdur Rahman for his support. This thesis would not have been possible without their continuous support.

Finally we thank our beloved parents for their never-ending support, motivation and believe in us.

Abstract

This thesis aims to detect mobile recharge card pin number portion by doing image processing. It works with Tesseract Engine and tess-two library for OCR operation. After detection, it extracts the OCRed text which is the pin number of recharge card and then it sends request to the respective mobile operator for recharging balance. We have built an android based cell phone application so that the thesis can be worthy of real life experience.

Table of Contents

DECLARATION :	3
ACKNOWLEDGEMENT :	4
ABSTRACT:	5
TABLE OF CONTENTS:	6-8
LIST OF FIGURES:	9
LIST OF TABLES:	10
CHAPTER- 1 INTRODUCTION:	11
1.1 OPTICAL CHARACTER RECOGNITION	11
1.2 OCR IN MOBILE DEVICE	12
1.3 AUGMENTED REALITY ON OCR	13
1.4 BACKGROUND & MOTIVATION	15
1.5 ORGANIZATION OF THESIS	15
CHAPTER - 2 LITERATURE REVIEW:	17
2.1 Methods of Optical Character Recognition	17
2.2 OCR Algorithm	18
2.2.1 Matrix matching	18
2.2.2 Feature extraction	19
2.3 ORC engines	19
2.3.1 ABBYY Mobile OCR	20
2.3.2 OpenRTK	20

2.3.3 Tesseract.....	21
2.3.4 Online ORC API.....	21
2.3.5 GOCR	21
2.3.6 Orcad.....	21
2.4 OCR SOFTWARE	22
2.4.1 ABBYY Business Card Reader	22
2.4.2 ABBYY Lingvo Dictionaries for Android	23
2.4.3 Text Fairy (OCR)	23
2.4.4 OCR Using Microsoft OneNote 2007.....	24
2.4.5 SimpleOCR	25
2.4.6 TopOCR	25
CHAPTER- 3 PROJECT PROPOSAL	27
3.1 Overview of Our Project	27
3.1.1 Mission	28
3.2 Goal	28
3.3 Scope	28
3.4 Deliverables	29
3.5 Risks and Rewards	29
3.6 Target Audience and Benefits	30
3.7 Project Plan	30
3.7.1 Summary of Project	31

3.7.2 Risk Management	31
3.7.3 Project Planning Dependencies	32
3.8 Requirements and Specification	32-33

CHAPTER – 4 IMPLEMENTATION.....34

4.1 ENVIRONMENT SET UP.....	34-35
4.2 USING OF TESS-TWO LIBRARY	36-38
4.3 USING OF TRAINED DATA	38
4.4 WORK FLOW OF THE SYSTEM	39-42
4.5 INTREGRATION OF USSD ON EXTRACTED PIN NUMBER	43

Chapter- 5 RESULT45

5.1 Accuracy Test for Different Operator(s) in Different Light Conditions (Individual).....	45
5.2 Accuracy Test average for Different Operator(s) in Different Light Conditions.....	48
5.3 Accuracy rate for people of different ages	49
5.4 Accuracy rate In terms of Different Mobile Phones.....	50

Chapter- 6 CONCLUSION AND FUTURE WORK52

6.1 Future work	52
-----------------------	----

References	53-54
------------------	-------

FIGURE LISTS

Figure-1.1 : shows OCR in Mobile Devices	13
Figure-1.2 : shows Augmented reality on OCR	14
Figure-2.1 : shows Matrix matching.....	19
Figure-2.2 : shows Feature extraction.....	19
Figure-2.3 : shows ABBYY Business Card Reader.....	23
Figure-2.4 : shows Text Fairy.....	24
Figure-2.5 : shows Microsoft OneNote 2007.....	24
Figure-2.6 : shows SimpleOCR	25
Figure-2.7 : shows TopOCR	25
Figure-3.1 : shows Activity flow of the Application	27
Figure-4.1: Available Software pop up window	35
Figure-4.2: tess-two library add in a project	37
Figure-4.3: Tesseract trained data for English	39
Figure4.4: work flow diagram of project	40
Figure- 4.5 :Different mobile operator scratch cards	41
Figure-4.6. GUI of the home page	41
Figure-4.7: When nothing, OCR Failed	42
Figure4.8: OCR successful on pin number	42
Figure-4.9: extracted pin number	42
Figure-4.10: USSD gateway	43
Figure 5.1: Accuracy rate under different light conditions for different operators	48
Figure 5.2: Accuracy average in different light conditions	49
Figure 5.3: Accuracy rate for different aged people for different operators	50
Figure 5.4: Accuracy for different Mobile Phones	51

TABLE LISTS

Table 3.1	Inscope and outscope.....	29
Table 5.1	Accuracy Test for Banglalink in Sun Light Condition.....	45
Table 5.2	Accuracy Test for GrameenPhone in Sun Light Condition.....	46
Table 5.3	Accuracy Test for Teletalk in Sun Light Condition.....	47
Table 5.4	Accuracy Test Result for Individual Operators in Different Light Conditions.....	48
Table 5.5	Average Accuracy rate in different light conditions.....	49
Table 5.6	Accuracy rate for people of different ages.....	49
Table 5.7	Accuracy rate In terms of Different Mobile Phones.....	50

CHAPTER-1

INTRODUCTION

Ever-changing technologies have created its demand in the world. People like to use their smart phones as minicomputer and eager to do all most everything if possible. **OCR** application is a desirable application, especially since the smart phones these days come with high quality camera which can be used to scan the document or image and printed text into machine-encoded text.

Mobile recharge cards are very essential for us because of emerging number of mobile users. Recharge card is reliable than other techniques.

This is not easy to dial the right pin number while anyone is walking or busy in work. Most of them face difficulties dialing the pin number correctly at one time. Many of us have *presbyopia* and the older people are often the victims of it.^[1]

So, we have come to a solution of that problem by doing OCR on scratch card pin numbers. This is much better and efficient way to smartly recharge a mobile phone balance.

1.1 OPTICAL CHARACTER RECOGNITION

Optical character recognition, usually abbreviated to **OCR**, is the mechanical or electronic translation of scanned images of handwritten, typewritten, or printed text into machine-encoded text.

Imagine we have got a paper document - for example, magazine article or PDF contract your partner sent to you by email. Obviously, a scanner is not enough to make this information available for editing, say in Microsoft Word. All a scanner can do is create an image or a snapshot of the document that is nothing more than a collection of black and white or color dots, known as a raster image.

In order to extract and repurpose data from scanned documents, camera images or image-only PDFs, you need OCR software that would single out letters on the image, put them into words

and then - words into sentences, thus enabling you to access and edit the content of the original document.

First, the program analyzes the structure of document image. It divides the page into elements such as blocks of texts, tables, images, etc. The lines are divided into words and then - into characters. Once the characters have been singled out, the program compares them with a set of pattern images.

It advances numerous hypotheses about what this character is. Basing on these hypotheses the program analyzes different variants of breaking of lines into words and words into characters. After processing huge number of such probabilistic hypotheses, the program finally takes the decision, presenting user the recognized text.

1.2 OCR ON SMARTPHONES

Our interest is in enabling OCR on mobile phones. Mobile phones are one of the most commonly used electronic devices today. Commodity mobile phones with powerful microprocessors (above 500MHz), high resolution cameras (above 2 megapixels), and a variety of embedded sensors (accelerometers, compass, GPS) are widely deployed and becoming ubiquitous. By fully exploiting these advantages, mobile phones are becoming powerful portable computing platforms, and therefore can process computing-intensive programs in real time.

In this paper, we explore the possibility to build a OCR-based applications on mobile phones. We believe this mobile solution to extract information from physical world is a good match for future trend. However, camera-captured documents have some drawbacks. They suffer a lot from focus loss, uneven document lighting, and geometrical distortions, such as text skew, bad orientation, and text misalignment.

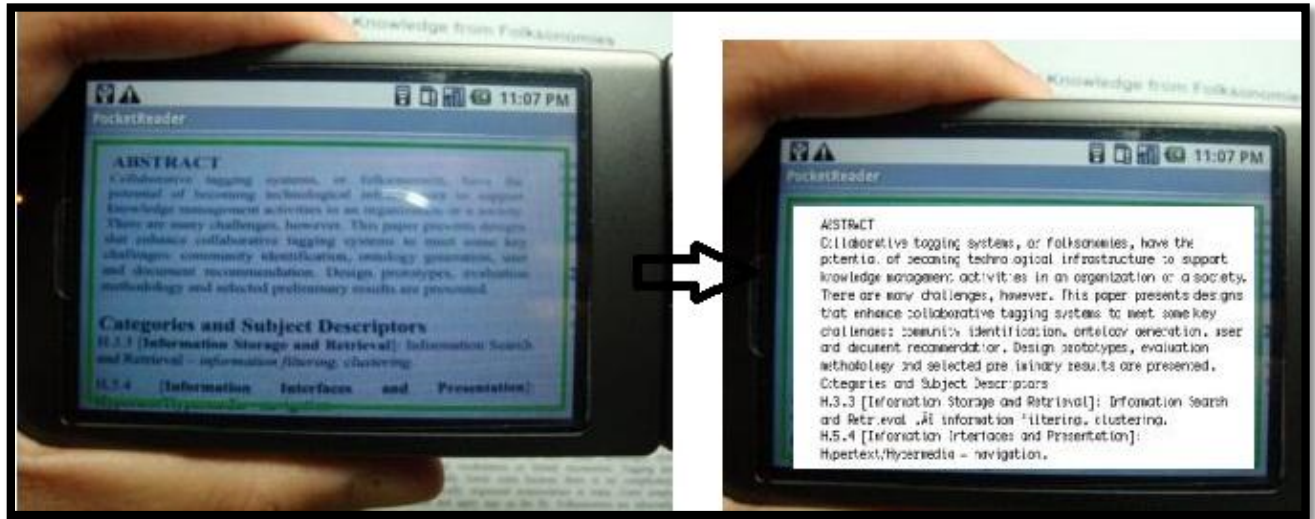


Figure-1.1: OCR in Mobile Devices

Moreover, since the system is running on a mobile phone, real time response is also a critical challenge. It utilizes embedded sensors (orientation sensor, camera) combined with image preprocessing suite to address those issues mentioned above. In addition, we have evaluated our extracted text by implementing an application called Mobile Scratch Card Recharge based on this OCR operation. Our experimental results demonstrate feasibility for building real-world OCR-based mobile applications.

