

Project and Professionalism

(6CS020)

Smart Calculator using OCR

Student Id : 2039244

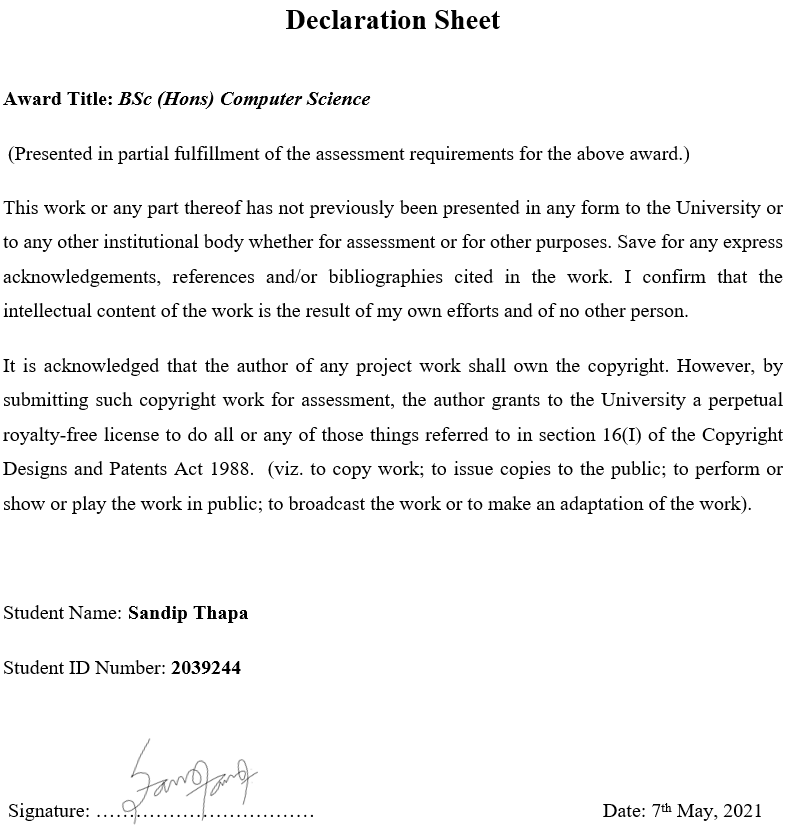
Student Name : Sandip Thapa

Supervisor : Mr. Manish Deuja

Cohort/Batch : 4

Words Count :

Submitted on : 23-09-2020



**Acknowledgement**

I would like to express my deepest appreciation to all those who provided me the possibility to complete this project. I would like to give special gratitude to my project supervisor **Mr. Manish Deuja** and tutors **Mr. Nirmal Thapa** and **Mr. Biraj Dulal**, whose stimulating suggestions and encouragement, helped me to coordinate on my project and writing this report.

Furthermore I would also like to acknowledge my colleagues that helped me wipe my doubts and misunderstanding while doing this project. Last but not the least, many thanks goes to the **Herald College** and **University of Wolverhampton** to provide me with this opportunity to develop a project on my own. Doing this project has taught me very useful and industry related skills. The professionalism aspect I learned during this project is surely going to help to kick start my career.

**Abstract**

This report is prepared in partial fulfillment of the assessment requirements presented by university of Wolverhampton and Herald College for the award of the BSc (Hons) Computer Science. The report reflects the works carried out during the development of the project. This report presents the actions carried out and technologies required for it to develop an android based OCR calculator system. It provides the information about the techniques and methodologies used during the process. Moreover it provides the theoretical knowledge about those techniques and methodologies. All the tools and skills required to pull this project out are mentioned as well. This report can also be presented as the documentation of the product developed. The product developed is just a smaller module that can be integrated into any larger role based system in future.

Contents

[1. Introduction 1](#_Toc71139509)

[1.1. Introduction to OCR 1](#_Toc71139510)

[1.2. Problem Domain 1](#_Toc71139511)

[1.3. This project as solution 2](#_Toc71139512)

[2. Aim 3](#_Toc71139513)

[3. Objectives 3](#_Toc71139514)

[4. Academic Question 3](#_Toc71139515)

[5. Scope and Limitation of the project 4](#_Toc71139516)

[6. AI aspect of the project 5](#_Toc71139517)

[6.1. Computer Vision 5](#_Toc71139518)

[6.2. History of computer vision 5](#_Toc71139519)

[6.3. How computer vision works 5](#_Toc71139520)

[6.4. Applications of Computer Vision 6](#_Toc71139521)

[6.5. Basis of Machine learning 6](#_Toc71139522)

[6.5.1. Supervised Learning 6](#_Toc71139523)

[6.5.2. Unsupervised Learning 6](#_Toc71139524)

[6.5.3. Reinforcement Learning 7](#_Toc71139525)

[7. Algorithm used in the project 8](#_Toc71139526)

[7.1. Convolutional Neural Network 8](#_Toc71139527)

[7.1.1. Input Layer 9](#_Toc71139528)

[7.1.2. Convolution Layer 9](#_Toc71139529)

[7.1.3. Pooling Layer 11](#_Toc71139530)

[7.1.4. Fully Connected Layer 12](#_Toc71139531)

[7.2. Math behind CNN 13](#_Toc71139532)

[7.2.1. Tensors and Vectorization 13](#_Toc71139533)

[7.2.2. Calculus and the chain rule 13](#_Toc71139534)

[7.2.3. The forward run 14](#_Toc71139535)

[7.2.4. Stochastic gradient descent 14](#_Toc71139536)

[7.2.5. Back propagation 15](#_Toc71139537)

[7.2.6. Activation Functions 15](#_Toc71139538)

[8. Literature Review 17](#_Toc71139539)

[8.1. Development of OCR 17](#_Toc71139540)

[8.1.1. First Generation OCR System 17](#_Toc71139541)

[8.1.2. Second Generation OCR System 17](#_Toc71139542)

[8.1.3. Third Generation OCR System 17](#_Toc71139543)

[8.1.4. Fourth Generation OCR System 17](#_Toc71139544)

[8.2. Working Principles 18](#_Toc71139545)

[8.3. Handwritten Character Recognition 19](#_Toc71139546)

[8.4. Similar Systems 20](#_Toc71139547)

[8.4.1. Photo Math 20](#_Toc71139548)

[8.4.2. Math Solver 20](#_Toc71139549)

[8.4.3. Mathway 21](#_Toc71139550)

[8.5. Review of Similar Systems 21](#_Toc71139551)

[8.6. Research paper on Nepali script 22](#_Toc71139552)

[8.7. Why Tesseract OCR? 23](#_Toc71139553)

[9. Artefact 24](#_Toc71139554)

[9.1. FDD 24](#_Toc71139555)

[9.2. SRS Legend 24](#_Toc71139556)

[9.3. SRS Table 25](#_Toc71139557)

[9.4. User Stories 26](#_Toc71139558)

[9.5. System Modeling 29](#_Toc71139559)

[9.5.1. Context Modeling 29](#_Toc71139560)

[9.5.2. Structural Modeling 31](#_Toc71139561)

[9.5.3. Process Modeling 32](#_Toc71139562)

[9.5.4. UI Model 34](#_Toc71139563)

[10. Full details of Artefact 35](#_Toc71139564)

[10.1. Project Development Methodology 35](#_Toc71139565)

[10.2. Preferred Methodology 36](#_Toc71139566)

[10.3. Techniques and Tools 37](#_Toc71139567)

[11. Model Development 38](#_Toc71139568)

[11.1. Data Collection and training 38](#_Toc71139569)

[11.2. Model Comparison and Performance Analysis 39](#_Toc71139570)

[11.2.1. Model 1 39](#_Toc71139571)

[11.2.2. Model 2 41](#_Toc71139572)

[11.2.3. Model 3 43](#_Toc71139573)

[11.2.4. Model 4 45](#_Toc71139574)

[12. Testing 47](#_Toc71139575)

[12.1. Basic calculation test 47](#_Toc71139576)

[12.2. Scientific calculation test 47](#_Toc71139577)

[12.3. Scanner functionality test 48](#_Toc71139578)

[12.4. Scanning options test 48](#_Toc71139579)

[12.5. Cropper functionality test 49](#_Toc71139580)

[12.6. Result display testing 49](#_Toc71139581)

[12.8. Result editing test 50](#_Toc71139582)

[13. Issues during the project 51](#_Toc71139583)

[14. Conclusion and Recommendation 52](#_Toc71139584)

[14.1. Conclusion 52](#_Toc71139585)

[14.2. Recommendation and further works 52](#_Toc71139586)

[15. Critical Evaluation of the project 53](#_Toc71139587)

[16. Evidence of Project Management 54](#_Toc71139588)

[16.1. Gantt chart 54](#_Toc71139589)

[References 55](#_Toc71139590)

[Appendices 58](#_Toc71139591)

Table of Figures:

[Figure 1: Architecture of CNN 8](#_Toc71139592)

[Figure 2: Convolution Operation 10](#_Toc71139593)

[Figure 3: Max Pooling vs Avg Pooling 11](#_Toc71139594)

[Figure 4: Sigmoid function 16](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139595)

[Figure 5: Tanh function 16](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139596)

[Figure 6: ReLU function 16](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139597)

[Figure 7: Architecture of Tesseract OCR (Patel & Patel, 2012) 18](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139598)

[Figure 8: Segmented Character from input image 19](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139599)

[Figure 9: Downsampled Character 19](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139600)

[Figure 10: PhotoMath Application 20](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139601)

[Figure 11: Math Solver Application 20](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139602)

[Figure 12: Mathway Application 21](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139603)

[Figure 13: Use Case Diagram for Basic Calculation System 29](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139604)

[Figure 14: Use Case Diagram for Image Capturing System 30](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139605)

[Figure 15: Use Case Diagram for Character Recognition System 30](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139606)

[Figure 16: Class Diagram 31](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139607)

[Figure 17: Context Diagram for Image Capturing System 32](#_Toc71139608)

[Figure 18: Context Diagram for Character Recognition System 33](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139609)

[Figure 19: History Screen 34](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139610)

[Figure 20: Calculator Screen 34](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139611)

[Figure 21: Solution Screen 34](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139612)

[Figure 22: Navigation Screen 34](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139613)

[Figure 23: Main Screen 34](file:///D:\FYP\Project%20and%20Professionalism\Report\FYP_Draft.docx#_Toc71139614)

[Figure 24: Incremental Model Visualization (EduCba, 2020) 36](#_Toc71139615)

[Figure 25: Classification report of first model 39](#_Toc71139616)

[Figure 26: Classification report of second model 41](#_Toc71139617)

[Figure 27: Classification report of third model 43](#_Toc71139618)

[Figure 28: Classification report of fourth model 46](#_Toc71139619)

[Figure 29: Gantt chart 54](#_Toc71139620)

# 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 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.

# Aim

This project is based on one and only aim to develop an AI enabled application that can perform mathematical calculations by just scanning the arithmetic equation. The equation may be in digital form or even handwritten form.

