

Smart Calculator using OCR

Student Name : Sandip Thapa (NP03A180140)

Section :

Module Leader :

Submitted on :

Word Count :

Contents

[1. Introduction 1](#_Toc41549764)

[1.1. Introduction to OCR 1](#_Toc41549765)

[1.2. Problem Domain 1](#_Toc41549766)

[1.3. This project as a solution 2](#_Toc41549767)

[2. Literature Review 2](#_Toc41549768)

[2.1. Development of OCR 2](#_Toc41549769)

[2.1.1. First Generation OCR System 2](#_Toc41549770)

[2.1.2. Second Generation OCR System 2](#_Toc41549771)

[2.1.3. Third Generation OCR System 2](#_Toc41549772)

[2.1.4. Fourth Generation OCR systems (OCRs today) 3](#_Toc41549773)

[2.2. Working Principles 3](#_Toc41549774)

[2.3. Similar Systems 4](#_Toc41549775)

[2.3.1. Photo Math 4](#_Toc41549776)

[2.3.2. Math Solver 4](#_Toc41549777)

[2.4. Analysis of the review 5](#_Toc41549778)

[2.4.1. Why Tesseract OCR? 5](#_Toc41549779)

[2.4.2. Review of similar systems 5](#_Toc41549780)

[3. Development 6](#_Toc41549781)

[3.1. Development Methodology 6](#_Toc41549782)

[3.2. Increment Model 7](#_Toc41549783)

[3.2.1. Phases in Incremental Model 8](#_Toc41549784)

[3.2.2 Why Incremental Model? 9](#_Toc41549785)

[3.2.3 Techniques and Tools 9](#_Toc41549786)

[4. Project Planning 10](#_Toc41549787)

[4.1. Work Breakdown Structure 10](#_Toc41549788)

[4.2. Gantt chart 11](#_Toc41549789)

[5. Probable Issues During the project 12](#_Toc41549790)

[5.1. Technical Issues 12](#_Toc41549791)

[5.2. Ethical Issues 13](#_Toc41549792)

[6. Conclusion and Recommendation 13](#_Toc41549793)

[6.1. Conclusion 13](#_Toc41549794)

[6.2. Recommendation 13](#_Toc41549795)

[References 14](#_Toc41549796)

[Wireframes 15](#_Toc41549797)

Table of Figures:

[Figure 1: Architecture of Tesseract OCR (Patel & Patel, 2012) 3](file:///D:\ADipIT06%20-%20Research%20&%20Development%20Skills%20in%20Computing\CW\CW1\CW.docx#_Toc41549830)

[Figure 2: PhotoMath Application 4](file:///D:\ADipIT06%20-%20Research%20&%20Development%20Skills%20in%20Computing\CW\CW1\CW.docx#_Toc41549831)

[Figure 3: Math Solver Application 4](file:///D:\ADipIT06%20-%20Research%20&%20Development%20Skills%20in%20Computing\CW\CW1\CW.docx#_Toc41549832)

[Figure 4: Incremental Model (javaTpoint, 2018) 7](#_Toc41549833)

[Figure 5: Incremental Model Visualization (GeeksforGeeks, 2020) 7](#_Toc41549834)

[Figure 6: Work Breakdown Structure 10](file:///D:\ADipIT06%20-%20Research%20&%20Development%20Skills%20in%20Computing\CW\CW1\CW.docx#_Toc41549835)

[Figure 7: Gantt chart 11](file:///D:\ADipIT06%20-%20Research%20&%20Development%20Skills%20in%20Computing\CW\CW1\CW.docx#_Toc41549836)

[Figure 12: Calculator Screen 15](file:///D:\ADipIT06%20-%20Research%20&%20Development%20Skills%20in%20Computing\CW\CW1\CW.docx#_Toc41549837)

[Figure 8: History Screen 15](file:///D:\ADipIT06%20-%20Research%20&%20Development%20Skills%20in%20Computing\CW\CW1\CW.docx#_Toc41549838)

[Figure 9: Navigation Screen 15](file:///D:\ADipIT06%20-%20Research%20&%20Development%20Skills%20in%20Computing\CW\CW1\CW.docx#_Toc41549839)

[Figure 10: Solution Screen 15](file:///D:\ADipIT06%20-%20Research%20&%20Development%20Skills%20in%20Computing\CW\CW1\CW.docx#_Toc41549840)

[Figure 11: Main Screen 15](file:///D:\ADipIT06%20-%20Research%20&%20Development%20Skills%20in%20Computing\CW\CW1\CW.docx#_Toc41549841)

Abstract

This paper explains about what *Optical Character Recognition* technologyis, and how it works. Moreover this paper gives a little gist about the working principle of OCR and how this technology can be used to develop a system like *Smart Calculator*. With the purpose of developing such system, this paper also describes the development methodology like *Incremental Model*.