1.3 AUGMENTED REALITY ON OCR

Augmented reality (AR) is cutting-edge technology that allows for a digitally enhanced view of the real world, connecting you with more meaningful content in your everyday life. With the camera and sensors in a Smartphone or tablet, AR adds layers of digital information – videos, photos, sounds – directly on top of items in the world around us. Hardware like Google Glass and Ather Labs 3D Augmented Reality glasses are all great products, of course. But the data is the special sauce that makes these tools work for us. ^[2]



Figure-1.2: Augmented reality on OCR

Like the Smartphone, augmented reality is also something neither for work nor at home. Instead it's a layer that can be applied to our home and work life. ^[3]

Augmented reality is changing the way we view the world -- or at least the way its users see the world. Picture yourself walking or driving down the street. With augmented-reality displays, which will eventually look much like a normal pair of glasses, informative graphics will appear in our field of view and audio will coincide with whatever we see. These enhancements will be refreshed continually to reflect the movements of our head.

OCR does not take into account what the item is that's being scanned, only the text. To further expand on this technology, we were recently asked to combine optical character recognition and augmented reality together and present a solution that provided a much stronger use case for the end user. ^[4]

The client asked us to provide an engine that would not only recognize a code comprised of printed text, but to also recognize the specific object being scanned, and thus engage the OCR component. To accomplish this, we called upon the open-source Tesseract project, hosted and maintained by Google.

1.4 BACKGROUND AND MOTIVATION

Mobile recharge cards are very essential for us because of emerging number of mobile users. There are other techniques of mobile phone recharge but recharge card is reliable than others.

Recharge cards are used by maximum users. Although recharge cards are more convenient for mobile users it has a problem (it helps us to keep our phone number secret). The problem is that if a user wants to recharge his mobile phone by recharge card he or she has to scratch the recharge cards silver layer, add prefix according to mobile operator specification and then put the pin number.

The process itself is not too difficult but a person in hurry finds it difficult and also if a person trying to recharge while walking finds it difficult too. Moreover, people having **presbyopia** finds it much more difficult as they need to wear glasses for reading purpose.

After observing these issues we have come with an idea that will make the mobile recharge easier than ever. Our idea is to build an Android based mobile application which will take the picture of the scratch card using the phone camera and collect the pin number and recharge the account.

The application will first take the picture and with the help of image processing it will search the boxed area where the pin number is written. Afterwards, it will catch the pin number from the image using OCR. Finally, the application will add the pin number with the subscriber prefix and recharge user's account balance.

Our target users are mostly the aged people even all the people. Nowadays android is a very popular OS for mobile phone and gradually Android phones become cheap. Our goal is to minimize the pain and save time of users.

1.5 ORGANIZATION OF THESIS

Chapter-1 describes the OCR idea on mobile devices and how augmented reality works on OCR. Background and motivation are also included in this section.

Chapter-2 constitutes of a review of the several research papers we have read to gather knowledge about OCR related works. Many ORC engines are available for Android like Tesseract

OCR engine, OCR algorithm, open source android application and we also review several papers related to text detection and translation in mobile.

Chapter-3 describes the proposal of the thesis and elaborates the methodology part.

In Chapter-4 we have the implementation process of our project. We have mentioned the environment set up process, tess-two library using process, how does tess-two work, trained data, USSD code running on OCRd Text.

Chapter-5 reviews the result of our thesis project. There are several accuracy results depending on different lights, users and mobile operators.

Chapter-6 summarizes the thesis work and the future work of this thesis.

CHAPTER-2

LITERATURE REVIEW

2.1 Methods of Optical Character Recognition

The main methods of the character recognition can be divided into the following groups by the used algorithm:

- Pattern systems
- Structural systems
- Feature systems
- Neuronal network systems

Each of the mentioned systems has both advantages and disadvantages which are namely the following: ^[E,F]

- Pattern algorithms are stable to small defects of the image and have sufficiently high recognition velocity. However even minor distortions of the image, which lead to the characters distortion, may influence negatively on the result of recognition
- Structural algorithms are very sensitive to the image defects. Besides, in contrast to the pattern and feature systems, effective automated learn procedures for structural systems are not implemented yet.
- Feature systems loose important information while calculating the character features and as a consequence make errors on objects classification referring them to the wrong classes.
- Although neuronal networks are able to recognise different fonts taking into consideration their defects and distortions, nevertheless they require complicated multi-layer structure and need a long training using sets of samples. This is not always practicable in industrial environment and at the same time the economic forces are of great importance here. ^[h]

Types

- Optical character recognition (OCR) – targets typewritten text, one glyph or character at a time.
- Optical word recognition – targets typewritten text, one word at a time (for languages that use a space as a word divider). (Usually just called "OCR")
- Intelligent character recognition (ICR) – also targets handwritten print script or cursive text one glyph or character at a time, usually involving machine learning.
- Intelligent word recognition (IWR) – also targets handwritten print script or cursive text, one word at a time. This is especially useful for languages where glyphs are not separated in cursive script.

OCR is generally an "offline" process, which analyzes a static document. Handwriting movement analysis can be used as input to handwriting recognition. Instead of merely using the shapes of glyphs and words, this technique is able to capture motions, such as the order in which segments are drawn, the direction, and the pattern of putting the pen down and lifting it. ^[G]

This additional information can make the end-to-end process more accurate. This technology is also known as "on-line character recognition", "dynamic character recognition", "real-time character recognition", and "intelligent character recognition".

2.2 OCR Algorithm^[b]

There are two basic types of core OCR algorithm, which may produce a ranked list of candidate characters.

2.2.1 Matrix matching^[c]

Matrix matching involves comparing an image to a stored glyph on a pixel-by-pixel basis; it is also known as "pattern matching" or "pattern recognition". This relies on the input glyph being correctly isolated from the rest of the image, and on the stored glyph being in a similar font and at the same scale.

This technique works best with typewritten text and does not work well when new fonts are encountered. This is the technique the early physical photocell-based OCR implemented, rather directly.

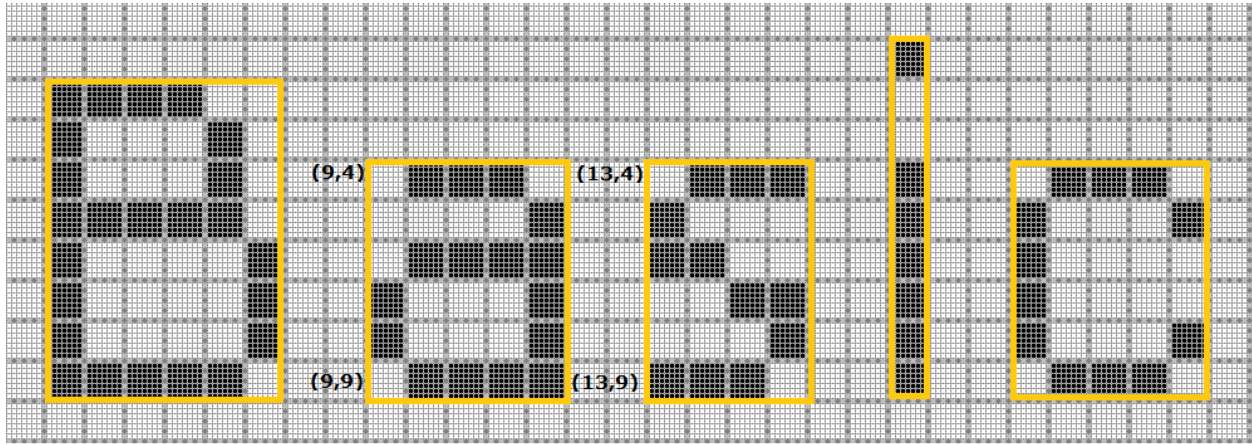


Figure 2.1: Matrix matching^[C]

2.2.2 Feature extraction

Feature extraction decomposes glyphs into "features" like lines, closed loops, line direction, and line intersections. These are compared with an abstract vector-like representation of a character, which might reduce to one or more glyph prototypes.

General techniques of feature detection in computer vision are applicable to this type of OCR, which is commonly seen in "intelligent" handwriting recognition and indeed most modern OCR software.^[d]

Nearest neighbor classifiers such as the k-nearest neighbor's algorithm are used to compare image features with stored glyph features and choose the nearest match.