# Objectives

To fulfill the aims mentioned above the project has set multiple objectives. Some of them are as follows:

* To make numerals recognition as accurate as possible.
* To make handwritten digit recognition better
* Make use of mobile phone’s camera as scanner to scan arithmetic equations.

# Academic Question

The project is based on the learning outcomes of academic career. So, it tries to cover some academic questions like:

* How can characters be extracted from image sources?
* Which algorithm is most suitable for character extraction?
* How can model trainings be done with datasets?
* How to choose better model among all?

The most challenging part of the project us to extract the numbers from the image. We can make use of convolutional neural network or use Tesseract OCR.

# Scope and Limitation of the project

Scope:

* Digital Character Recognition

The project aims to develop an android application that would be able to recognize characters from an image with digital characters.

* Handwritten Character Recognition

The developed application will not just be able to recognize just digitally printed characters but will also recognize handwritten characters. Users can write their characters on a sheet of paper and the application will be able to recognize them.

* Manual and Scientific Calculator

As being a calculator application, it is necessary to have manual calculator in the system as well. For that, the application has included manual calculator which will be able to perform basic arithmetic to scientific calculations like trigonometric and logarithmic values.

* Arithmetic equation

The application will not only be able to calculate two variables at a time but we can also calculate longer arithmetic equations like *2+6/3\*8-2.*

Out of Scope/limitations:

* Multi language numerals not supported

The application is developed based on English language numerals. But there are different numeral systems in every language. For example Nepalese language has its own numeral system. The first limitation of this application is it does not support such multiple numeral systems.

* Algebraic equations not supported

Although the application can perform arithmetic calculation, it cannot perform algebraic equations to find the values of variables.

* Character not supported

Due to the ease of model training, it is trained with slash character instead of division. So users have to use slash symbol for division.

# AI aspect of the project

## Computer Vision

Computer vision is one of the field of artificial intelligence that works on how computers can gain high-level knowledge from digital images or videos. In other words, computer vision is a field that enables computers to see, identify and work on images in the same way as we human do and even surpass human ability in many cases (Huang, 1996). In computer vision, we train machines to interpret and understand the visual world by providing computers human-like perception capabilities (Sebe, et al., 2005). For that, we use deep learning models on digital images and videos due to which machines can correctly identify and classify objects.

## ­­­­­History of computer vision

Studies in computer vision started only in the 1950s. They made use of some of the first neural networks to detect the edges of an object and to sort simple objects into categories. It was only in the 1970s, the first commercial use of computer vision was done. The system interpreted digital or handwritten text using Optical Character Recognition. As the internet started to mature from the 1990s, large set of images were available online for analysis. These growing datasets helped to train the models and make it possible for machines to identify specific objects from photos and videos.

Today, mobile technology with built-in camera, cheap and powerful computers and new algorithms like convolutional neural network have converged to bring revolution in computer vision. The effect of these advancements have increased the accuracy rates for object detection and classification using computer vision have gone from 50 percent to 99 percent in less than a decade. (Ranjay, 2017)

## How computer vision works

The very first step in computer vision is feature engineering. As part of this, the computer acquires images and videos and convert them into an array of pixels. Then the model identifies the features such as edge and corner of an object. Before predicting the objects it is necessary that we train the model by feeding thousands of such labeled data. Once after the model analyzes pixels, computer vision uses neural network to predict the content of the image.

## Applications of Computer Vision

* Health Care

In health care sector, computer vision enabled machines can help doctors to detect different symptoms of diseases in MRI scans and X-rays.

* Facial Recognition

Facial Recognition is one of the emerging application of computer vision. From general social networking sites to law enforcement agencies, facial recognition technology is being used to identify the faces. Moreover facial recognition lock system in gadgets are being popular these days.

* Autonomous vehicles

In autonomous vehicles, the cameras attached capture the surroundings. Then the computer vision software process it to find the traffic signs, roads and other vehicles. Based on these observations the vehicle identifies its path and avoid the obstacles. (Kagami, 2010)

## Basis of Machine learning

Computer vision is based on machine learning technique of Artificial Intelligence. There are mainly three types of machine learning. Namely, supervised learning, unsupervised learning and reinforcement learning.

### Supervised Learning

Supervised learning is the machine learning technique in which we train the machine using well labeled data. This means data is already provided with correct answers. This learning technique allows us to collect data and predict an output based on previous experiences.

### Unsupervised Learning

Unsupervised learning is a machine learning technique in which we do not need to train the model ourselves. The model itself works to discover patterns and information on unlabeled data. Clustering is an example of unsupervised data.

### Reinforcement Learning

Reinforcement learning is a machine learning technique about taking a suitable action to maximize the reward in a specific situation. In reinforcement learning there is no answer tagged to the training data set like in the supervised learning. Reinforcement agent itself decides what action to carry out in order to perform the given task. And in the absence of a proper training dataset, the machine is has to learn from the past experiences.

As computer vision is based on training the machine with images and videos, it mostly falls under the category of supervised learning. But the some study has shown that the machine learning strategies in computer vision are supervised, unsupervised as well as semi-supervised. Computer vision uses various machine learning algorithms like, CNN, RNN, K-Means Clustering and LSTM. These different algorithms falls under different learning category. Thus pushing computer vision to fall under multiple category of machine learning. (Khan & Al-Habsi, 2019)

# Algorithm used in the project

Computer vision is a field of deep learning that deals with images on all scales. It allows our machines to understand the content in the image. The Convolutional Neural Network is used as the main architecture or the algorithm behind computer vision which is nothing but the derivative of feed-forward neural networks.

## Convolutional Neural Network

In “machine learning, a convolutional neural network is nothing but a class of deep, feed-forward artificial neural networks. It is mostly used in analyzing visual imagery. CNN is a neural network with convolution operation instead of matrix multiplication in at least one of the layers. Convolutional neural networks are usually composed by a set of layers that can be grouped by their functionalities.

Our computers see images as high dimensional vectors. It would take a huge amount of parameters to characterize the network. To solve this problem convolutional neural networks reduce the number of parameters and adapt the network architecture specifically to vision tasks.”

The whole working mechanism can be described by the following diagram.

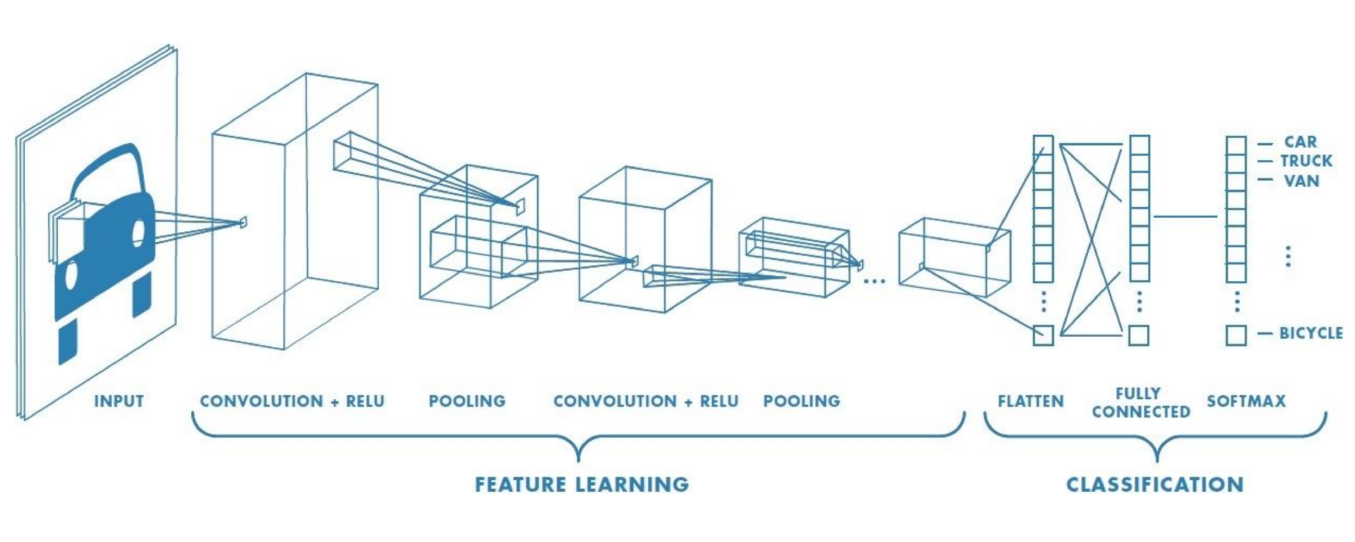


Figure 1: Architecture of CNN

### Input Layer

The input layer holds the input of the whole CNN. In the neural network of image processing it holds the pixel matrix of the image. A CNN usually takes 3 order tensor as input. For example, an image with H rows, W columns and 3 channels(R, G, B color channels). However, higher order tensor inputs can also be handled by CNN.

### Convolution Layer

Convolutional layer is the layer where convolution operation is carried out. The convolution layer is the core building block of CNN. “The parameters consist of multiple sets of learnable filters. Every filter is small in size (width and height), but extends fully through the full depth of the input volume. During the feed-forward pass, we slide each filters across the width and height of the input volume and compute the dot products between the matrix of the filter and the input.” A set of filters in every convolution layer outputs a separate 2-D activation map. After defining the stride and padding we define the convolution product between tensor and a filter. Convolution product is nothing but a sum of element wise product of 2D matrix formed out of pixels and filter.