# Introduction

## Introduction to OCR

Optical Character Recognition is a technology that allows machine to recognize the text whether it is scanned or printed text images or handwritten text. The machine can do further processing on the data extracted from that text. It can be considered same as the combination of human eye and mind. An eye can see the text from some source but mind is the one that actually processes and interprets that text.

OCR system is made up of combination of both hardware and software. Hardware such as, optical scanner or some specialized circuit board is used to read or extract text. And software does the advance processing.

The most common use of OCR is to convert hard copy documents into softcopy files such as PDFs. This will make easier to edit the document. There are many other applications of OCR, such as: image text extraction, extracting texts from scanned documents, License plate recognition and answer paper checker. (Patel, et al., 2012)

The main concept of this project is to implement OCR in an android based calculator application. The objective of this application is to scan handwritten or printed numbers and perform mathematical calculations on it.

## Problem Domain

In this modern era of technology, people don’t want to waste time in less important tasks like performing mathematical calculation in traditional ways which are generally time consuming. For that they have calculators, but they still need manual input from the users. They need to follow some sets of guidelines and go through each processes to get correct answer. It would be much more time saving if user don’t have to manually enter the data for calculation. If the calculator gets the data with just one click, it will definitely be more efficient and time saving.

Besides, while checking the answer paper, teachers need to solve the question first by themselves and then only can correct the paper. Students are also in dilemma after solving a problem whether their answer was correct or not. It would be great help for them if they knew if their answer was correct.

## This project as a solution

The main objective of this project is to develop an app that can perform mathematical calculations without even having to type to give input. It scans for a mathematical problem mentioned in any paper and gives back the answer. Since it is a mobile application, user can use their camera to scan the problem. The app then extracts the problem from paper using OCR and perform operations on it and return the result. Teachers and students can check whether their answers were right or not just by a click. Since this app knows all the mathematical rules and principles, users don’t need to bother about remembering rules like BODMAS, which they had to keep in mind if they were to perform calculation manually in traditional calculators.

# Literature Review

## Development of OCR

### First Generation OCR System

Character Recognition was first originated in early 1870s with the invention of retina scanner. The first generation OCR appeared only in the beginning of 1960s with the development of the digital computers. This generation machines could read symbols specially designed for them. The first commercialized OCR of this generation was *IBM 1418*, which was designed to read special IBM font, *407*. The recognition method was template matching, which compares the character image with a library of prototype images for each character of each font. (Shodh Ganga, 2015)

### Second Generation OCR System

This generation machines were able to recognize machine printed as well as hand written characters. But the character set was limited to numerals and a few letters and symbols. Such machines appeared in between middle of 1960s to early 1970s. (Shodh Ganga, 2015)

### Third Generation OCR System

This generation OCR systems mainly focused on overcoming the challenges like poor document quality, large printed and hand written character sets. Low cost and high performance were also important concerns. (Shodh Ganga, 2015)

### Fourth Generation OCR systems (OCRs today)

This generation system focuses on complex documents which contain texts, graphics, tables, mathematical symbols, unconstrained handwritten characters, low-quality noisy documents and many more. (Shodh Ganga, 2015)

In this project we are going to use OCR to recognize handwritten or machine printed numerals and mathematical symbols. We can then perform mathematical calculations on retrieved data. There already exists some products that use OCR to do mathematical calculations. Some of them are presented below in the section *2.3.*

### Working Principles

The objective of OCR is to extract the text and convert it into editable form. For that, a document is first scanned using an optical scanner which produces an image form of the document. Now this text image is converted into editable character code such as ASCII. The basic working principle of OCR can be show as following figure.