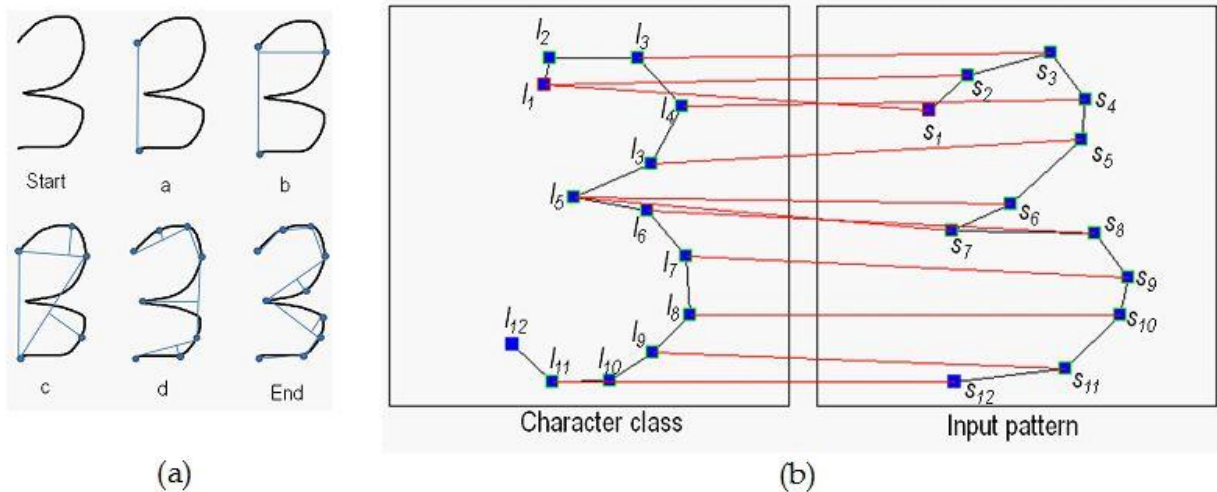


Figure 2.2: Feature extraction^[d]

Software such as Cuneiform and Tesseract use a two-pass approach to character recognition.

The second pass is known as "adaptive recognition" and uses the letter shapes recognized with high confidence on the first pass to better recognize the remaining letters on the second pass. This is advantageous for unusual fonts or low-quality scans where the font is distorted (e.g. blurred or faded).

2.3 ORC engines

Many ORC engines are available for Android. Most of them are online/cloud based OCR engine. Every kinds of image processing put a lot of presser on CPU. For this reason online/cloud based ORCs are much popular. ^[8,9,b,i] Some OCR for cellphones are,

- ABBYY Mobile OCR
- OpenRTK
- Tesseract

2.3.1 ABBYY Mobile OCR

ABBYY Mobile OCR is a powerful software development kit which allows developers of Android apps to integrate highly accurate optical character recognition technologies that convert images and photographs into manageable and searchable text.

The capabilities of the ABBYY Mobile OCR SDK built in many ABBYY mobile applications for Android, iOS and windows with OCR features. ^[8]

Platform Specific

- Android 2.2 and higher(ARM processors)
- Android NDK API is utilized
- Mobile OCR library is supplied as static library for Android
- Sample application is included

2.3.2 OpenRTK

OpenRTKis OS-independent OCR Toolkit so that it can be migrated from one OS to another OS easily and keeps the original performance. ^[9] At present, OpenRTKcan support **iPhone, iPad, Android, winCE** and other popular **Embedded OS**. Based on OpenRTKfor Embedded and Mobile System from ExperVision, clients can develop various OCR applications and get the following benefits:

- High OCR Performances
- Lowest Resource Consumption
- OCR Consulting Service

2.3.3 Tesseract

The Tesseract engine was originally developed as proprietary software at Hewlett Packard labs in Bristol, England and Greeley, Colorado between 1985 and 1994. In 2005 it had been released as open source by HP and University of Nevada, Las Vegas (UNLV). Tesseract development has been sponsored by Google since 2006.

Tesseract is considered one of the most accurate open source OCR engines currently available.

Tesseract version 2 and lower could only accept TIFF images of simple one column text as inputs. These early versions did not include layout analysis and so inputting multi-columned text, images, or equations produced a garbled output.

Since version 3.0 Tesseract has supported output text formatting, OCR positional information and page layout analysis. Support for a number of new image formats was added using the Leptonica library. Tesseract can detect whether text is monospaced or proportional.

The initial versions of Tesseract could only recognize English language. Now it can process many other languages including Bangla. Tesseract can be trained to work in other languages too.

Tesseract is a bare-bone OCR engine. It's suitable for use as a backend, and can be used for more complicated OCR tasks including layout analysis by using 3rd party frontends (GUI) like OCRopus, SunnyPage OCR, VietOCR etc. [9,10,11,12]

2.3.4 Online OCR API

This is a cloud based online OCR api. Efficient applications can be built using an OCR technology from smartphones apps (iPhone, Android, Blackberry, etc.) to ajax based web applications. We can focus on adding features to our programs while we focus on infrastructure maintenance. [16,17]

2.3.5 GOCR

GOCR/JOOCR is an OCR program, under GNU Public License. It converts images into text files. Joerg Schulenburg started the program, and now leads a team of developers. GOCR can be used with different front-ends, which makes it very easy to port to different OS and architectures. It can open many different image formats, and its quality have been improving in a daily basis. [13]

2.3.6 Orcad

Orcad is an optical character recognition program, and part of the GNU Project. It is free software, and is licensed under the GNU GPL.

Based on a feature extraction method, it reads images in portable pixmap formats known as Portable anymap, and produces text in byte (8-bit) or UTF-8 formats. Also included is a layout analyser, able to separate the columns or blocks of text normally found on printed pages. ^[14]

2.3.7 OCRopus

OCRopus™ is an OCR system written in Python, NumPy, and SciPy focusing on the use of large scale machine learning for addressing problems in document analysis.

The recommended system configuration is Ubuntu 12.10 (64 bit) with at least 4 Gbytes of memory and a fast processor.

Primary limitations right now are that performance on multi-column documents and documents containing images isn't very good. ^[15]

2.4 OCR SOFTWARE

2.4.1 ABBYY Business Card Reader

It's a app for capturing contact information from business cards into smartphone's contact list or into the CardHolder, the app's own data storage module. User don't need to tap in any data — simple take a snapshot of a business card and the app will do the rest. It's costs \$9.99 for Android platform. ^[8,20]

With ABBYY Business Card Reader you can:

- Instantly transfer business card data in 20 languages right onto your smartphone.
- Save contacts in the CardHolder, a special archive with convenient search, sorting and grouping functionality.
- Share contact data via e-mail, SMS or WiFi (thanks to the wireless communication technology AllJoyn from Qualcomm)
- Transfer contacts to another smartphone or computer.

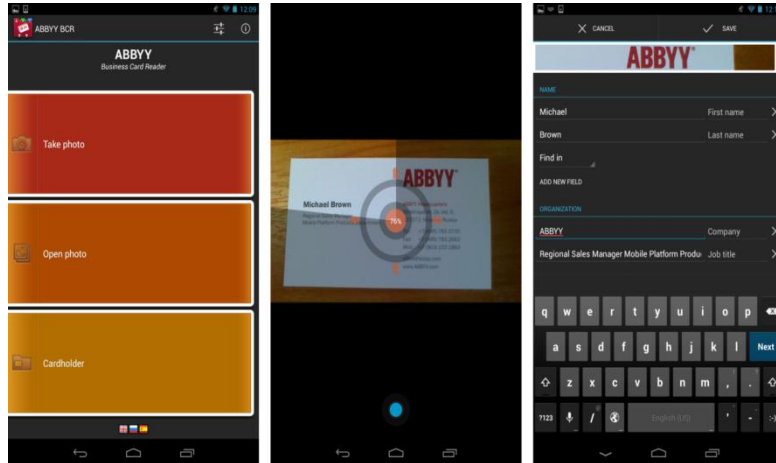


Figure 2.3: ABBYY Business Card Reader

System requirements

- OS Android version 2.2 and higher
- Camera with auto focus.

2.4.2 ABBYY Lingvo Dictionaries for Android

It is a dictionary application for Android and iOS smartphones and tablets that offers a fast and easy way to access up-to-date dictionary content for a variety of languages.

Electronic versions of dictionaries from the world's leading publishers, including Macmillan, Compact Verlag, K Dictionaries, and dictionaries compiled by ABBYY lexicographers. You can be confident in your translations. [8,19]

2.4.3 Text Fairy (OCR)

It's a kind of mobile phone text scanner. Text fairy helps us to convert images into document/text format. [22]

Features

- Edit extracted text
- Copy extracted text into the clipboard for use in other apps
- Convert the scanned page into PDF
- Recognizes printed text from more than 50 languages

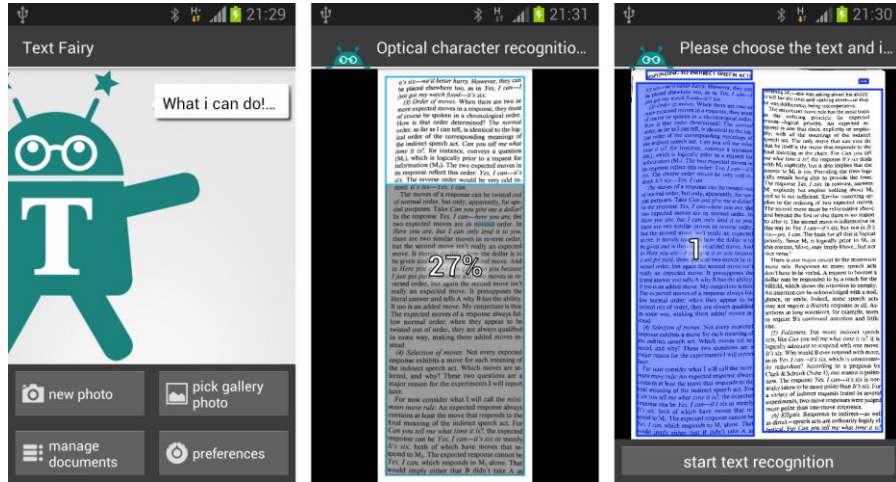


Figure 2.4: Text Fairy

Requires Android

- Version 23 and up.
- Varies with devices.

