1. Literature Review

1.1. Development of OCR

1.1.1. 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)

1.1.2. 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)

1.1.3. 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)

1.1.4. 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 *1.4.*

1.2. 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.

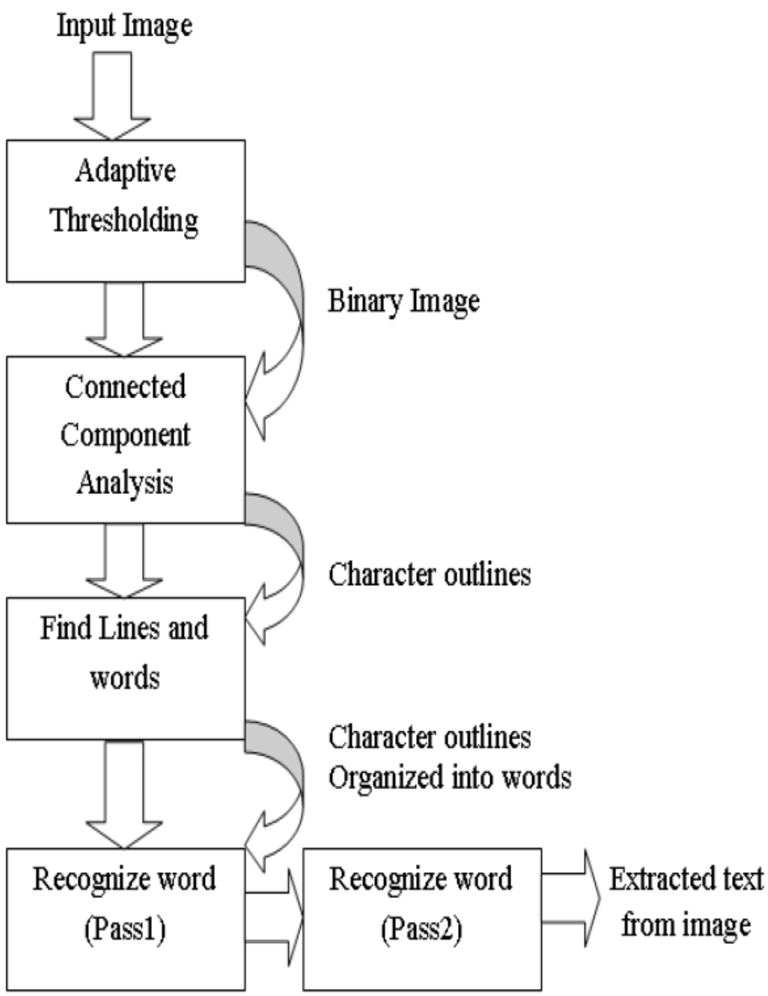


Figure 1: Architecture of Tesseract OCR (Patel & Patel, 2012)

1.3. 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 downsampled 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 2: Segmented Character from input image

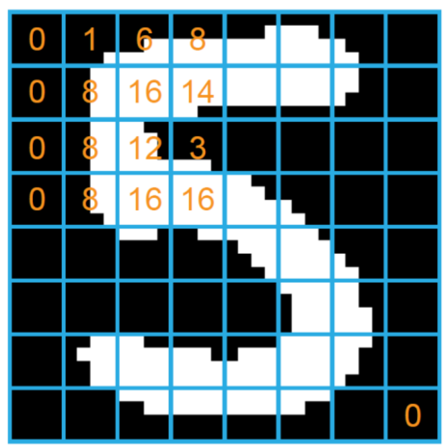


Figure 3: Downsampled Character

1.4. Similar Systems

1.4.1. 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)

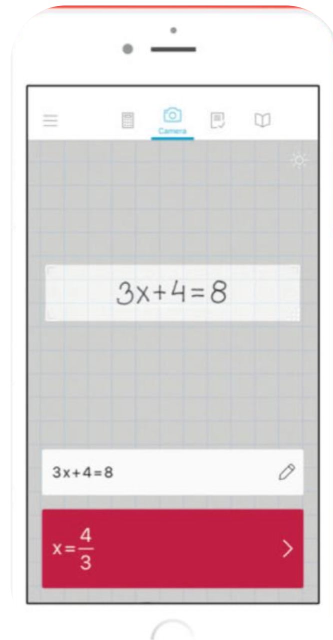


Figure 4: PhotoMath Application

1.4.2. 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)