An image, in general can be mathematically represented as a tensor with the following dimension:

*dim(image)=(nH, nW, nC)*

Where:

*nH : the size of the height*

*nW*: the size of the width

*nC : the number of channels*

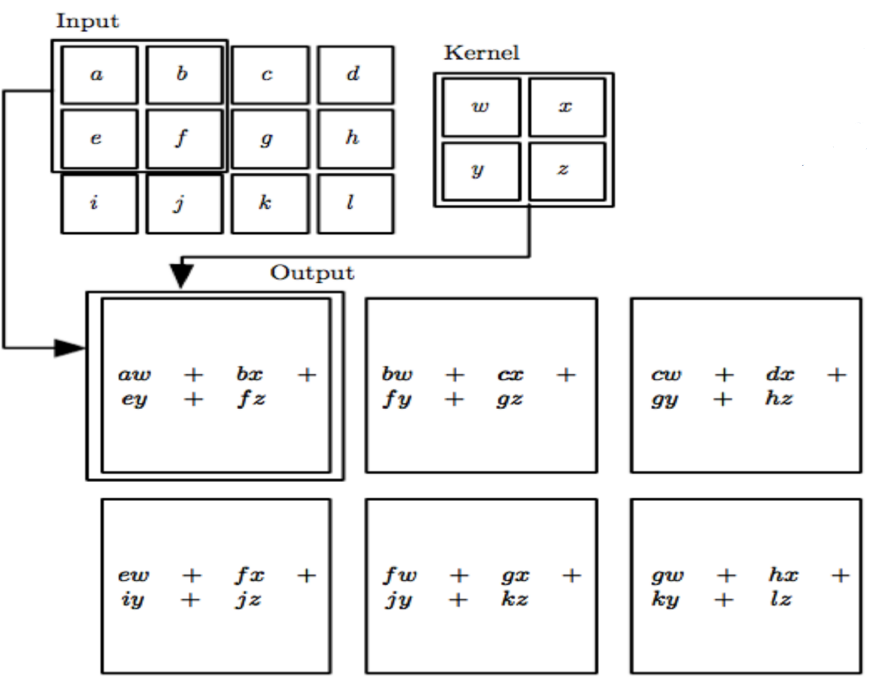


Figure 2: Convolution Operation

Mathematically summarizing the convolution layer:

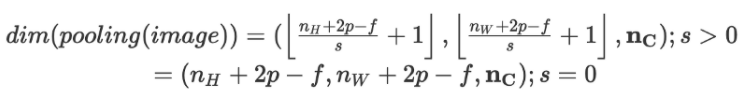
* It accepts the input of size W1 x H1 x D1
* Requires four hyper-parameters:
  + Number of filters, **K**
  + Their spatial extent, **F**
  + The Stride, **S**
  + The amount of padding, **P**
* The output of size W2 x H2 x D2 where:
  + W2 = (W1 – F + 2P) / S + 1
  + H2 = (H1 – F + 2P) / S + 1
  + D2 = K

### Pooling Layer

The use of a pooling layer is to reduce the dimension of the input image. By sampling the convolved feature maps, the useful information of the image is preserved and the redundant data is removed, thus effectively preventing the over fitting problem and also increasing the computation speed.

In this step image is down sampled by summing up the information. The pooling action is carried out through each and every channels and thus it only affects the dimensions and keeps the channels intact.

Given an image, we slide filter through height and width of input volume with no parameters to learn. Then following a certain stride, we apply a function on the selected elements. It can be represented mathematically as:



We generally use max pooling and average pooling in this layer. Below is the illustration of max pooling and average pooling is shown



Figure 3: Max Pooling vs Avg Pooling

### Fully Connected Layer

Fully connected layer is usually the last layer of the network. This layer combines the information gathered from the former layers to achieve the explicit expression of classification. The input of the fully connected layer comes in the form of output from the final pooling layer. That output from pooling layer is first flattened and then fed into the fully connected layer in the form of 1D array. (Wu, 2017)

## Math behind CNN

### Tensors and Vectorization

The foundation of CNN is based on vector and matrix. Geometrically, vector is a directed line segment with magnitude and direction with it. A vector can be viewed as a matrix with one column and n rows.

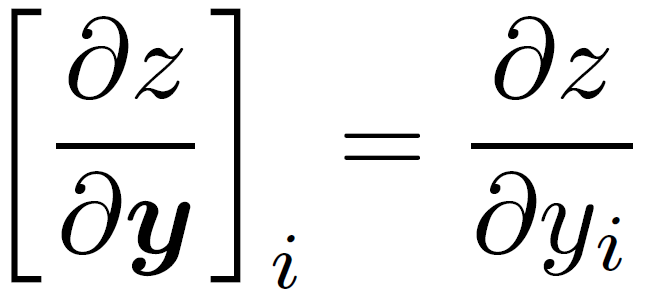
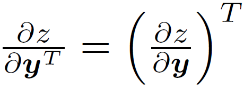
Tensors are nothing but higher order matrices, which is the key component of CNN. A scalar value is zero order tensor. A vector is 1 order tensor. A matrix is order 2 tensor and colored image is an order 3 tensor. An image with *H* rows and *W* columns is a tensor with size *H x W x 3.* It is always beneficial to represent images as a tensor as it preserves the information about colors in the image.

In CNN the input and parameters are represented as tensors. Given a tensor, we can arrange it into a vector. In CNN, the flatten layer converts the multidimensional output matrix into 1D vector. (Briley, 2014)

A = (2, 5, 3, 7)*T* =

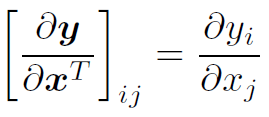
### Calculus and the chain rule

In CNN learning process is dependent to the vector calculus and chain rule. Suppose z is a scalar and y € **R***H* is a vector and z is the function of y, then the partial derivative of z with respect to y is a vector, defined as

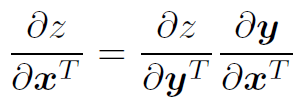


In other words, is a vector with the size same as that of y. Also,

Similarly suppose x € **R**W is another vector and y is the function of x. Then the partial derivative of y with respect to x is



Now, we can see that z is a function of x in a chain-like argument. A function maps x to y and another function maps y to z. From this, we can use chain rule to compute , as



### The forward run

In the prediction task, we only run the CNN model forward direction. Suppose in the image classification task, all the parameters of the model have been learned. Starting from the input x1, we pass it through the processing of first layer and get output x2. Now x2 is passed into the second layer as the input. This process repeats until we get xn which outputs the probabilities of x1 belonging to certain class. Now from those probabilities we output one with the max value using the max function as

arg max *x*i

### Stochastic gradient descent

The main approach of CNN is to minimize the loss as much as possible. For that we optimize the parameters used. Suppose we are training a CNN model for an example x1. In training we run CNN model in both forward and backward directions. From the forward run we get xn to achieve a prediction. That prediction is compared with the target class corresponding to x1. Now with the help of target value and predicted value we can calculate loss Z. The most common way to calculate the loss is to calculate mean squared error.

**MSE** =

*Where,*

*n = total number of inputs*

*= actual value*

*= predicted value*

The loss Z guides us to modify our parameters and stochastic gradient descent is the way to do so. Now backward pass is carried out by calculating the gradient and updating parameters. The process keeps on repeating until the loss is minimum.

Let W be the weight of parameters, then SGD updates the parameters as

W1 = W1 – η

Where, η is a constant called learning rate.

Here the partial derivate measures the rate of increase of z with respect to change in W1. In mathematical optimization this partial derivate is called gradient. We update W1 in backward direction using above equation in order to minimize the loss. This process is known as gradient descent.

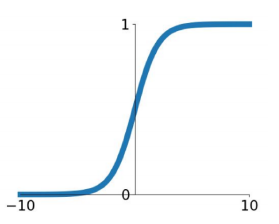
### Back propagation

It is easy to compute the partial derivate of last layer i.e. ith layer as xn is directly connected to Z. The term can be used to update parameters that are backwards. For example in (i-1)th layer xi is the output and how xi should be updated to reduce the loss. So, part that is crucial for error supervision that has propagated from Z to current layer in a layer by layer fashion. In this fashion we can propagate back to the first layer. Then the forward pass begins again and whole process keeps on repeating until and unless loss is minimum.

### Activation Functions

Activation functions are one of the key component of any neural networks. They decide whether to activate a specific neuron or not. This means it can control what input to pass in the model. Activation functions help to output the results in certain range. This help in the process of error back propagation where activation functions help the gradient descent to find the minimum value in the curve. Some of the commonly used activation functions are:

* Sigmoid

Sigmoid is a non-linear activation function that normalizes the output between 0 and 1. It is mostly use in binary classification models. It is also known as logistic function. (Zhang, et al., 2019)

σ(x) =

Figure 4: Sigmoid function

* Tanh

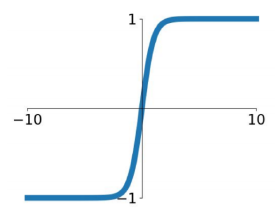
The hyperbolic tangent function is quiet similar activation function like the sigmoid but it normalizes the output between -1 to 1. This function is mostly used in binary classification models.

Figure 5: Tanh function

y = )

* ReLU

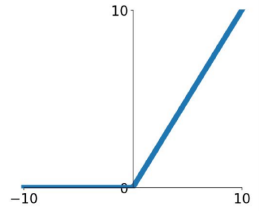
In deep learning models, ReLU is the most used activation function. This function normalizes the output for positive range i.e. zero to maximum value. This function is used to solve the vanishing gradient problem.

Figure 6: ReLU function

y = max(0,x).

# 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 System

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 *4.4.*

## 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.

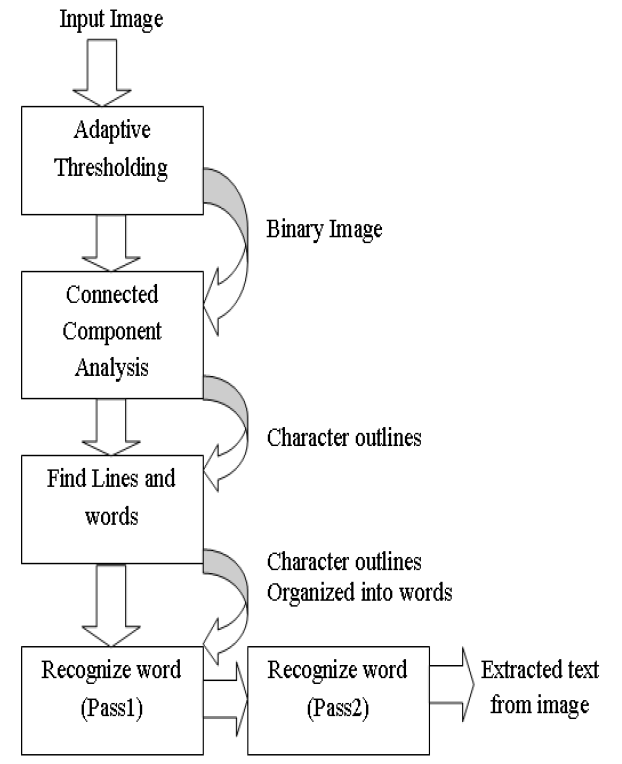


Figure 7: Architecture of Tesseract OCR (Patel & Patel, 2012)

## Handwritten Character Recognition

Tesseract performs well for printed numbers with the detection rate more than 85% for the fonts within its database. But for the handwritten numbers it drops to about 50%. The main reason for this difference in result is due to the variation in the size of numbers that are handwritten and also due to the lack of matching fonts in its database.