2.4.4 OCR Using Microsoft OneNote 2007

For the occasional basic OCR stuff, MS OneNote’s optical character recognition feature is a timesaver. You might have missed it” ; it’s called *Copy Text from Picture*.

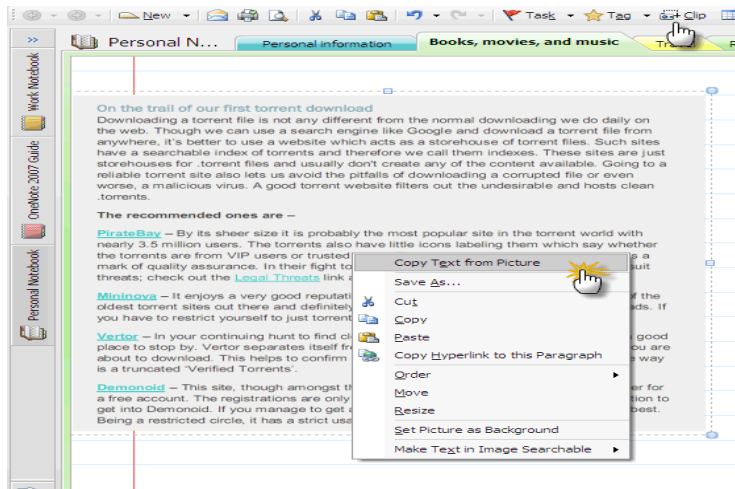


Figure 2.5: Microsoft OneNote 2007

OneNote is simplicity personified. But it’s not too great for handwritten characters or even fuzzy ones. But for a quick job, I am all for OneNote’s clip and paste. [9,26]

2.4.5 SimpleOCR

SimpleOCR can convert handwriting into text. But the software offers handwriting recognition only as a 14 day free trial. Machine print recognition though *does not* have any restrictions. [9,25]

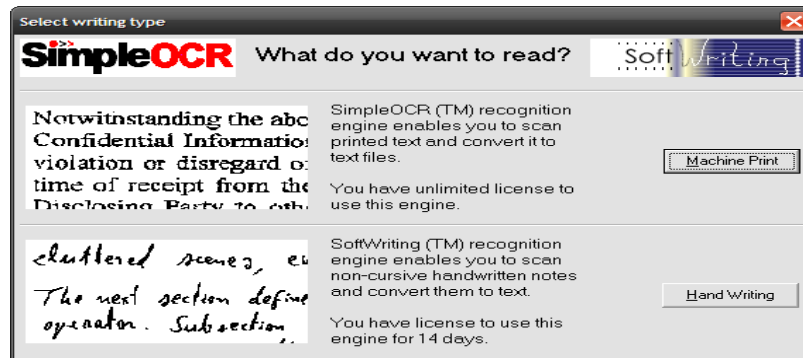


Figure 2.6: SimpleOCR

- SimpleOCR offers some control over the conversion through text selection, image selection and text ignore features.
- Conversion to text takes the process into a *validation* stage; a user can correct discrepancies in the converted text using an in-built spell-checker.
- The converted file can be saved to a doc or txt format.

2.4.6 TopOCR

TopOCR, in a breakaway from typical OCR software, is designed more for digital cameras (at least 3MP) and mobile phones along with scanners. Like SimpleOCR, it has a two window interface The source *Image* window and the *Text* window. [9,24]

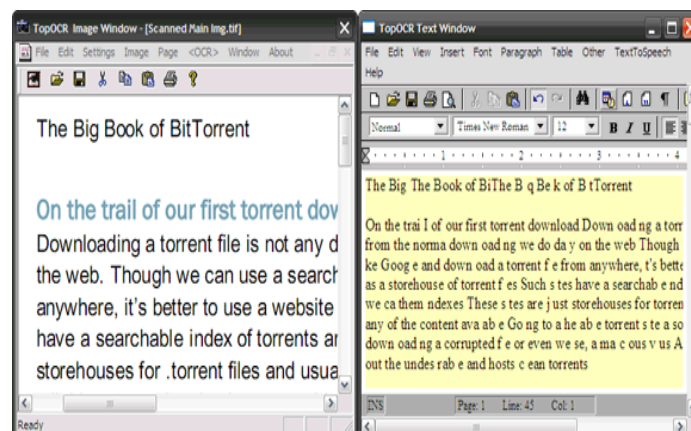


Figure 2.7: TopOCR

The image sourced from a camera or a scanner in the left window gets converted to the text format in the text editor on the right. The text editor functions like WordPad and can use Microsoft's *Text to Speech* engine.

- TopOCR functions well with straight oriented text but the usual failing of OCR with columned text remains.
- The software works with 11 languages.

2.4.7 FreeOCR

This free OCR software uses the TesseractOCR engine. It is thought of as one of the most accurate open source OCR engines available. FreeOCR is a simple Windows interface for that underlying code.^[9,13]

It supports most image files and multi-page TIFF files. It can handle PDF formats and is also compatible with TWAIN devices like scanners. FreeOCR also has the familiar double window interface with easy to understand settings. Before starting the one click conversion process, you can adjust the image contrast for better readability.

CHAPTER-3

PROJECT PROPOSAL

A Project Proposal itself describes a project. What the software going to do and what are the possible steps will be taken for the completion of the project will also found in the project proposal. Moreover, the analysis and requirements for the project will also be described in the project proposal. For this thesis project we are actually going to describe what we are going to do for our project and what will be the requirement, risk and rewards and other project dependencies in the project proposal.

3.1 Overview of Our Project

The proposed application will be created on android which will follow certain operations for the recharge purpose. At first user will have to start the application then select the desired operator and press the capture button on the application. The rest will be done by the application itself.

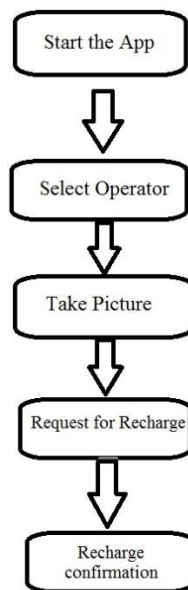


Figure 3.1: Activity flow of the Application

3.1.1 Mission

Scratch card recharge needs a subscriber prefix and the secret pin to recharge the phone. For those who need power glass for reading it is difficult for them if they do not have their glass outside home. Moreover, recharging while walking or travelling is also difficult. Most of the time they enter wrong codes which make them to re-enter the pin along with the subscriber prefix. If a person enters wrong pin three times consecutively he or she will not be able to recharge that day. Moreover, finding a recharge shop open at night is not only difficult but also impossible then recharge card is the only way to recharge. Recently online recharge option has been avail for the mobile users but that is not safe at all.

Android based easy mobile recharge application provides mobile phone users to recharge their phone using the camera. They just need to scratch the card, open the application then set the camera in front of the card and the application will do the rest.

The application will have easy to use user interface, subscriber choosing option, confirmation message after recharge is successful.

The application only needs an android phone which is now available for each of the mobile users at cheaper rate. Users having eye problem do not need their glass for recharging purpose.

3.2 Goal

This project will produce an android application by which users can easily recharge their mobile balance through scratch cards. The app will reduce the spend time for recharging balance. It will obliterate the dialing problem. It will have a high-quality appearance in smart phone. The smart phone users will find a real time image processing experience on few seconds. This will prevent cheating or the submission of false information.

3.3 Scope

We will build the application for android phones which will be easy to use for maximum mobile phone users. We assume that the application will recharge the phone on the first attempt and the processing time will be admissible.

We are trying to make it somehow manual for the users as they need to select the subscribers because it might not be done in our project time frame but if possible we will integrate automatic subscriber selection later.

Most of the mobile users will use this application as this application will reduce the amount of difficulties they face due to mobile recharge. After we successfully build this application we will upload it in the website where users can search and download it on their phone.

We want to focus on the application itself, and the features of that application that help build a good android community. Works with Android OS that most smart phone community are already familiar with.

Table 3.1: In Scope and Out Scope

In Scope	Out Scope
Build an android application	Build application for Symbian, iOS , Windows Phone, BlackBerry and others.
Working in only mobile scratch cards. Ex. Airtel, Robi, Banglalink, Grameenphone	Working with Other Scratch cards. ex. Qubee, Banglalion
System work after removing the silver shading area on Scratch cards	Automatically detect the pin without removing shading
Work only with minimum standard of image.	Work with any kind of image.

3.4 Deliverables

The deliverables of our project will be: An android application, Customization guide, online help

3.5 Risks and Rewards

There is a potential conflict in our application whether the customers accept it cordially or reject it. We can only succeed if users find the app appealing, and mobile operators find it worthy of making more revenue. We already have a design in mind that will address this risk and we will review it near future.

There are significant technical difficulties in building the application. Although our team has experience with the relevant tools and technologies, we will certainly make some mistakes which may ruin the whole project.

The schedule for this project is very short period of 12 months. As we are only 3 members on our project, it will be difficult to adjust it. We will manage this by planning a conservatively

scoped functional core and series of functional enhancements that can be individually slipped to later releases if needed.

If we accomplish the elements of our plan, our application will generate traffic which will result in more scratch card users.

3.6 Target Audience and Benefits:

3.6.1 Target Audience:

Smart phone users having android OS are the target audience. All kind of people will have fun by using the application. Especially people having vision problem or presbyopia will exult after using this. Moreover, women who feel unsafe with recharging from recharge stores as they have to give their phone number to the store for recharge purpose.

3.6.2 Benefits to Customers

The customer of the project will be the old-aged people, the people who find it difficult to recharge from pre-paid card and who want to keep their number safe from unwanted people.

Benefit for End Users:

Reliable medium for recharging. It will save time for recharging. Their number will be safe. At present users have to give their number for recharging. Moreover, aged people will find it easier than ever.