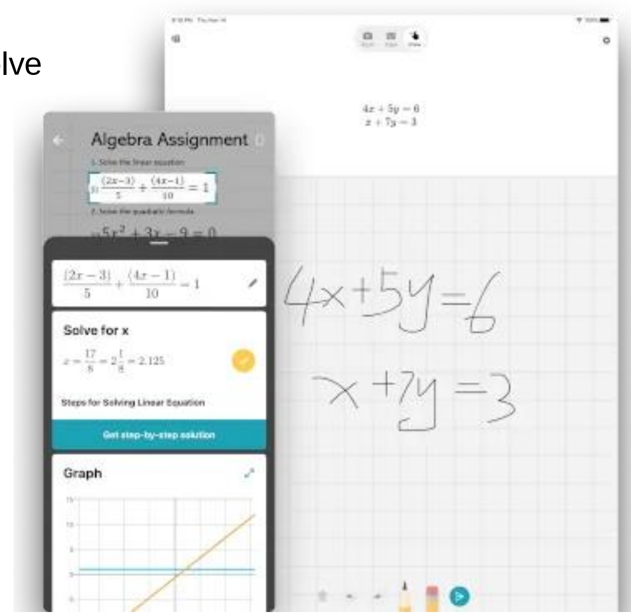


Figure 5: Math Solver Application

1.4.3. 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 6: Mathway Application

1.5. Review of Similar Systems

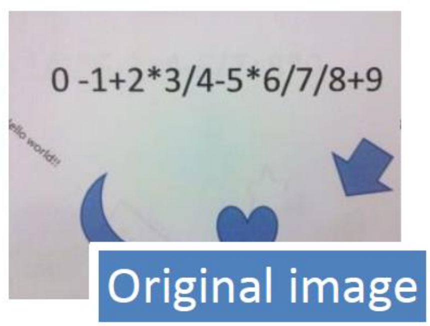
All of the above mentioned systems have their separate math content team. Because of this there is solution to every math problem from arithmetic to calculus. These systems read and solve mathematical problems by just using the camera of mobile phones. The most astonishing feature of these systems is 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.

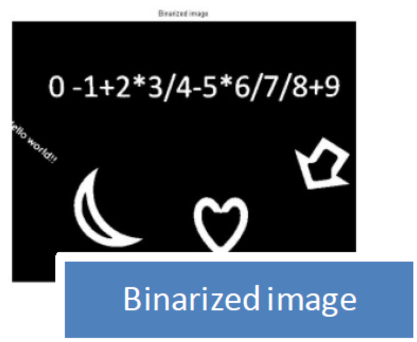
The only bad aspect or the limitation of these systems is that they support only English language. They can only perform calculations on English numerals. This is the aspect where my project is going to work on. My smart calculator will be able to perform calculations on Nepali numerals too.

1.6. Mobile Camera based OCR for Mathematical Calculation

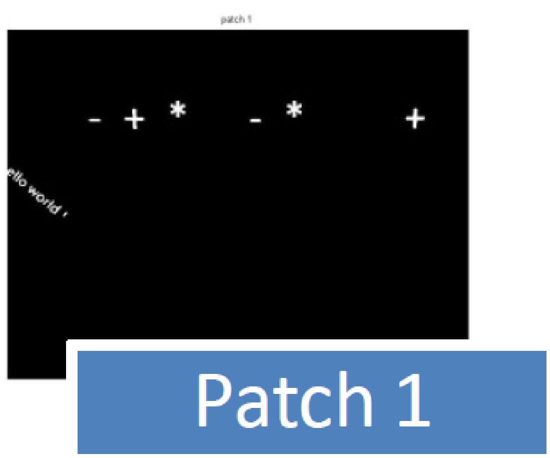
To use the mobile camera as an OCR device and perform mathematical calculations, following steps are to be followed.

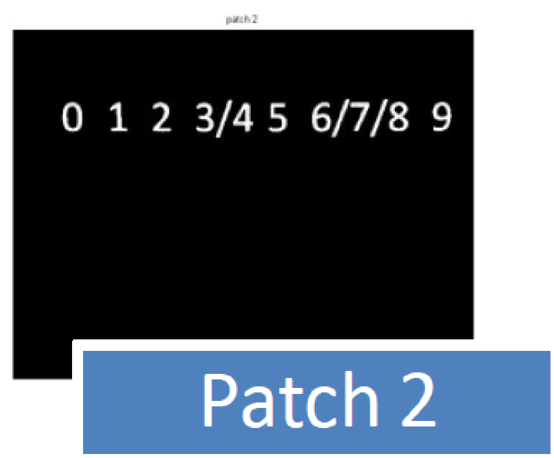
After capturing the image it is then converted into grayscale image on which noise reduction and sharpening action is carried out. Then the captured image goes under binarization process. This helps to compensate the over/lower exposure of the light on images.