To overcome this problem a machine learning algorithm based on Support Vector Machines (SVM) can be applied. This algorithm analyzes data and recognize patterns. This algorithm first converts character images into vector form. After using line segmentation region labels are used to determine the bounding box for each individual characters. A small amount of padding is added to the border, as shown in Figure 2. The segmented character is now down sampled to 32x32 pixels and then divided into 64 4x4 regions. The count in each region is the determined vector value, as shown in Figure 3. This conversion thus results in a 64 dimensional vector for each character image. (Sikka & Wu, 2012)

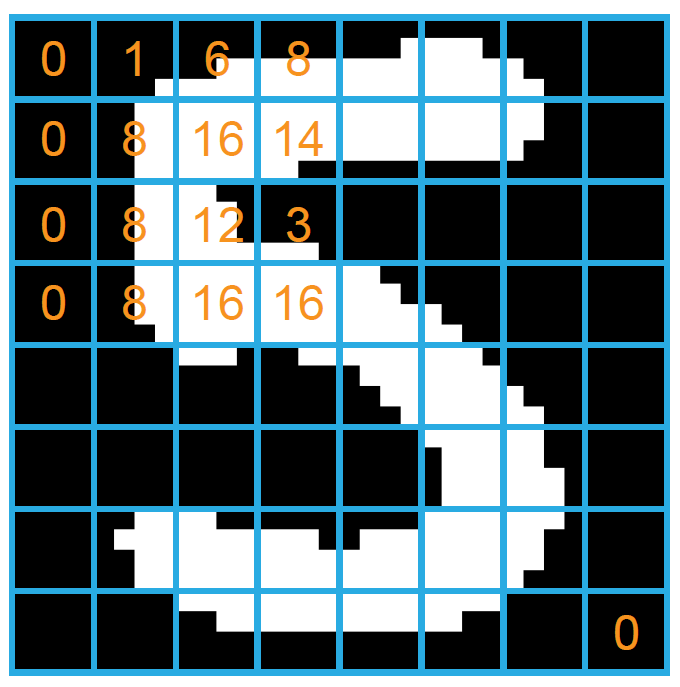


Figure 8: Segmented Character from input image

Figure 9: Downsampled Character

## 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 10: 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 11: Math Solver Application

### Mathway

Mathway is little bit different system than the other two. It works by letting the user to choose the field of mathematics of which the problem is to be solved and then allows user to input the problem by either typing or scanning. The problem is then processed and provides the result in conversational style like the chat bot does. (Mathway, 2020)

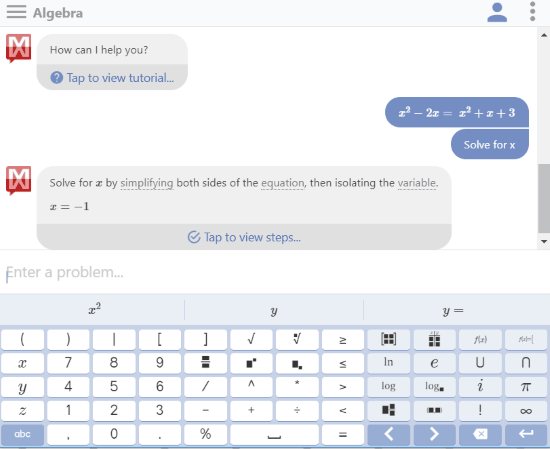
This system implements Lexical analysis in order to solve the problem. It first breaks the problem into tokens. In an expression 1+2, the tokens are 1, + and 2. Then the tokens are fed into the parser, which has the knowledge about relationships between tokens and can call the appropriate function which in this case is add.

Figure 12: Mathway Application

## 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.

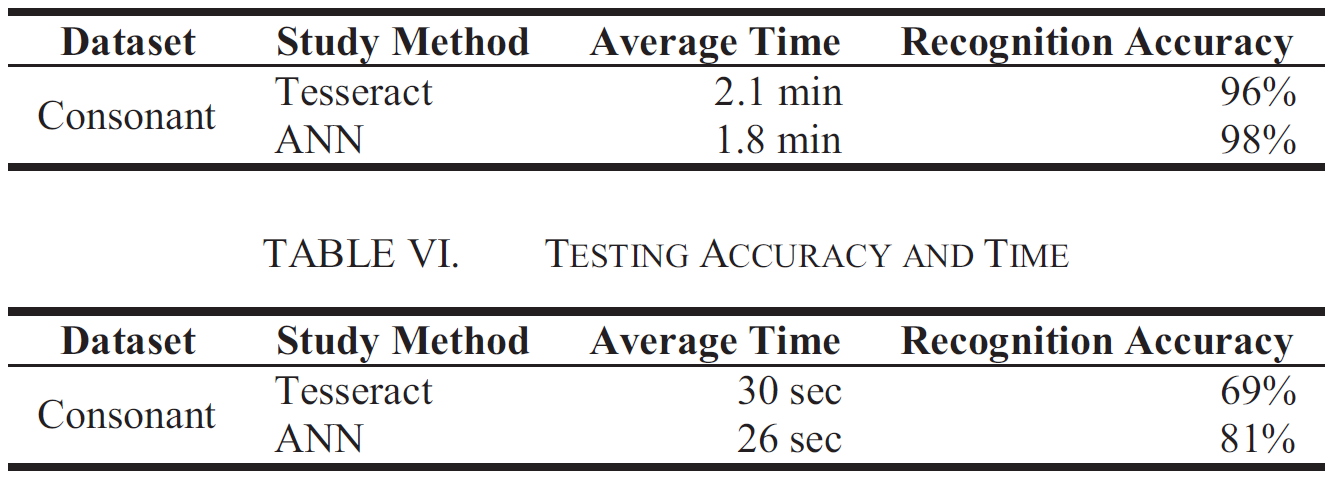
## Research paper on Nepali script

These days there are huge number of OCR system available to use. This is something to be considered as good but among all of those systems, majority of them are based on English script. There are very less amount of such systems with multiple script support. And talking about Nepali script, there are about none such systems.

Students from Tribhuvan University performed research to evaluate the performance of Nepali script using Tesseract and Artificial Neural Network. They collected dataset of 69 Nepali fonts with 2,484 character samples. They typed the character set for each font and printed. It was then scanned in 300 dpi resolution in PDF format.

As the part of pre-processing, noise was to be removed. For that Median filter was used. Then, image segmentation was carried out for the ANN as ANN only takes input in the form of feature vectors. The original image was first segmented into lines, then into words and finally, the words into individual characters. Then the images were rescaled to same size and converted to bitmap image with values 0’s and 1’s.

Now the images were passed into Tesseract and ANN separately to make the classification. Tesseract produced 96% accuracy in training phase and 69% in testing phase whereas ANN obtained 98% accuracy in training phase and 81% accuracy in testing phase. (Prajapati, et al., 2018)



## 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%

# Artefact

## FDD

## SRS Legend

Sub Systems:

MCS : Mathematical Calculation System

CRS : Character Recognition System

HCS : Handwritten Character Recognition System

IPS : Image Processing System

MCS - F - 1.0

Types of Requirements:

F : Functional Requirements

NF : Non-Functional Requirements

UR : Usability Requirements

Numbering

**Legend**

## SRS Table

|  |  |
| --- | --- |
| **Requirement Code** | **Requirement Description** |
| MCSF **1.0** | The system should use the built in ALU to perform arithmetic calculations. |
| MCSNF 1.1 | The system should provide the result in less than 1.5 seconds. |
| MCSNF 1.2 | The system should follow BODMAS rule for calculations. |
| MCSUR 1.1 | The system should have “All Clear” button to reset the entry. |
| MCSUR 1.2 | The system should have “Clear Entry” button to erase the latest entry. |
| MCSUR 1.3 | The system should display result in bigger font than the input. |
|  |  |
| MCSF **2.0** | The system should be able to perform scientific calculations. |
| MCSNF 2.1 | The system should be able to point out the error if any occurs. |
| MCSNF 2.2 | The scientific notations should be placed separately. |
|  |  |
| MCSF **3.0** | The system should keep records of the calculations history. |
| MCSNF 3.1 | The system should display maximum five history records. |
| MCSNF 3.2 | The system should have clear history option. |
| MCSUR 3.1 | The system should have black background with white text color. |
|  |  |
| CRSF **1.0** | The system should recognize numbers as well as mathematical notations. |
| CRSUR 1.1 | The notations should be clear and familiar to the users. |
| CRSF **2.0** | The system should recognize characters with noises as well. |
| CRSNF 2.1 | The system should recognize the characters in less than 2 seconds. |
|  |  |
| HCSF **1.0** | The system should recognize hand written characters. |
| HCSNF 1.1 | The system must have handwritten accuracy over 80%. |
| HCSUR 1.1 |  |
|  |  |
| IPSF **1.0** | The system should be able to capture image to perform calculations. |
| IPSNF 1.1 | The system should use mobile camera to capture the image. |
| IPSNF 1.2 | The system should capture the in 720\*720 resolution. |
| IPSNF 1.3 | After capturing the image there should be crop option. |
| IPSNF 1.4 | There should be save option for image. |
| IPSUR 1.1 | There should be camera icon to switch between standard and scanning calculator. |
|  |  |
| IPSF **2.0** | The system should be able to perform calculations on images from gallery. |
| IPSNF 2.1 | While importing, the images should be shown from latest to old. |
| IPSNF 2.2 | While importing only one image should be selectable. |
| IPSUR 2.1 | The selected image should be faded. |
|  |  |

## User Stories

**Story ID: S001**

**User story:** 8

As a user, I should be able to perform basic calculations manually.

**Priority:** 5

**Dependencies:** NA

**Story points:**

**Acceptance Criteria:**

* Assuming application already installed in the user device.
* Design as per the mock up.
* User input and output both should be displayed at the same time.
* There should be option to switch to scanner screen from the calculator screen.
* In case of format error, message should be prompted saying “Incorrect format”.
* Back button should redirect to home screen.

**Story ID: S002**

As a user, I should be able to perform scientific calculations.

**Priority:** 5

**Dependencies:** S001

**Story points:** 2

**Acceptance Criteria:**

* Design as per mockup.
* User should proceed from home screen to reach calculator screen.

**Story ID: S003**

**User Story:**

As a user, I should be able to perform calculations using scanner.

**Priority:** 10

**Dependencies:** NA

**Story Points:**

**Acceptance Criteria:**

* User must be able to switch to scanner screen from home screen.
* Design as per mock up.

**Story ID: S004**

**User Story:**

As a user, I should be able to use both camera and gallery to scan problems.

**Priority:** 10

**Dependencies:** S003

**Story Points:**

**Acceptance Criteria:**

* On clicking scan button, options of camera and gallery should be prompted.
* On using camera, clicked photo should be stored in the device.

**Story ID: S005**

**User Story:**

As a user, I should be able to crop the captured or imported image.

**Priority:** 10

**Dependencies:** S004

**Story Points:**

**Acceptance Criteria:**

* There should be option to confirm captured image or retake the image.
* There should be an option to crop the confirmed image.

**Story ID: S006**

**User Story:**

As a user, I should be able see the scanned image and output of the image on the same screen followed by the result.

**Priority:** 7

**Dependencies:** S005

**Story Points:**

**Acceptance Criteria:**

* Design as per the mockup.
* Scan result must be editable.
* Calculation error should not be editable.