3.6.3 Potential Downside

When users try to recharge balance, they may try it with already the used one. It causes sending request to server many times. Usually it increases overhead and hampers the system with unwanted request. System may be slow time to time depending on image processing.

3.7 Project Plan

The project plan will consist the overall plan of the project as well as the methodologies

3.8.1 Summary of Project

An android OS based auto digital cash recharge from a Scratch Card. Operator selection by user. Recharge will be automatic process. Built in camera required. The project team is the authors of this thesis paper. All of them are undergraduate students of BRAC University.

3.7.2 Risk Management

The main risks of this project are:

- There is a potential conflict between the goals of a high-quality appearance and one that is completely customizable. We can only succeed if users find our application as a suitable way of recharge. We already have a design in mind that will address this risk and we will add more features to make it more suitable for daily use.
- There are significant technical difficulties in building an application. This will be a risk because one person on our team has much experience with the relevant tools and technologies. On the other hand others will learn, we will certainly make some mistakes in doing suboptimal choices. We will address this risk by scoping the project such that we have enough time to train and to review the design and implementation.
- The schedule for this project is very short. We will manage this by planning a conservatively scoped functional core and series of functional enhancements that can be individually slipped to later releases if needed.
- The performance of the system will be significantly impacted by the decisions made during the design task. None of our current team members has experience with effective UI design for different devices. To address this, we will arrange a review meeting with an experienced person or hire a consultant from the software market.
- We could be underestimating known tasks. To avoid this we will maintain a discussion mailing panel. Which helps us to know how we moving toward our goals.
- We could have misunderstood the user's requirements. Therefore, we will always interact with some of our target users.
- We could face major difficulties with the technology chosen for this project. If this will become a big issue, we will definitely switch into new platform soon. This event can play an unexpected role in achieving our goal.
- We could have low quality that demands significant rework.
- We could lose resources. E.g., team members could get sick, spend time on other projects, or quit even hardware failure. We will try to finish our main tasks within project timeline.

3.7.3 Project Planning Dependencies

This project conflicts or compete for resources with team member's student life. We have determined how many hours each person can actually dedicate to this project. The same human or machine resources could be allocated for planning of future versions during this release time period. This project not depends on the success of any other project. This project is stands alone. Our recourses are limited. It may hamper the release date of the project.

3.9 Requirements and Specification

The requirement and specification of the project consist of the agreed goals and necessary environment dependencies for this project.

3.8.1 Environment

The environment necessary for the development for the project is Linux (Ubuntu 12.04) operating system for development, Android SDK, ADT Bundle, Android NDK, A smartphone with minimum android version 2.3.3.

3.8.2 Stakeholders

Stakeholders for this project are the users who will use the application generated by our project. They are the main focus of the project as the success of the project depends on them.

3.8.3 User Stories

In order to recharge the mobile phone users have to go to the store and give their number for this purpose. As a result some users get phone calls from unwanted sources specially businessmen and women are the main victims. Moreover, old aged people always face difficulties recharging using recharge cards. Often they are unable to recharge from the recharge cards. The idea of this project actually came up for eradicating the problems those people often face.

3.8.4 Functional Requirements

The application will take the image from the pin number portion of the recharge card. The application itself will continuously extract number from the image until user gets the exact pin number. It will use OCR operation for extracting number. Moreover, it will automatically add prefix for the specific operator for the recharge purpose. Finally, it will request for recharge as per users confirmation.

3.8.5 Non-Functional Requirements

The non-functional requirement will be focused on the time that the application will take. It should not take more than one minute for the whole recharge process starting from the operator selection part. However, the autofocus might take more time for focusing but it should not be more than ten seconds.

3.8.6 Environmental Requirements

Hardware requirement for running the application is an android smartphone with minimum android version 2.3.3, at least 256 mb ram and has at least 30 mb free space in phone memory.

CHAPTER – 4

IMPLEMENTATION

We have worked on Ubuntu 12.04 LTS operating system. We develop the application for android platform and install a Java programming environment called Eclipse. We use Eclipse as part of the Google Android Developer Tools (ADT) bundle. In addition, we build NDK on tess two library for OCR .

The workflow diagram later shows the activity of the application.

4.1 ENVIRONMENT SET UP

4.1 .1 ADT set up ON UBUNTU

1. First head over to:

<http://developer.android.com/sdk/index.html>

and download the latest ADT bundle - 64 bit version:

```
adt-bundle-linux-x86_64.zip
```

2. Start a terminal and unzip the downloaded file:

```
unzip ~/Downloads/adt-bundle-linux-x86_64.zip
```

3. To start Eclipse, execute in a terminal:

```
cd adt-bundle-linux-x86/eclipse  
./eclipse
```

4.1.2 NDK PLUGIN IN ECLIPSE

On help bar then on select 'Install New Software'.

A new window called 'Available Software' will pop-up

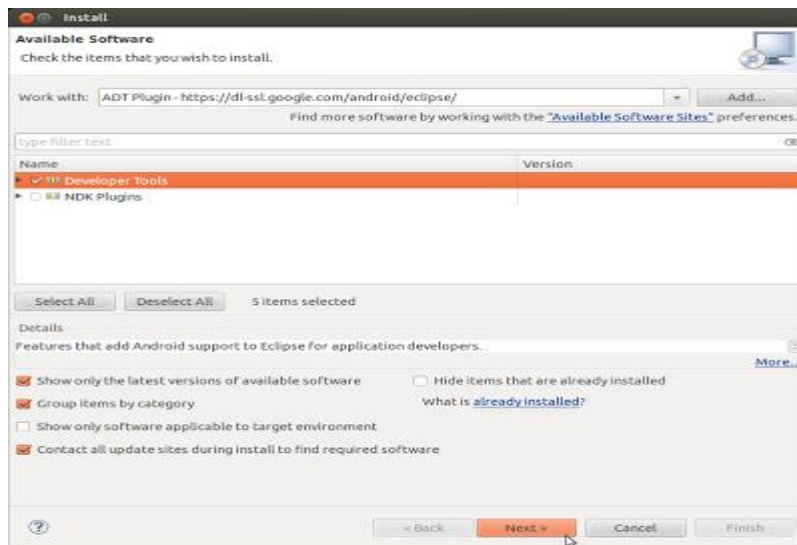


Figure-4.1: Available Software pop up window

Download the ADT plug-in in the Eclipse

Add "ADT plug-in" in the Name

Add "https://dl-ssl.google.com/android/eclipse/" in the Location

Then we install the ndk-plugin.

4.1.3 NDK BUILD WITH TESS-TWO

NDK is a toolset that allows you to implement parts of your app using native-code languages such as C and C++. For our application we reuse existing code libraries (tess-two) written in C language. We have used Android framework provides to use native code. We have worked on the NDK r9-d

```
export PATH=$PATH:<ndk src path directory>
cd <tess-two project directory>/tess-two
ndk-build
```

4.2 USING OF TESS-TWO LIBRARY

Tesseract is an Open Source OCR engine, available under the Apache 2.0 license. It can be used directly, or (for programmers) using an API. It supports a wide variety of languages.

Tess-two is a fork of Tesseract Tools for Android (tesseract-android-tools) that adds some additional functions. Tesseract Tools for Android is a set of Android APIs and build files for the Tesseract OCR and Leptonica image processing libraries.

This project works with Tesseract v3.03. The required source code for Tesseract 3.03 and Leptonica 1.70 is included within the tess-two/jni folder.

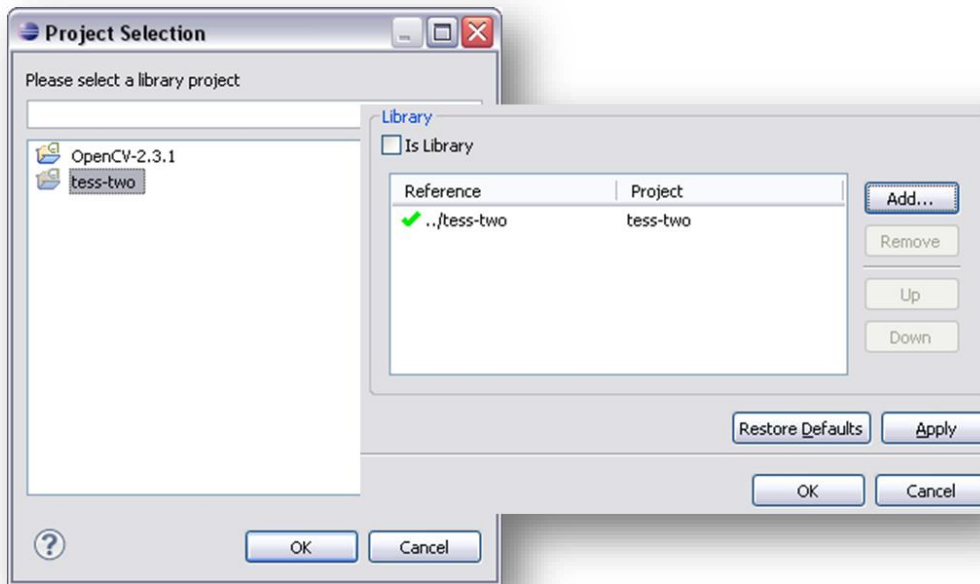


Figure-4.2: tess-two library add in a project

This is now ready to OCR any image using the library.