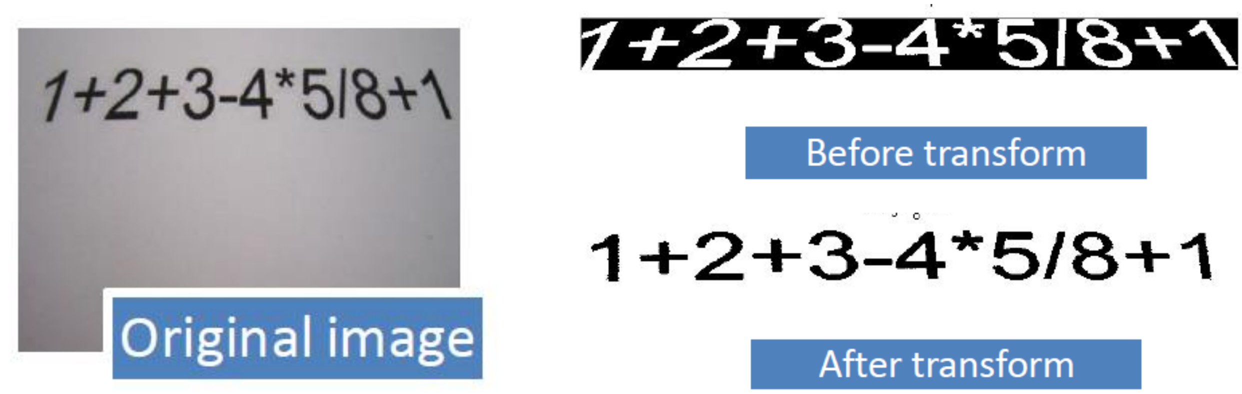


Now, the binarized images are separated into different patches containing only one category of data. In mathematical calculation this helps to distinguish operators from numbers.





Most of the times images captured have high possibility to be rotated or have some perspective distortions. We need to adjust such images before feeding to get more accurate results. We can use the perspective transformation method to adjust such images. (Wang, et al., 2012)



Now, the OCR engine extracts numbers from the image in the String data type. We can convert it into integer and feed for calculation and finally get the output.

In general the whole process can be carried out as the chart below shows.

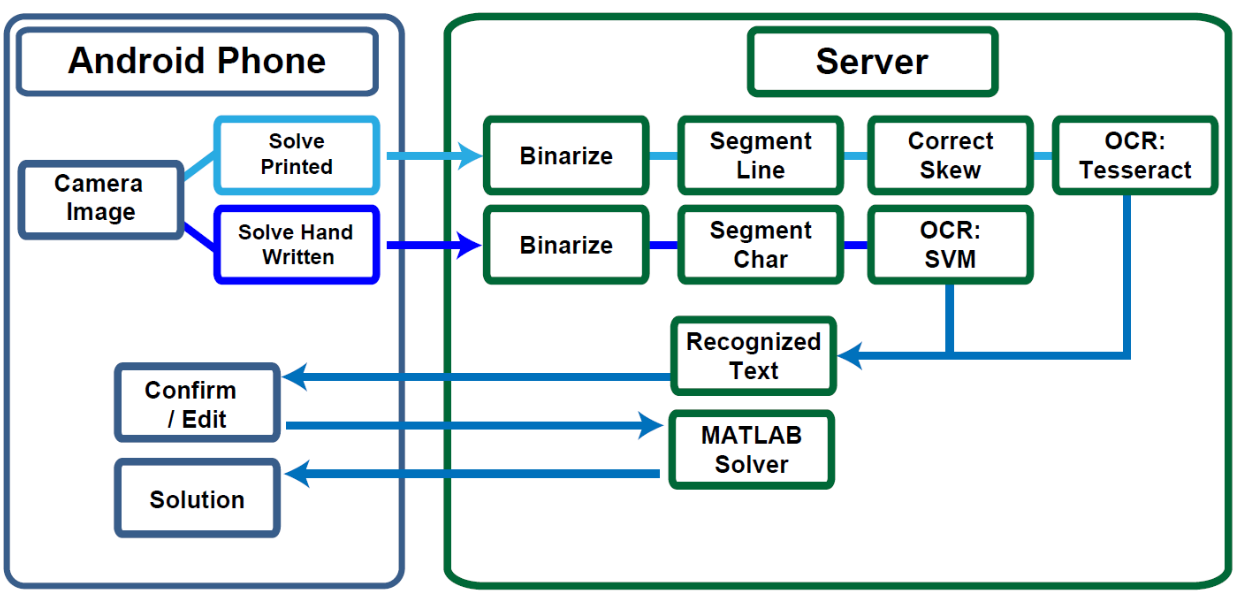


Figure 7: Process for mobile camera based OCR

1.7. 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%

References

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

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].

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].

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

Wang, L., Dai, J. & Du, L., 2012. *Mobile Camera Based Calculator,* s.l.: Stanford University.