## System Modeling

### Context Modeling

#### Use Case Diagram

* Use Case Diagram for Basic Calculation System

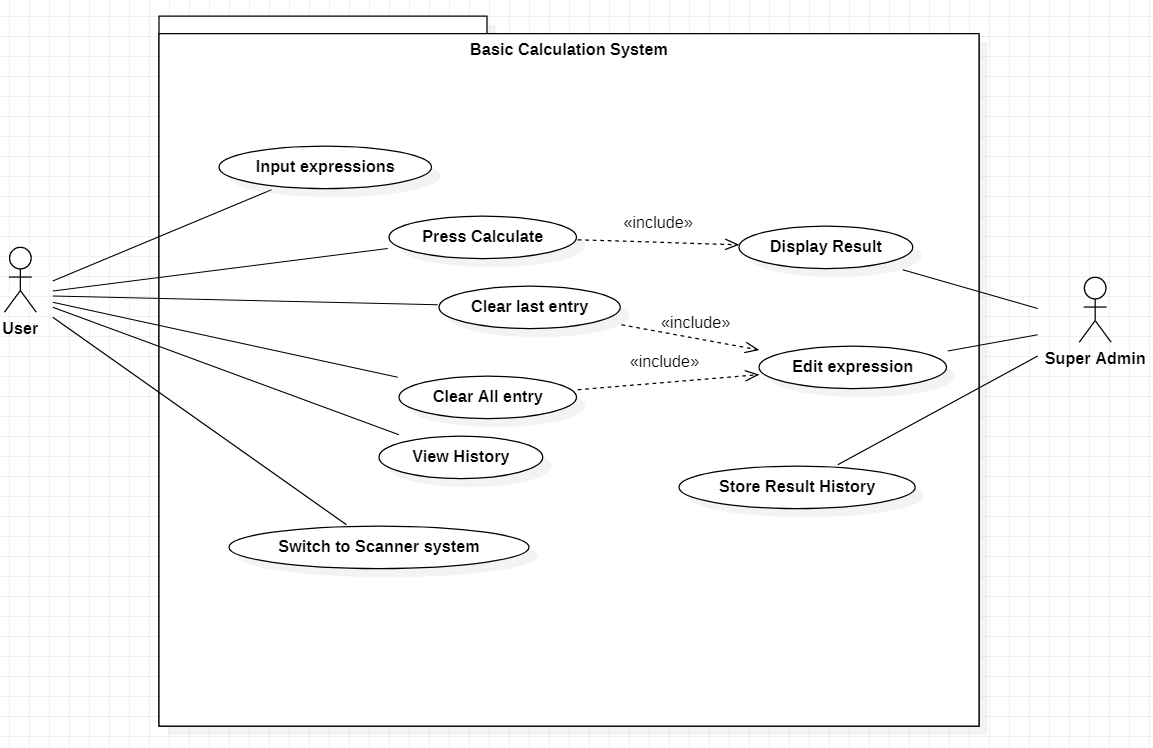


Figure 13: Use Case Diagram for Basic Calculation System

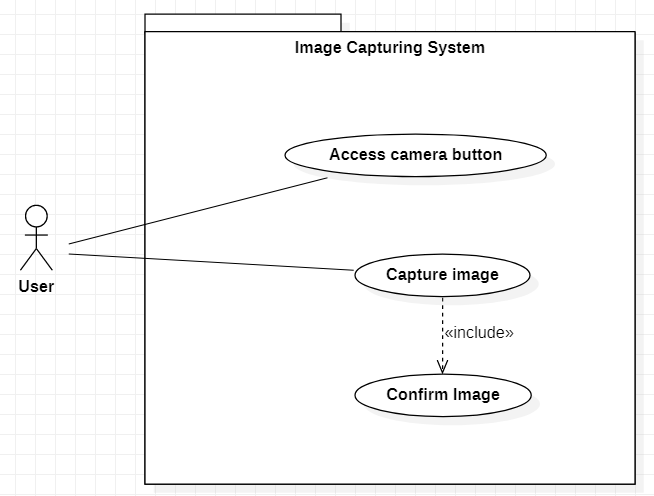
* Use Case Diagram for Image Capturing System

Figure 14: Use Case Diagram for Image Capturing System

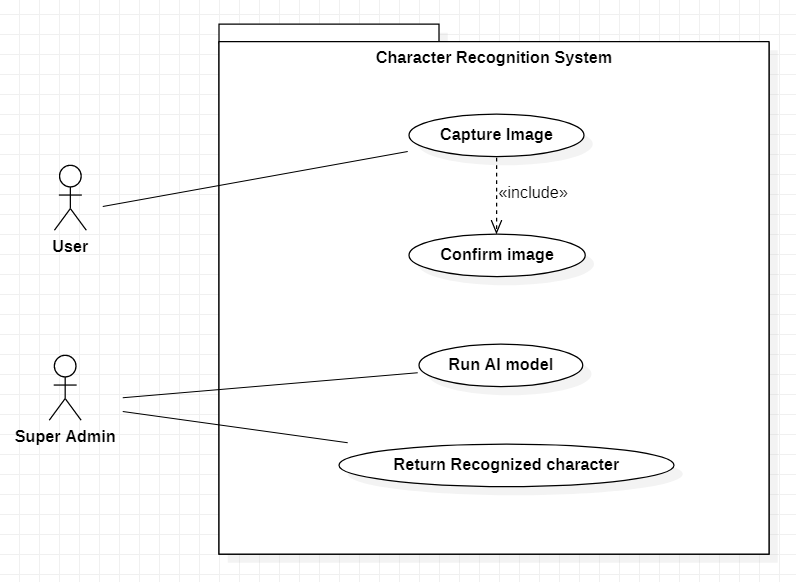
* Use Case Diagram for Character Recognition System