First, we need to get the picture itself. For that, we use a simple code to capture the image. After we have the bitmap, we just need to perform the OCR which is relatively easy. Be sure to correct the rotation and image type by doing something like

```
// path = path to the image to be OCRed
ExifInterface exif = new ExifInterface(_path);

int exifOrientation =
exif.getAttributeInt(ExifInterface.TAG_ORIENTATION,
ExifInterface.ORIENTATION_NORMAL);
```

Rotating Bitmap & convert to ARGB_8888, required by tess-two

```
bitmap = Bitmap.createBitmap(bitmap, 0, 0, w, h, mtx, false);
```

We have the image in the bitmap, and we can simply use the TessBaseAPI to run the OCR like:

```
TessBaseAPI baseApi = new TessBaseAPI();
```

4.3 USING OF TRAINED DATA

We have used tess-two which compiles the Tesseract and Leptonica libraries for use on the Android platform. It contains an Android library project that provides a Java API for accessing natively-compiled Tesseract and Leptonica APIs.

To use the trained data we have to keep trained data to subdirectory named "tesdata".

Filename ▼	Summary + Labels ▼
tesseract-ocr-3.02.grc.tar.gz	Ancient Greek Language data for Tesseract 3.02
tesseract-ocr-3.02.epo_alt.tar.gz	Esperanto alternative language data for Tesseract 3.02
tesseract-3.02.02-win32-lib-include-dirs.zip	VC++ libraries of Tesseract OCR 3.02.02 (32-bit)
tesseract-ocr-setup-3.02.02.exe	Windows installer of tesseract-ocr 3.02.02 (32-bit)
tesseract-ocr-3.02.02.tar.gz	Tesseract OCR 3.02.02 Source Featured
tesseract-ocr-3.02.02-doc-html.tar.gz	Tesseract 3.02.02 html doc Featured
tesseract-ocr-3.02.eng.tar.gz	English language data for Tesseract 3.02

Figure-4.3: Tesseract trained data for english

Google provide the trained data for English language. We have worked on this trained data. A trained data file “ tesseract-ocr-3.02.eng.tar.gz ” for English language data for tesseract 3.02 must be extracted to a subdirectory named “tesdata”.

4.4 WORK FLOW OF THE SYSTEM

The following is the workflow diagram of the software. After selection of the operator, the process requests the camera and starts to do OCR operation. If OCR operation is failed, it runs the OCR until it successfully does the OCR operation or operation is stopped by user.

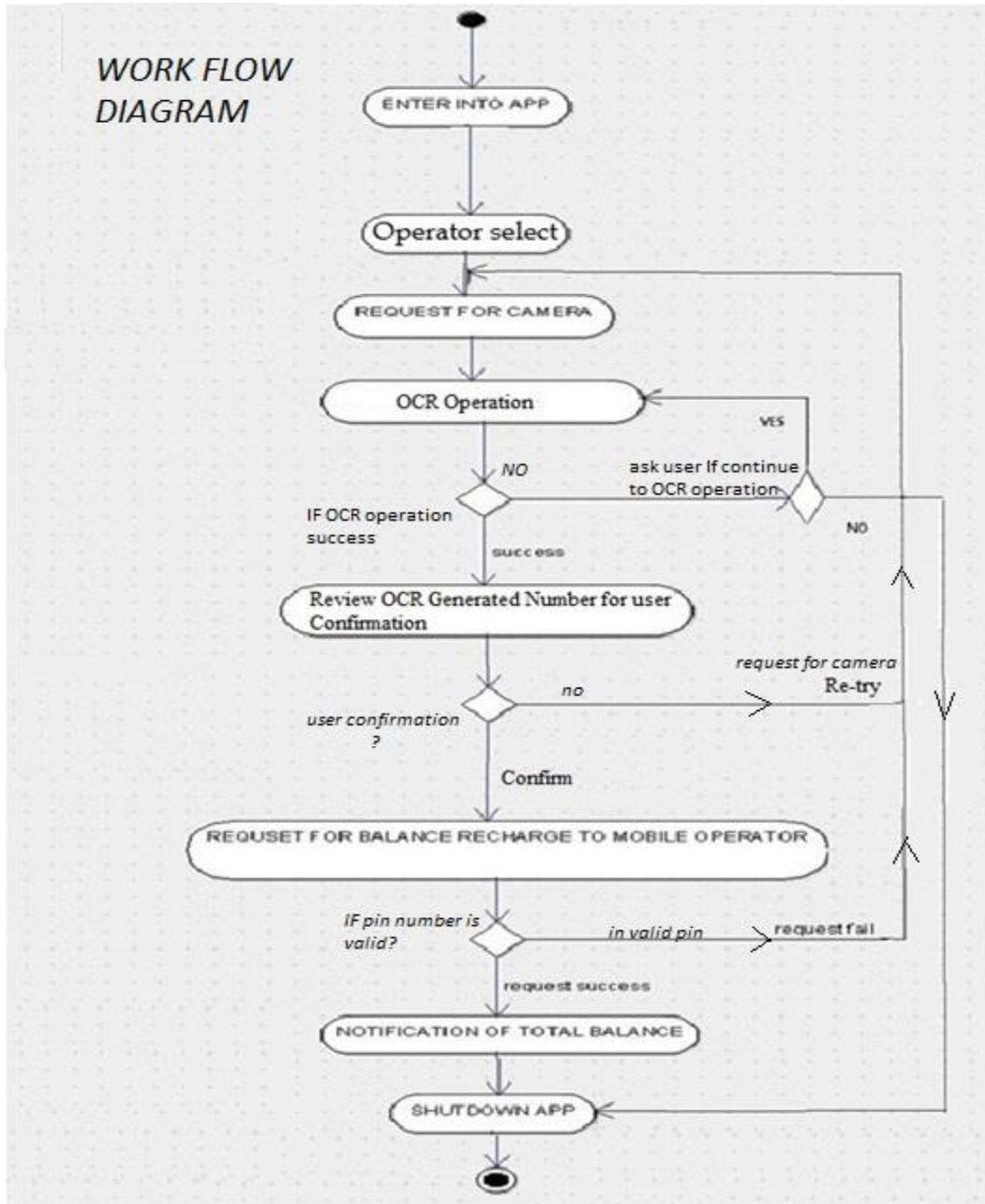


Figure-4.4: work flow diagram of project

Step-1: Scratch the mobile recharge card clean. So that the noise level condition is limited.



Figure-4.5: Different mobile operator scratch cards

Step-2: After starting the application select the mobile desired operator.



Figure-4.6: GUI of the home page

Step-3: Camera Activity starts the OCR Operation

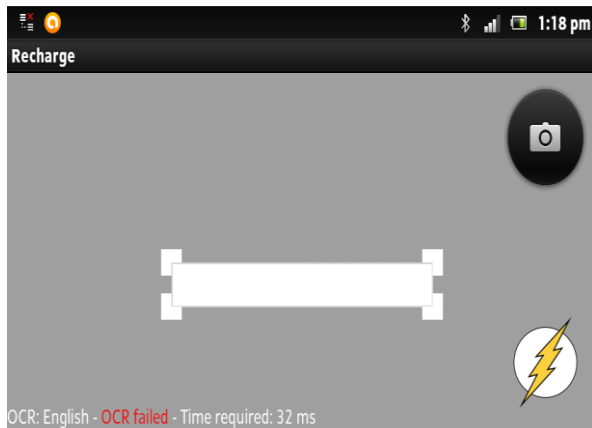


Figure-4.7: When nothing, OCR Failed

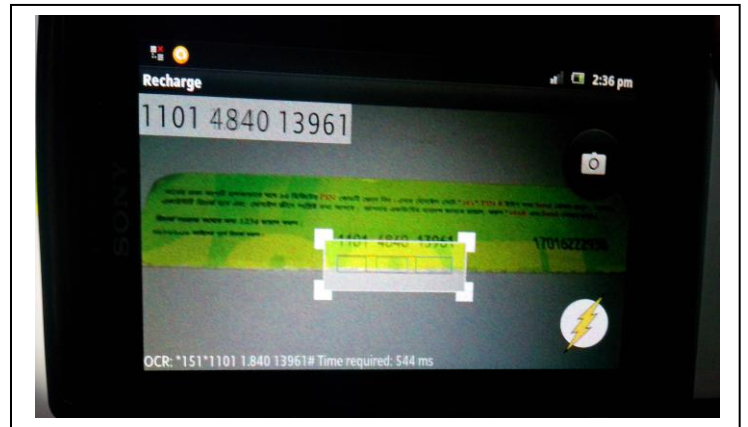


Figure-4.8: OCR successful on pin number

Step-3: Extract the pin number when ocr is successful

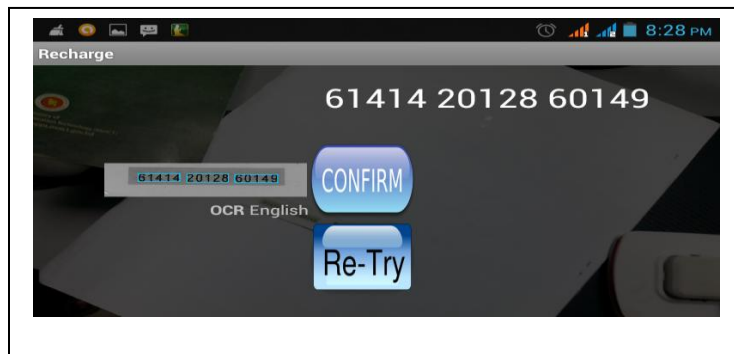


Figure-4.9: extracted pin number

4.5 INTREGRATION OF USSD ON EXTRACTED PIN NUMBER

USSD is a protocol used by GSM cellular telephones to communicate with the service provider's computers. USSD stands for Unstructured Supplementary Services Data. A gateway is the

collection of hardware and software required to interconnect two or more disparate networks, including performing protocol conversion. [27]

We have run the extracted text (pin number of scratch card) as a USSD code and send request to the cell phone operator.