Scanned Document

Pre-Processing

Recognition

Editable Text

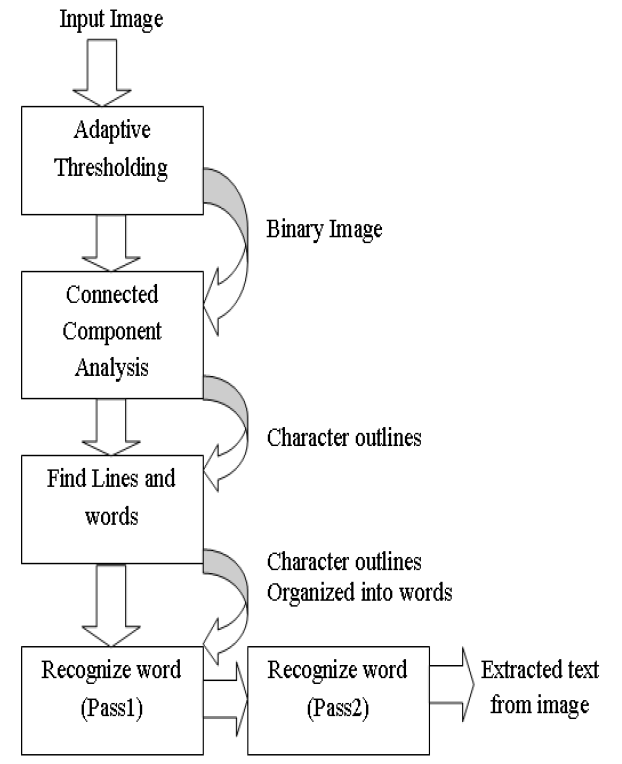
Tesseract OCR is used to get image input from documents. It works on step by step manner as per the diagram shown below.

Figure 1: Architecture of Tesseract OCR (Patel & Patel, 2012)

## Similar Systems

### Photo Math

Photomath is one of the best math problem solver application. It uses phone’s camera to capture a picture of the math problems. Then the picture is scanned by the application. The application uses advanced OCR technology in order to recognize both, handwritten and printed characters. The recognized characters are then processed through Photomath’s own algorithm that examines every character and determines the formula for the scanned problem. Finally, a problem solving algorithm is applied to the formula and the solution is provided with every solving steps. (Photomath, 2020)

Figure 2: PhotoMath Application

### Math Solver

Microsoft Math Solver can not only solve simple mathematical calculations, it can solve various math problems like quadratic equations, calculus and statistics. The application can also show graphs of the equations.

We can either type our problem query using a scientific calculator in the application or draw it on the phone’s screen. But most importantly we can just use our phone’s camera to scan the problem on our books or on the copy written by us. (Microsoft, 2019)

Figure 3: Math Solver Application

## Analysis of the review

### Why Tesseract OCR?

Some of the reasons to use Tesseract OCR are as follows:

* It is platform independent
* Supports multiple languages (Google Open Source, 2020)
* High accuracy
* Open source
* Ease of access and use
* Tesseract has the font accuracy in the range of 85-90%

### Review of similar systems

All of the above mentioned systems have their separate math content team. Because of this there is solution to every math problem from arithmetic to calculus. These systems read and solve mathematical problems by just using the camera of mobile phones. The most astonishing feature of these systems is that they provide step-by-step solutions too. We can even choose multiple explanation methods for same problems. Moreover, they also provide animated calculation steps.

The only bad aspect or the limitation of these systems is that they support only English language. They can only perform calculations on English numerals. This is the aspect where my project is going to work on. My smart calculator will be able to perform calculations on Nepali numerals too.

# Development

## Development Methodology

Software development methodology is a series of processes or tasks used in software development in order to improve quality of a product. It can also be known as system development life cycle. Following certain methodology allows us to document policies, procedures and processes making development process easier.

As most of the development works happens in a team, so there needs to be a good communication between developers. This is where a methodology followed comes in handy. It sets norms between a group of people working on a project about how they are going to pass information between each-others. Whether that be documentation, discussion, or diagrams on paper. (Gianpaul Rachiele, 2018)

Some of the Software Development Methodologies are:

* Agile
* Waterfall
* Incremental
* Extreme Programming
* Rapid Application Development

Among these, Incremental model is best suited for my project and is defined below in section *3.2.*

## Increment Model

Incremental Model is one of the methodologies of software development where requirements are divided into multiple standalone modules. Each of this modules go through phases involved in this methodology.