Figure 15: Use Case Diagram for Character Recognition System

### Structural Modeling

#### Class Diagram

Figure 16: Class Diagram

### Process Modeling

#### Context Diagram

* Context Diagram for Character Recognition System

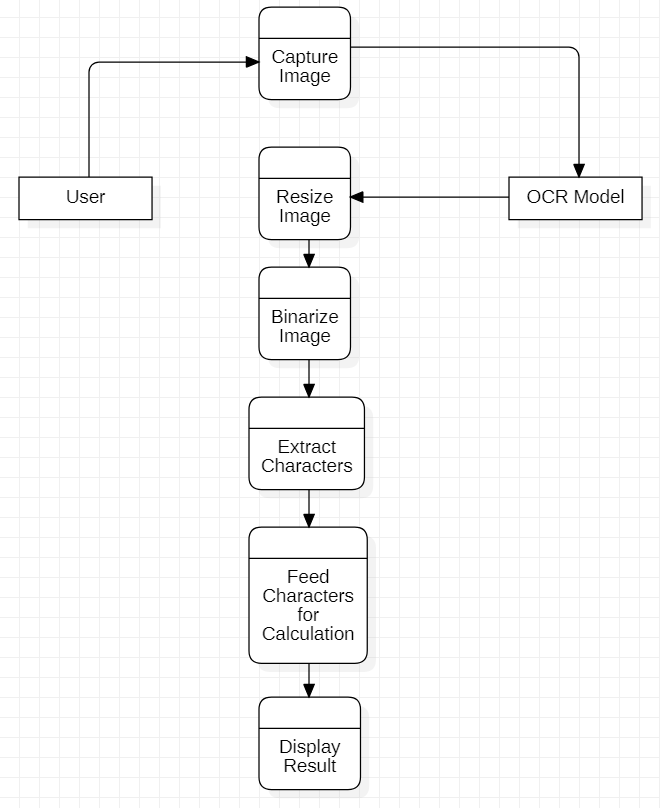


Figure 17: Context Diagram for Image Capturing System

* Context Diagram for Character Recognition System

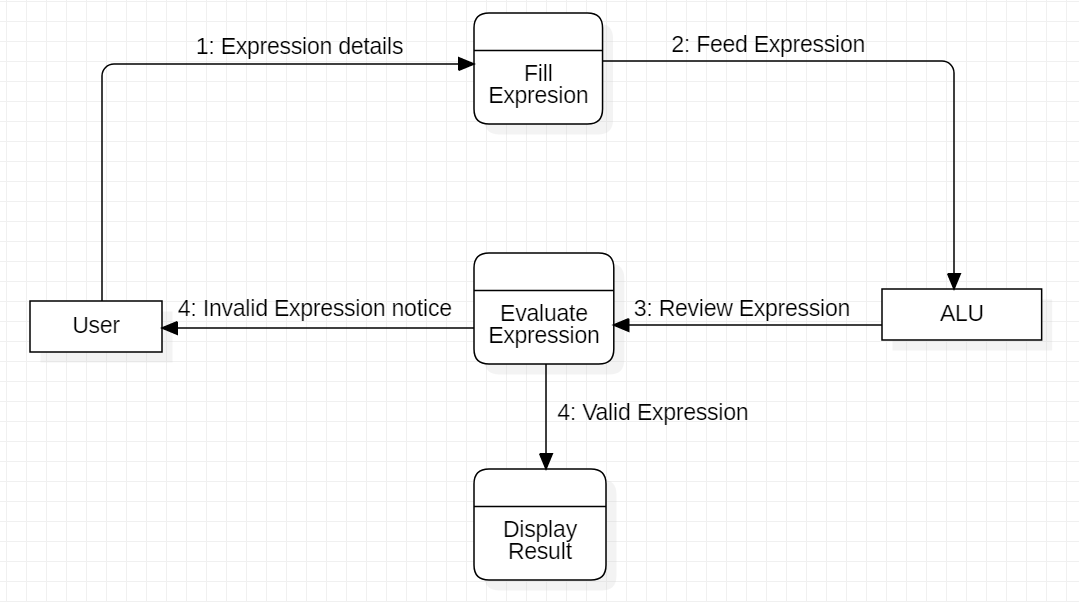


Figure 18: Context Diagram for Character Recognition System

### UI Model

#### Wireframe

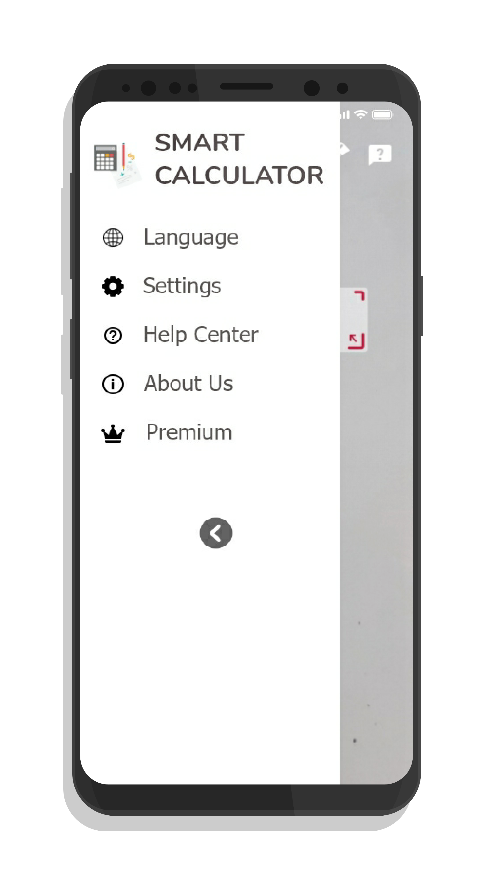
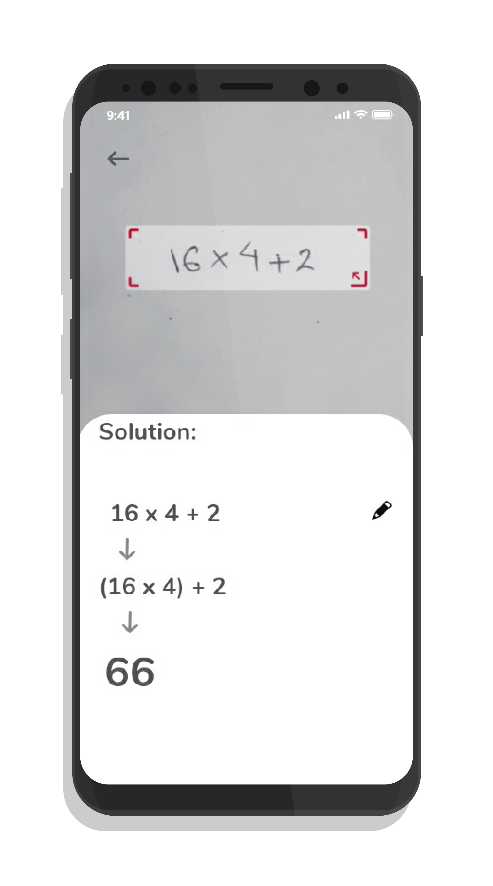
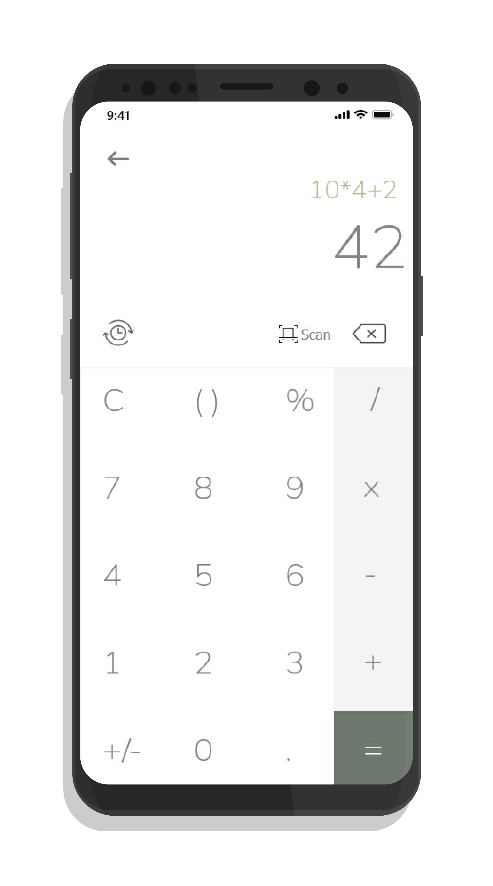
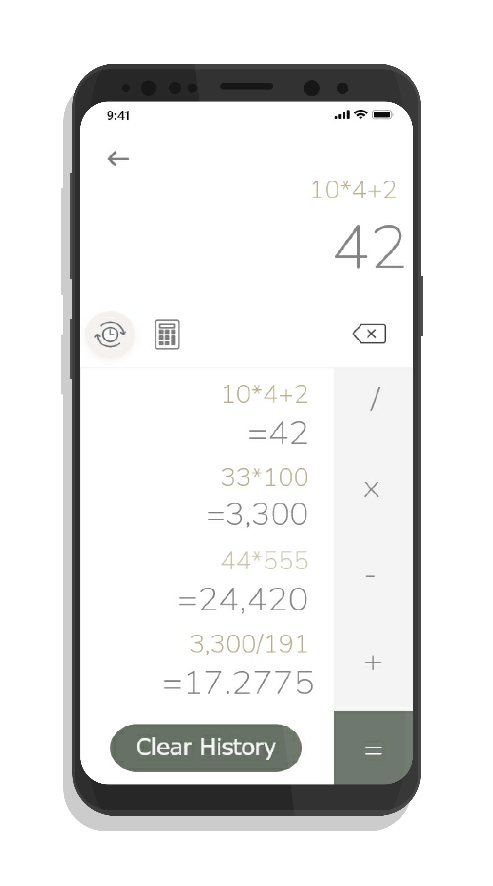


Figure 19: History Screen

Figure 20: Calculator Screen

Figure 21: Solution Screen

Figure 22: Navigation Screen

Figure 23: Main Screen

# Full details of Artefact

## Project 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 10.2.

## Preferred Methodology

For this software development Increment methodology is best suited.

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.

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, Software Product is incrementally developed and delivered.

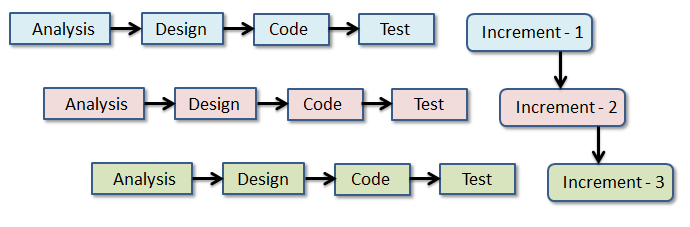


Figure 24: Incremental Model Visualization (EduCba, 2020)

Reasons to choose Incremental:

* Generates working software quickly and early during the software life cycle.
* Works on iteration, so it would be easy to develop promotional website at first and then mobile application.
* Parallel development can be done
* Supports change of requirements
* Testing and debugging is easier because of small iterations
* More flexible
* Easy to manage risks

## 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)

From all the above mentioned techniques, Object Oriented Analysis and Design techniques is most suited for this project as the core programming language to be used in this project is *Python*, which is Object Oriented programming language. Similarly, for the front end part *Java* will be used which is also based on OOP concept.

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*
* Word processor for report
  + *Microsoft Office*, Microsoft Excel
* Core Programming Language
  + *Python with Tesseract and OpenCV modules*
  + *Java*
* Code Editor
  + *Android Studio*
  + *Jupyter Notebook*
* Version Controlling
  + *Github*

# Model Development

## Data Collection and training

Machine learning is all about the data. Machine learning algorithms are supervised using huge amount of data so that they can make prediction based on information they gathered from those data. Such algorithms find relationships and patterns in the given training data and develop understanding of it. Based on those understandings they can make predictions on unseen data.

There are several factors to take care of during data training. One of the major is quantity of available features in the dataset. If there are not enough features to train the model, the model suffers from under fitting and the model performs poorly even on the training dataset. Whereas in its counter-part, if the training dataset has unnecessarily huge amount of features, it suffers from over fitting. In this case, model performs excellently on the training dataset but when exposed to unseen data it performs worse. So we need to find the right amount of data features to train our model.

In our case we don’t face such problem because we are performing on image data. The possible problem in such data is dimensionality. As the image is converted into an array of numbers that is perceived as a matrix by the computer. Converting every pixel of the image into array, it produces a matrix with huge dimension. We have to reduce its dimension without losing the information. This is where Principle Component Analysis comes to rescue. As we were using pre-prepared dataset, we do not have to worry about that.

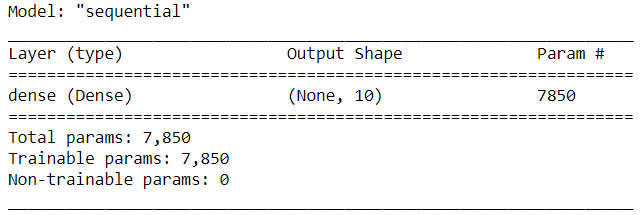
For the model development the MNIST dataset of handwritten digits was taken. The dataset has 60000 training images and 10000 testing images. The images are black and white with 28x28 pixel size. (LeCun, et al., 2021)

Multiple models were built and trained using the same dataset. The one with the best performance was chosen to be the final model. The data was separated into training and testing set so that our model can make predictions on data that has not been seen before and we can get actual performance of the model.

## Model Comparison and Performance Analysis

### Model 1

First model is made up of no hidden layers. It is comprised of 10 dense layers with input size 784. The Sigmoid function is used as activation function. Adam is used as an optimizer and for the loss calculation sparse categorical cross entropy is used. Accuracy is used as the performance metric. The model was trained with only 5 epochs. The model produced 92.51% of accuracy.



Below is the classification report of the model. It shows that the digit 1 was correctly classified most of the times and digit 2, 3 and 9 were correctly classified least.

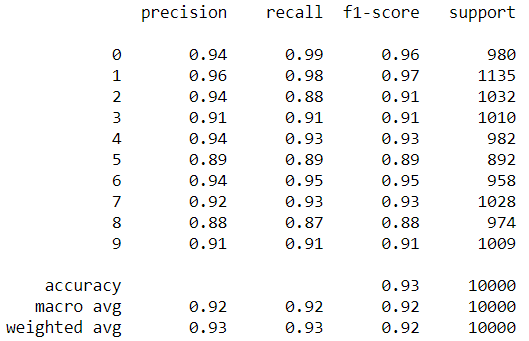
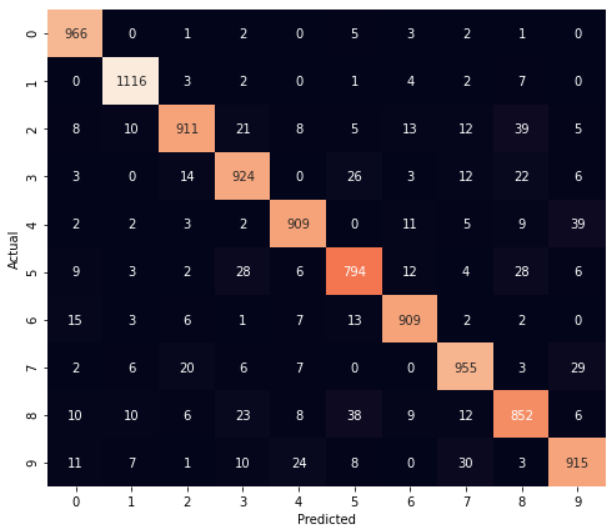


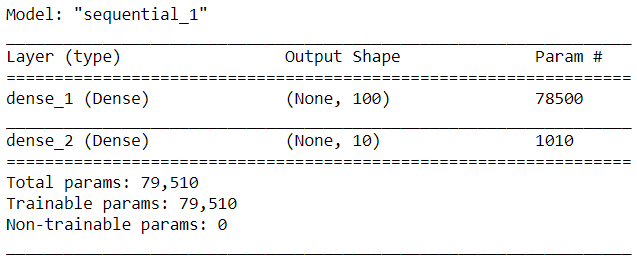
Figure 25: Classification report of first model

The below diagram is the confusion matrix of the model. This matrix compares the actual target values with those predicted by our machine learning model. As seen in the classification report above digit 1 was classified correctly most of the time. From the matrix below also we can see that digit 1 was classified most of the times i.e. 1116 times. The diagonal column represents the target values that are classified correctly.