Figure-4.10: USSD gateway

In USSD services there are short cuts as well. Rather than accessing a service using for an example *500#, and then browsing through the menu, someone may be able to access the ultimate service intended to use directly by, *500*3*2*86568855487*9700*123#. This is another advantage of USSD services where a knowledge user would be capable of performing the task faster.

In our application to recharge balance:

“ * ” + “ Operator-prefix ” + “ * ” + “ Pin number ” + “ # ”

Example for Airtel bd :

Dial *787* 6482371214717856 #

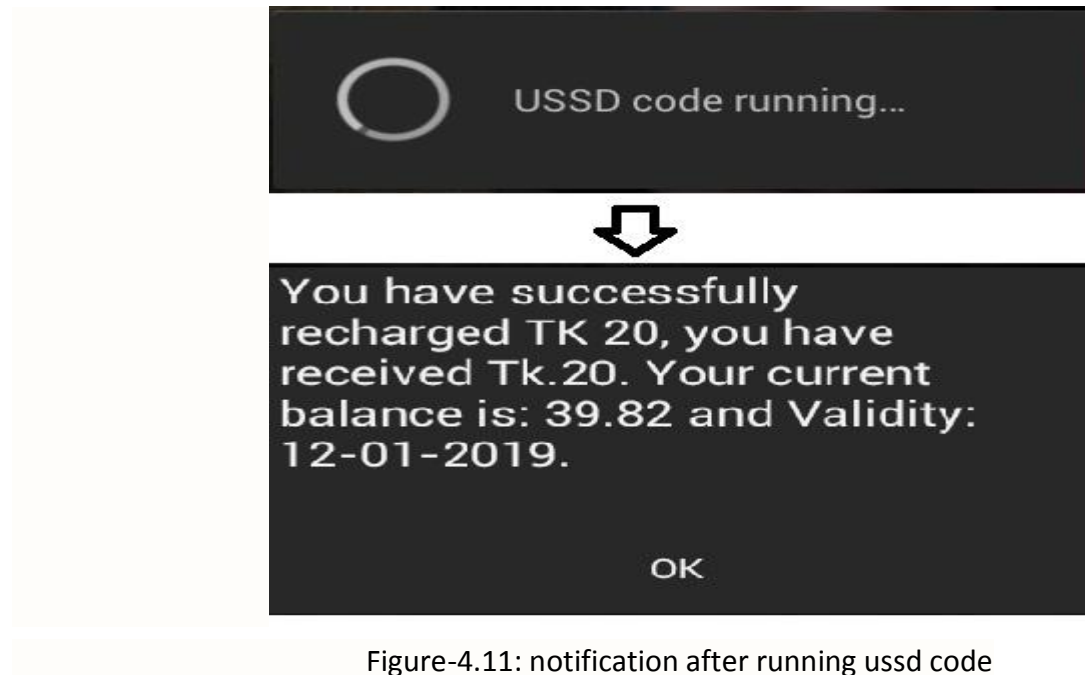


Figure-4.11: notification after running ussd code

CHAPTER-5

RESULT

This chapter includes the results we have found by testing our application on different conditions. We have considered different criteria's for calculating the result which is different light conditions, for different aged people, in different devices for different operators

5.1 Accuracy Test for Different Operator(s) in Different Light Conditions (Individual)

We have collected 10 pre-paid recharge cards for each operators for testing our applications accuracy rate. We have done the accuracy test in different light conditions: Sun light, Flash light and Normal light. We have calculated the accuracy rate in terms of the recognized pin numbers by tesseract from the total pin numbers that a recharge card have and created table for each operator in different light conditions which in total is fifteen tables. Below we will be discussing three table for each operators in Sun light condition in order to show how we have calculated the accuracy rate. After that we will show the total average of accuracy rate under different light conditions for different operators.

Table 5.1

Accuracy Test for Banglalink in Sun Light Condition:

Operator: Banglalink	Total Pin Numbers	Recognized Pin Numbers	Accuracy Rate (%)	Accuracy Average
Card 01	15	9	$(9*100/15)=60$	76.663
Card 02	15	11	$(11*100/15)= 73.33$	
Card 03	15	10	$(10*100/15)=66.66$	
Card 04	15	10	$(10*100/15)= 66.66$	
Card 05	15	12	$(12*100/15)=80$	
Card 06	15	13	$(13*100/15)=86.66$	
Card 07	15	14	$(14*100/15)=93.33$	
Card 08	15	10	$(10*100/15)= 66.66$	
Card 09	15	11	$(11*100/15)=73.33$	
Card 10	15	15	$(15*100/15)=100$	

From table 5.1 we figure out the accuracy rate in sun light condition generated by tesseract. The average accuracy rate for Banglalink in Sun light is 76.663%.

Table 5.2

Accuracy Test for GrameenPhone in Sun Light Condition:

Operator: GrameenPhone	Total Pin Numbers	Recognized Pin Numbers	Accuracy Rate (%)	Accuracy Average
Card 01	16	9	$(9*100/16)=56.25$	74.289
Card 02	16	11	$(11*100/16)=68.75$	
Card 03	16	10	$(10*100/16)=62.5$	
Card 04	16	15	$(15*100/16)=93.75$	
Card 05	16	12	$(12*100/16)=75$	
Card 06	15	13	$(13*100/15)=86.66$	
Card 07	15	14	$(14*100/15)=93.33$	
Card 08	15	10	$(10*100/15)= 66.66$	
Card 09	15	11	$(11*100/15)=73.33$	
Card 10	15	10	$(10*100/15)=66.66$	

From table 5.1 we figure out the accuracy rate in sun light condition generated by tesseract. The average accuracy rate for GrameenPhone in Sun light is 74.289%.

Table 5.3

Accuracy Test for Teletalk in Sun Light Condition:

Operator: Teletalk	Total Pin Numbers	Recognized Pin Numbers	Accuracy Rate (%)	Accuracy Average
Card 01	13	10	$(10*100/13)=76.92$	79.234
Card 02	13	11	$(11*100/13)=84.61$	
Card 03	13	8	$(8*100/13)=61.53$	
Card 04	13	10	$(10*100/13)=76.92$	
Card 05	13	12	$(12*100/13)=92.30$	
Card 06	13	13	$(13*100/13)=100$	
Card 07	13	9	$(9*100/13)=69.3$	
Card 08	13	10	$(10*100/13)=76.92$	
Card 09	13	7	$(7*100/13)=53.84$	
Card 10	13	13	$(13*100/13)=100$	

From table 5.1 we figure out the accuracy rate in sun light condition generated by tesseract. The average accuracy rate for Teletalk in Sun light is 79.234%.

Table 5.4

Accuracy Test Result for Individual Operators in Different Light Conditions:

	Sun Light	Flash Light	Normal Light
Banglalink	76.663	69.456	64.76
GP	74.289	60.62	48.64
Teletalk	79.234	74.23	63.08
Airtel	78.12	67.5	45
Robi	74.52	62.6	64.48

From table 5.4 we have figure out the average accuracy rate for different operators in different light conditions. By analyzing the table 5.4 we found that Teletalk has the highest accuracy in Sunlight which is 79.234% as well as in Flash light which is 74.23%. Whereas, Banglalink has the highest accuracy rate in Normal light. We can also analyze the accuracy result from the figure 5.1 where we can see the accuracy rates shown for different operators in different light conditions. In the figure 5.1 the x-axis shows different operators where three different columns are for different light condition and the y-axis shows the precession.

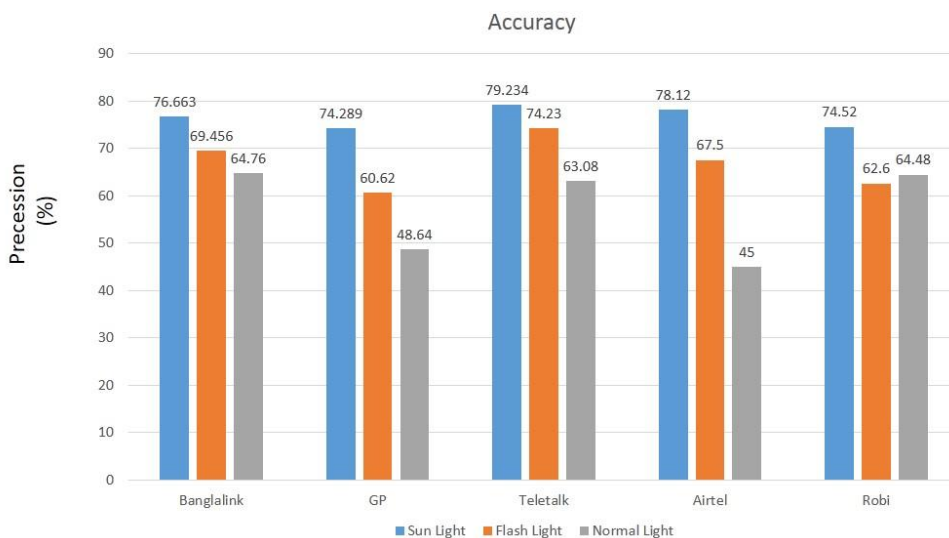


Figure 5.1: Accuracy rate under different light conditions for different operators.

5.2 Accuracy Test average for Different Operator(s) in Different Light Conditions

Previously, we have shown the accuracy rate for different operators in different light conditions individually. Here we will show the average accuracy in different light conditions for different operators.

Table 5.5

Average Accuracy rate in different light conditions:

	Precession
SunLight	76.5652
FlashLight	66.8812
Fluorescent	57.192

From table 5.5 we can see the highest accuracy rate is in Sun light condition which is 76.5652%. We can see the figure 5.2 generated by the value of table 5.5 where x-axis shows the different light conditions and y-axis shows the precession.