Figure 4: Incremental Model (javaTpoint, 2018)

In this model a simple working system with basic features is built and delivered to client at first. Then the other features are added in many successive iterations. Due to this reason, this model is also known as Successive Version Model. (GeeksforGeeks, 2020)

In the below figure, A, B, C are Software Products that are incrementally developed and delivered.

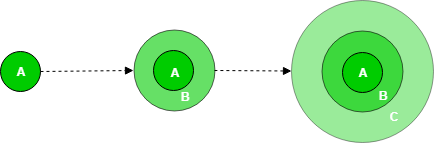


Figure 5: Incremental Model Visualization (GeeksforGeeks, 2020)

### Phases in Incremental Model

1. **Requirement Analysis**

This is the most important phase of software development. The quality of the software depends on how well the requirements are gathered. Requirements are analyzed and refined so that no requirements are ambiguous. This helps in understanding the project more clearly. In this phase, interaction with client plays a vital role to clarify confusions and prioritize the requirements. (javaTpoint, 2018)

1. **System Design**

System Designing, also known as system modeling is the process of developing abstract models of a system. In general it is a process of converting textual requirements gathered from previous phase into graphical representation. It helps the developer to understand the functionality of the system. This phase also decides how the system will operate in terms of the hardware, software and network infrastructure. (Dennis, et al., 2019)

Activities carried out in this phase:

* UI/UX Modeling (User Interface Design)
* Context Modeling
* Data Modeling (Database Design)
* Process Modeling (Flow Design)
* Architecture Modeling (Structure Design)

1. **Implementation (Development)**

Implementation phase is the one in which developers actually code the system. In this phase developers try to replicate system from design phase in an actual system. This is the longest phase of SDLC. The system is first developed in small programs called units, which are then integrated in the next iteration. After developing each units they are tested for its functionality in the next phase of the methodology. (Tutorialpoints, 2020)

1. **Testing**

After the completion of development phase, each unit of the system is checked for whether it met the requirements or not. Different approaches like verification and validation is used to test the system. In technical terms, testing is the process of executing a program with an intent to find an error. It is believed that, complete test of a system is impossible as the domain of possible inputs is too large. (Tutorialpoints, 2020)

In the commercial production, software is deployed after successful testing. Regular maintenance is carried out and software keeps on getting better. These processes can also be considered as phases of software development process in Incremental Model.

### Why Incremental Model?

* Parallel development can be done
* Supports change of requirements
* Testing and debugging is easier because of small iterations
* More flexible
* Easy to manage risks (Ghahrai, 2016)

### Techniques and Tools

Software Development techniques includes preparing a plan of upcoming work, estimating task to perform, allocating resources and monitoring quality and deadlines. There are various techniques available for various development methodologies. Some of them are:

* Strategic Planning Technique
* Structured Programming Technique
* Object Oriented Analysis and Design Technique
* Software Testing Technique (Spring Digital, 2019)

When applying any techniques to the project we need to use specific tools for successfully implementing techniques. We needs tools to plan, design and develop the system. Some of the tools that come handy during software development are as follows:

* Modeling tools for designing (*Star UML, Adobe XD, Corel Draw*)
* Word processor for report (*Microsoft Office*)
* Integrated Development Environment (*Java*)
* Database Management tools (*MySQL, Oracle*)
* Code Generator (*Visual Studio*)

# Project Planning

## Work Breakdown Structure

Figure 6: Work Breakdown Structure

### Gantt chart

Figure 7: Gantt chart

# **Probable Issues During the project**

## Technical Issues

* Hardware failure

One of the major issue that may occur during the project is hardware failure. No matter how careful we are, hardware like hard drive may crash at any time. Likewise problem in different peripherals like mouse and keyboard may occur.

To mitigate this problem project will be stored in cloud storage as well as eternal hardware. Extra pair of peripherals will also be available to use during the project.

* Version Controlling

As I am a beginner programmer, I have no experience to manage the flow of programming. While adding new features I might mess up the previous code and project may fail.

To overcome this issue I will use version controlling tools like “Github”. This way I can fall back to the previous running code.

* Lack of Dataset to train the system

Artificial Intelligence works on the basis of data provided to it. Similarly OCR will only recognize the data that is already fed to it. So we need huge amount of dataset to teach the system which may not be available free to use. And as a student I might be able to purchase those datasets.