### Model 2

In the second model we introduced hidden layers. 100 hidden layers were introduced with each having relu activation function. 10 other dense layers were also used with sigmoid function as an activation function. All the hyper parameters were same as in the first model except we increased number of epochs to 10 from 5.



Introducing hidden layers increased accuracy of the model drastically. Now the model produced the accuracy of 97.64%.

Below is the classification report of the model. It also shows that the digit 1 was correctly classified most of the times. The precision of prediction of other digits in this model has increased than in previous one.

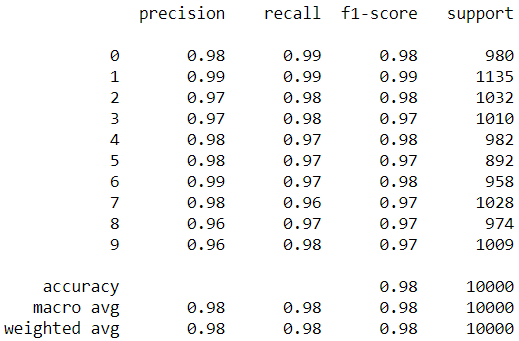
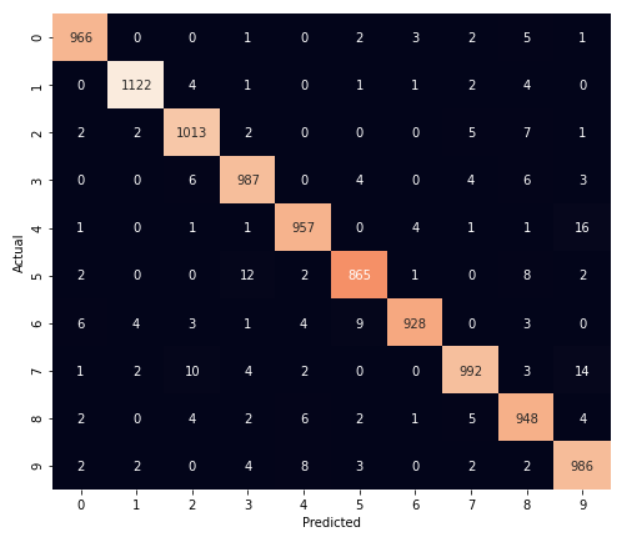


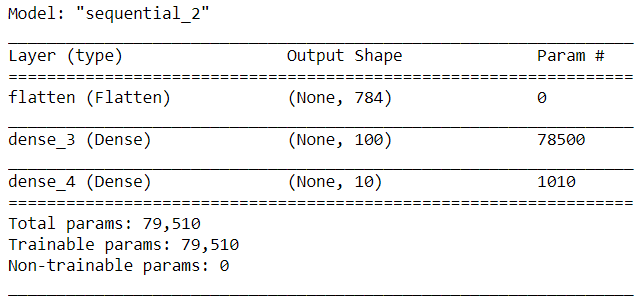
Figure 26: Classification report of second model

The below diagram is the confusion matrix of the model. This matrix compares the actual target values with those predicted by our machine learning model. As seen in the classification report above digit 1 was classified correctly most of the time. From the matrix below also we can see that digit 1 was classified most of the times i.e.1122 times which is higher than the previous model. We can see that other digits were also classified correctly more times relative to the previous model.



### Model 3

In this model we added flatten layer so that we don’t have to reshape our input dataset. All the layers and hyper parameters were left unchanged.



The addition of flatten layer increased the accuracy by very little margin. Now the accuracy of the model is 97.79%.

Below is the classification report of the model. The precision and recall of most of the digits are increasing than in the previous ones.

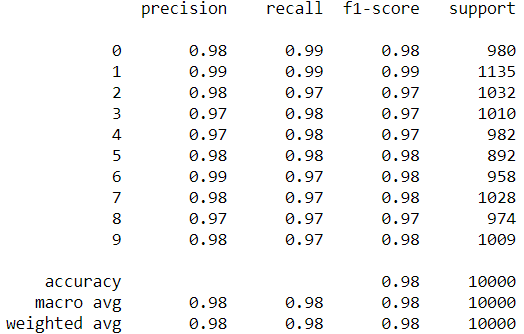
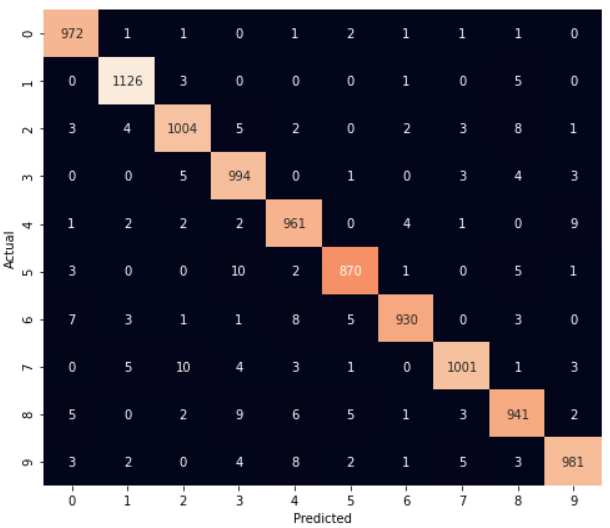


Figure 27: Classification report of third model

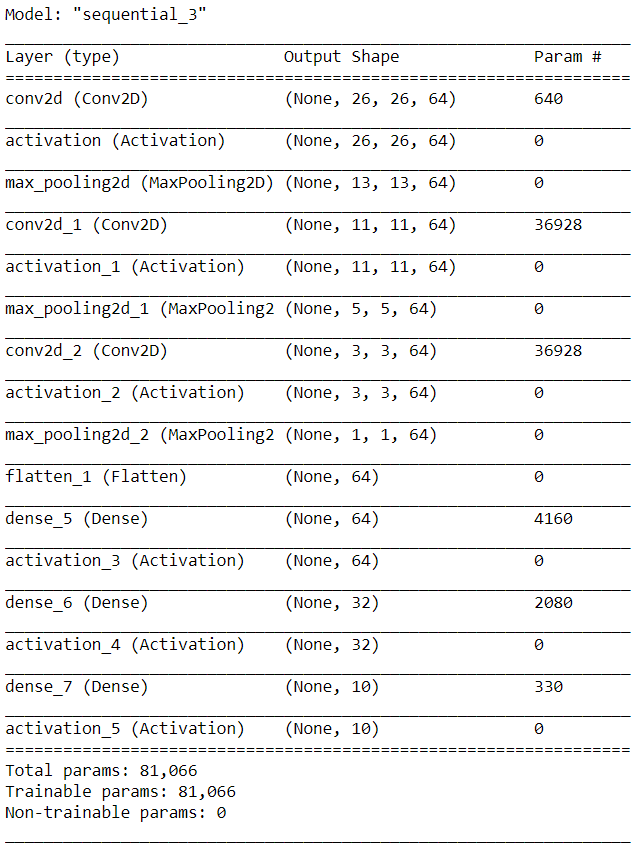
The below diagram is the confusion matrix of the third model. As seen in the classification report above, the number of correct classifications has also increased in the confusion matrix. From the matrix below we can see that digit 5 was classified correctly 865 times whereas in the previous model it was classified correctly only 794 times. We can see that other digits were also classified correctly more times relative to the previous model.



### Model 4

Finally in the fourth model, we implemented Convolutional Neural Network. Three convolutional layers were added. In each convolution layer max pooling was done with ‘relu’ activation function. In these layers filters were also used of size (3,3).

Now the output was passed to fully connected layer where flattening was done. This converted our 2D array to 1D. Finally in the last layer, ‘softmax’ activation function is used that gave class probabilities. The one with the highest class probability is considered to be the digit in the image.



On using accuracy as the performance metric, it gave accuracy of 98.61%. This is the highest accuracy among all the developed model. Thus this model is to be used.

From the classification report of the model below, we can see that precision, recall and f1-score of this model is highest than any other models. The scores are almost perfect with 99% accuracy.

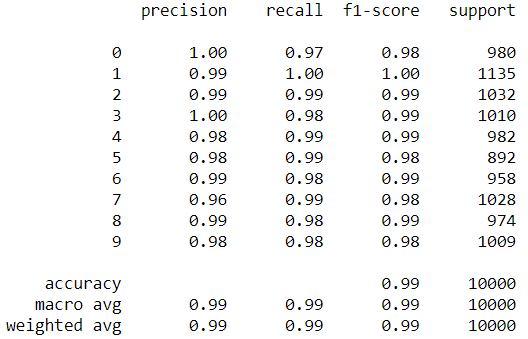
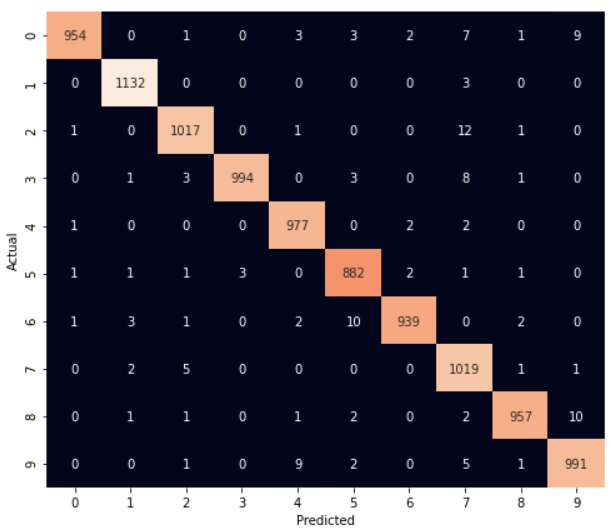


Figure 28: Classification report of fourth model

The below diagram is the confusion matrix of the fourth model. We can see that now very few misclassification are being made.