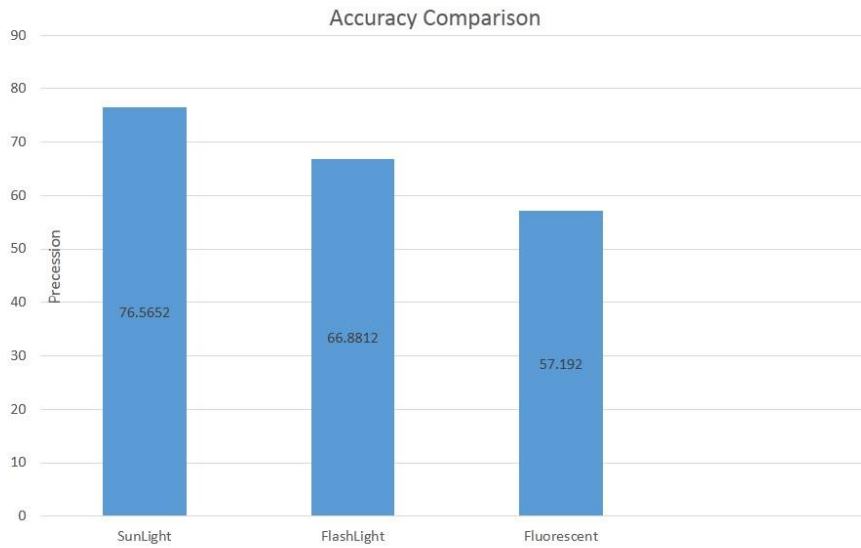


Figure 5.2: Accuracy average in different light conditions.

5.3 Accuracy rate for people of different ages

We have given application to people of different ages and calculated the accuracy for different operators. In table 5.6 we will show the class of different age of people and the average accuracy for them.

Table 5.6

	Teletalk	GP	Airtel
Age(12-18)	72.3	68.1	66.5
Age(18-24)	82.74	78.7	69.67
Age(24-32)	76.345	82	75.6
Age(32+)	59.7	68.1	55.43

From table 5.6 we can see that people of age 18-24 has the highest accuracy rate for teletalk which is 82.74%. For GrameenPhone and Airtel people of age 24-32 has the highest accuracy rate which are 82% and 75.6%. We can see the figure 5.3 for best view of the accuracy rate.

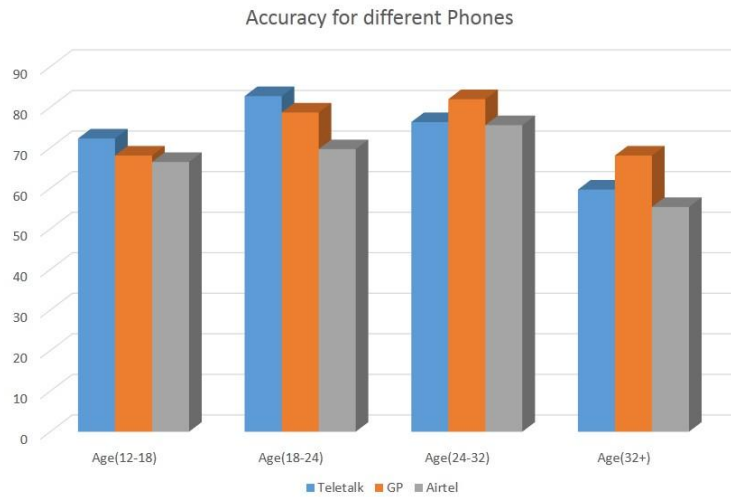


Figure 5.3: Accuracy rate for different aged people for different operators.

5.4 Accuracy rate In terms of Different Mobile Phones

We have tested our application in different mobile phones and collected the data for calculating the accuracy rate. Table 5.7 shows the average accuracy rate for different smart phones for different operators.

Table 5.7

	Teletalk	GP	Airtel
Sony	74.76	67.64	59.32
Samsung	71.34	78.7	67.7
Symphony	66.45	70	73
Walton	59.7	66	68.3

From table 5.7 shows that Sony mobile phone has the highest accuracy for teletalk which is 74.76. Samsung mobile phone has the highest accuracy for GrameenPhone which is 78.7. And Walton has the highest accuracy for Airtel which is 68.3. We can analyze the table 5.7 with the figure 5.4 for best view of accuracy result.

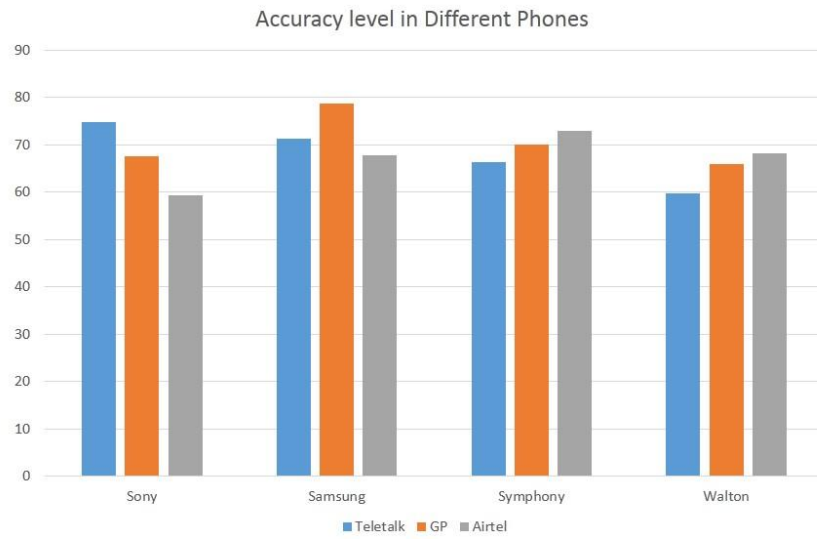


Figure 5.4: Accuracy in different Mobile Phones

CHAPTER-6

CONCLUSION AND FUTURE WORK

We have demonstrated our OCR based android application which can extract pin number from the recharge card and recharge the phone balance automatically. We have developed the application using open source projects such as tess-two and an OCR based project named android-ocr. We have conducted few tests on our application in order to figure out the accuracy level of our application in different condition. The result shows, the highest accuracy is around 76% in sunlight, around 66% in Flashlight and around 57% in normal light condition. Considering the result we can say that our application works better in sun light condition compared to flash light and normal light condition.

The main problem we found that the OCR displays unexpected characters if it detects the presence of opaque substance on the scratch card. The problem occurred mainly because we have not worked on this part as our main focus was to extract the pin number and recharge successfully. The application might sometimes might not work properly due to insufficient amount of light and light reflection from the scratch card. We could not reduce this problem because it was beyond our limitation.

6.1 Future work

Our future work will be increasing our application's accuracy rate by at least 15% so that it gets better user response. At present we have built our application only for android phones but in future we will built the application for iOS and windows phone.

References:

Publication:

- A. <http://robotics.usc.edu/publications/media/uploads/pubs/635.pdf>
- B. http://its.lnu.edu.ua/edocs1/1/docs/_sort/Books%20Collection/Flynn%20R.R.%20%28ed.%29%20%20Computer%20Sciences%20%28vol.%20%29%20%282002%29%28en%29.pdf#page=164
- C. <http://www.codeproject.com/Articles/476142/Optical-Character-Recognition>
- D. <http://www.intechopen.com/books/advances-in-character-recognition/online-handwritten-chinese-japanese-character-recognition>
- E. Rice S., Nagy G., Nartker T., Optical Character Recognition: An Illustrated Guide to the Frontier, Springer, 1999.
- F. Mori S., Optical Character Recognition, Wiley-Interscience, 1999.
- G. Parker J., Algorithms for Image Processing and Computer Vision, John Wiley & Sons Inc, 1996.
- H. Callan R., The Essence Of Neural Networks, Prentice-Hall, 1999.
- I. P.E. Gill, W. Murray and M.H. White. Practical Optimization, Academic Press, 1981.

Website:

1. <http://en.wikipedia.org/wiki/Presbyopia>
2. <http://www.howstuffworks.com/augmented-reality.htm>
3. <http://www.wikitudo.com/developer/news/-/blogs/text-recognition-and-augmented-reality>
4. <http://techcrunch.com/2013/12/22/augmented-reality-world-is-almost-everywhere/>
5. <http://www.ipr.ira.uka.de/get.php?id=576>
6. <http://www.google.com/patents/US4339745>
7. <http://www.iitg.ernet.in/rcilts/phasel/ocr2.html>
8. <http://www.abbyy.com/>
9. <http://www.expervision.com/ocr-software>
10. <http://tesseract-ocr.googlecode.com/svn-history/r719/trunk/doc/tesseract.1.html>
11. <http://code.google.com/p/tesseract-ocr>
12. <https://github.com/rmtheis/tess-two>
13. <http://jocr.sourceforge.net/>
14. <http://www.gnu.org/software/ocrad/ocrad.html>
15. <https://code.google.com/p/ocropus/>

16. <http://ocrapiservice.com/>
17. <https://github.com/smart-mobile-software/ocrapiservice>
18. <http://www.topocr.com/snaphelp.html>
19. <https://play.google.com/store/apps/details?id=com.abbyy.mobile.lingvo.market>
20. <https://play.google.com/store/apps/details?id=com.abbyy.mobile.bcr>
21. <https://play.google.com/store/apps/details?id=com.abbyy.mobile.textgrabber.full>
22. <https://play.google.com/store/apps/details?id=com.renard.ocr>
23. www.free-ocr.com/
24. www.topocr.com/
25. www.simpleocr.com/
26. www.howtogeek.com/howto/.../ocr-anything-with-onenote-2007-and-2010
27. <http://www.katoida.eu/using-ussd-for-survey-polls-vs-sms-by-mireille-raad/>