To mitigate this problem we can ask for the data set that is already used by the senior students who had done similar type of projects.

* Incompatible SDLC methodology

Due to the insufficient experience in using different software development methodologies, the methodology chosen might not be compatible.

To mitigate this problem supervisor plays vital role. His guidance will help to choose the best methodology before starting the project. Moreover internship in similar project related companies will also help to build some experiences.

## Ethical Issues

Since this project is focused in building an app that can do quick mathematical calculations. So the questions like, student will be lazy and they might cheat during evaluation may arise.

To mitigate this issue we have to make the purpose of this application clear to every user that this system’s aim is only to make calculations quicker and not promote cheating.

# Conclusion and Recommendation

## Conclusion

In conclusion, **Smart Calculator** is a mobile application that uses emerging Artificial Intelligence technology known as “OCR”. This application can do mathematical calculations just by scanning the handwritten or printed problems on paper. It can also be used as normal and traditional calculator as well. The main purpose of building this application is to make calculations a little faster and easier to teachers and students. There are already many similar applications but they are confined within English language. So this application tries to break this barrier a little bit and allows users to perform calculations even in Nepali language or numerals.

## Recommendation

No any system in the world is perfect. Neither is this one. This is a simple system that can do simple calculations. There are lots of rooms to improve. This system can do basic calculations like addition, subtraction, multiplication and division by scanning the problem. We can improve it to solve the algebraic equations, quadratic equations, calculus as well as geometric problems.

We can also work on the area to increase the accuracy of character recognition. As mentioned in section *2.4.2* most of the OCR systems are based on some of the widely spoken languages. One of the future task can be to conduct research on languages other than widely spoken and implement in this application.

# References

Dennis, A., Wixom, B. H. & Roth, M. R., 2019. Design. In: *Systems Analysis and Design.* 7 ed. s.l.:John Wiley and Sons, Inc., p. 385.

GeeksforGeeks, 2020. *GeeksfoGeeks.* [Online]   
Available at: https://www.geeksforgeeks.org/software-engineering-incremental-process-model/  
[Accessed 4 May 2020].

Ghahrai, A., 2016. *DevQA.* [Online]   
Available at: https://devqa.io/software-development-methodologies/  
[Accessed 15 May 2020].

Gianpaul Rachiele, 2018. *Medium.* [Online]   
Available at: https://medium.com/@gianpaul.r/software-development-methodologies-a856883a7630  
[Accessed 5 May 2020].

Google Open Source, 2020. *Tesseract OCR.* [Online]   
Available at: https://opensource.google/projects/tesseract  
[Accessed 15 May 2022].

javaTpoint, 2018. *javaTpoint.* [Online]   
Available at: https://www.javatpoint.com/software-engineering-requirement-analysis  
[Accessed 5 May 2020].

Microsoft, 2019. *Microsoft Math Solver.* [Online]   
Available at: https://math.microsoft.com/en  
[Accessed 20 April 2020].

Patel, C. I., Patel, A. & Patel, D., 2012. Optical Character Recognition by Open source OCR Tool Tesseract. *International Journal of Computer Applications,* 55(10), pp. 50-56.

Patel, C. I. & Patel, D., 2012. Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study. *International Journal of Computer Applications,* 55(10), pp. 50-56.

Photomath, 2020. *Photomath.* [Online]   
Available at: https://photomath.net/en/help/how-does-photomath-work  
[Accessed 20 April 2020].

Shodh Ganga, 2015. *Shodhganga:a reservoir of Indian theses.* [Online]   
Available at: http://14.139.13.47:8080/jspui/handle/10603/130552  
[Accessed 20 April 2020].

Spring Digital, 2019. *Spring Digital.* [Online]   
Available at: https://www.springdigital.com.au/software-apps/tools-and-techniques-for-software-development/  
[Accessed 16 May 2020].

Tutorialpoints, 2020. *SDLC.* [Online]   
Available at: https://www.tutorialspoint.com/sdlc/sdlc\_quick\_guide.htm  
[Accessed 5 May 2020].

# Wireframes

Figure 8: Calculator Screen

Figure 9: History Screen

Figure 10: Navigation Screen

Figure 11: Solution Screen

Figure 12: Main Screen