# Testing

As we are doing this project on Incremental fashion, each increment will give us simple working system with basic features. So, we can perform testing after every increment. And for the testing we will only conduct *Black Box Testing.*

## Basic calculation test

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **USER STORY ID** | **TEST CASE DESCRIPTION** | | **TO TEST BASIC CALCULATION** | | | |
| T001 | S001 | **TEST CASE DESCRIPTION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **STATUS** |
| To test basic arithmetic calculations. | 1. Access the application  2. Home screen appears  3. Click calculator button  4. Calculator opens up  5. Enter equation | 1. 2 + 2  2. 6 + 8 | 4  14 | 4  14 | PASS |

## Scientific calculation test

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **USER STORY ID** | **TEST CASE DESCRIPTION** | | **TO TEST SCIENTIFIC CALCULATION** | | | |
| T002 | S002 | **TEST CASE DESCRIPTION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **STATUS** |
| To test scientific calculations. | 1. Access the application  2. Home screen appears  3. Click calculator button  4. Calculator opens up  5. Enter scientific equation | 1. sin60  2. sin6+cos6 | 0.866  1.099 | 0.86602  1.09905 | PASS |

## Scanner functionality test

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **USER STORY ID** | **TEST CASE DESCRIPTION** | | **TO TEST SCANNING FUNCTIONALITY** | | | |
| T003 | S003 | **TEST CASE DESCRIPTION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **STATUS** |
| To test scanner functionality. | 1. Access the application  2. Home screen appears  3. Click scan button  4. Click scan now  5. Choose import option | NA | NA | Camera opened as scanner | PASS |

## Scanning options test

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **USER STORY ID** | **TEST CASE DESCRIPTION** | | **TO TEST SCANNING OPTIONS FUNCTIONALITY** | | | |
| T004 | S004 | **TEST CASE DESCRIPTION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **STATUS** |
| To test scanning options. | 1. Reach scanning screen  2. Click scan now | NA | Option to choose between camera and gallery | Dialog box with camera and gallery options prompted | PASS |

## Cropper functionality test

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **USER STORY ID** | **TEST CASE DESCRIPTION** | | **TO TEST CROPPER FUNCTIONALITY** | | | |
| T005 | S005 | **TEST CASE DESCRIPTION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **STATUS** |
| To test crop functionality. | 1. Reach scanning screen  2. Click scan now  3. Choose import option  4. Upon successful import cropper appears | NA | Cropper appears | Cropper with random size appeared | PASS |

## Result display testing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **USER STORY ID** | **TEST CASE DESCRIPTION** | | **TO TEST RESULT DISPLAY FUNCTIONALITY** | | | |
| T006 | S006 | **TEST CASE DESCRIPTION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **STATUS** |
| To test result display functionality. | 1. Access the application  2. Reach scanning screen  3. Import image  4. Click Calculate button | Image feed | Digit extraction and result display | Image is displayed at top, extracted equation below that and result at the bottom. | PASS |

## Result editing test

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **USER STORY ID** | **TEST CASE DESCRIPTION** | | **TO TEST RESULT DISPLAY FUNCTIONALITY** | | | |
| T007 | S006 | **TEST CASE DESCRIPTION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **STATUS** |
| To test scanned result edit functionality. | 1. Access the application  2. Reach scanning screen  3. Import image  4. Scanned result is displayed  5. Click in that text area | Image feed | Digit extraction and result displayed. | Digit extracted and upon clicking result able to edit digits. | PASS |

# Issues during the project­

* Hardware failure

One of the major issue that occurred during the project was hardware failure. The battery of the laptop used for the project was damaged and had to replace with new one. No matter how careful I was, I had not thought that this problem may arise. No matter what hardware problem may have occurred, I was always prepared. To mitigate this problem, project was stored in cloud storage as well as external hardware.

* Version Controlling

As I am a beginner programmer, I have no experience to manage the flow of programming. While adding new features I was messing up the previous code and working features would stop working anymore.

To overcome this issue I used version controlling tools like “Github”. This way I could fall back to the previous running code at any point of time. I was able to keep track of my progress as well as had courage to try and go beyond my limitation to test new features.

* Lack of proper 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 I needed huge amount of dataset to teach the system. It was impossible for me to produce that amount of data myself as well. I had to do lots of research to find the proper training dataset.

* Memory Shortage

Due to the insufficient experience in data training field I had no idea about the device specifications needed to train model with such huge dataset. As a result, while training the model the memory available in my device turned out to be insufficient.

To tackle this problem I used Google Colab, a free service provide Google. But unfortunately the memory provided by it was also not sufficient. Finally I used one of my colleague’s device to train the model.

# 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.

## Recommendation and further works

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 add a feature to plot graph of the equations provided.

We can also work on the area to increase the accuracy of character recognition. This system works almost perfectly for the digital images. But in the case of handwritten images there are still rooms to make improvements. We can increase the accuracy for handwritten digits even more. For that we can collect large variety of handwritten dataset. The system is based on English numerals only. So, another future task could be to conduct research on languages other than English and implement their numerals in this application.

# Critical Evaluation of the project

The project tries to develop a simple system that may be able to make users life somehow easy. It implements OCR technology in a common mobile device with only one aim to make mathematical calculation easy with the use of OCR.

As the project was proposed at the beginning I had absolutely zero idea about the knowledge and skills required. As the project proceeded I started to realize it was an over ambitious project. High programming knowledge and theory behind it was needed. Huge amount of research was needed. The knowledge I had was never enough so I had learn everything from the scratch. This affected the pace of the project. When some errors occurred, I had to halt whole project and search for the solution for weeks.

Now on the other hand, the project initiation and carrying did not happened as planned. The project was built in the incremental fashion. The features could not be delivered within the deadline. During the initial days of the project it took way more time to finish simple tasks. But as the deadline was approaching, the project picked up the pace. During those last hours, tasks were carried out within few days which otherwise would have taken weeks during the initial days. The project is completed one week later than it was planned.

In nutshell, the project was completed smoothly and not so much of problems occurred. It is completed before deadline and most importantly meets all the features as mentioned in the project proposal.

# Evidence of Project Management

## Gantt chart

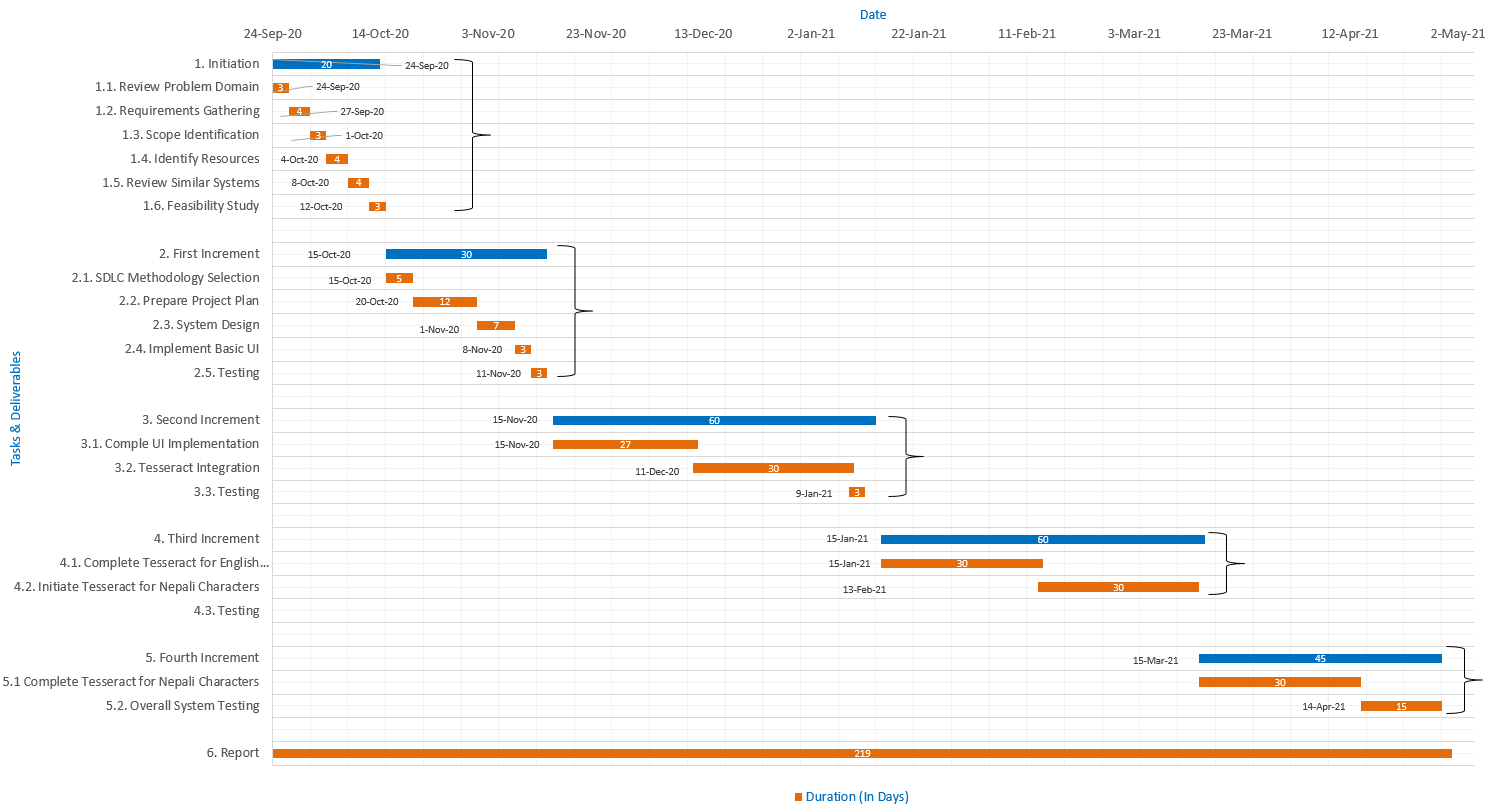


Figure 29: Gantt chart

# References

Briley, W. R., 2014. *An Introduction to Vectors and Tensors from a Computational Perspective,* Chattanooga: University of Tennessee.

CERN, Geneva, 1996. *19th CERN School of Computing.* Geneva, s.n.

EduCba, 2020. *Incremental Model.* [Online]   
Available at: https://www.educba.com/incremental-model/  
[Accessed 20 September 2020].

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

Kagami, S., 2010. *High-Speed Vision Systems and Projectors for Real-Time Perception of the World,* Sendai: Tohoku University.

Khan, A. I. & Al-Habsi, S., 2019. *Machine Learning in Computer Vision.* Muscat, Remote Sensing and GIS Research Center.

LeCun, Y., Cortes, C. & Burges, C., 2021. *THE MNIST DATABASE.* [Online]   
Available at: http://yann.lecun.com/exdb/mnist/  
[Accessed 10 April 2021].

Mathway, 2020. *Mathway.* [Online]   
Available at: https://www.mathway.com/  
[Accessed 08 October 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].

Prajapati, S., Joshi, S. R., Maharjan, A. & Balami, B., 2018. *Evaluating Performance of Nepali Script OCR using Tesseract and Artificial Neural Network.* Kathmandu, IEEE.

Ranjay, K., 2017. *Computer Vision: Foundations and Applications,* s.l.: Stanford University.

Sebe, N., Cohen, I., Garg, A. & Huang, T. S., 2005. *Machine Learning in Computer Vision.* 1st ed. New York: Springer.

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].

Sikka, A. & Wu, B., 2012. *Camera Based Equation Solver for Android Devices,* California: Stanford University.

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].

Wu, J., 2017. *Introduction to Convolutional Neural Networks,* Nanjing: National Key Lab for Novel Software Technology.

Zhang, X. et al., 2019. *Research on Scene Classification Method of High-Resolution Remote Sensing Images Based on RFPNet,* Beijing: MDPI.

# Appendices