

# PicMath : A Camera Calculator

Silpa P<sup>1</sup>, Vinya Vijayan<sup>2</sup>

<sup>1</sup>Dept of Master of Computer Applications

<sup>2</sup>Asst.Professor, Dept of Master of Computer Applications

<sup>1,2</sup> College of Engineering, Trivandrum, Kerala, India

**Abstract-** *PicMath is an application described as a "Camera Calculator", it uses a camera to identifies problems in mathematics and to provide the solution on screen. It is used to learn mathematics. It helps to study how to solve mathematical problems, check homework assignments and study for upcoming exams. The mathematical areas in which this application will work is as follows: arithmetic, integers, fractions, decimal numbers, powers, roots, logarithms, trigonometric identities, factorials. It provides instant help, Smart camera calculator, Handwriting recognition.*

**Keywords-** Image Processing, Optical Character Recognition(OCR), Operator and operand Recognition, pytesseract.

## I. INTRODUCTION

PicMath will help to interpret problems with comprehensive math content from arithmetic to simple calculus to drive learning and understanding of fundamental math concepts. With PicMath, study how to approach problems or check your homework for any printed or handwritten document. It work on the mathematical areas are as follows: arithmetic, integers, fractions, decimal numbers, powers, roots, logarithms, trigonometric identities, factorials, relational operations. It also solve arithmetic operations including multiple operations as well as single operation multiple times, percentage calculation etc.

## II. METHODOLOGY

An open source OCR engine tesseract is used to recognize text on an image. Tesseract takes its input as a binary image so that it needn't any page format analysis. A traditional processing of tesseract is as follows:

- Analysis of connected components: store outline of each component
- proceeds a double-pass process: pass 1, to identify each and every word and passed to an adaptive classifier. Pass 2, solve fuzzy spaces.

Take user provided image that contains mathematical equations to be solved, and use it as the input to the application. This image then undergoes some preprocessing,

and pass to OCR engine tesseract. The tesseract engine will perform the following operations on the inputed image file:

Line finding: fitting baseline, detection of pitch and chopping, finding proportional word. word recognition: cutting joined characters, binding split characters. Static character classifier, Analysis based on linguistics, adaptive classifier. The tesseract engine retrieves the characters on the image that provided as the input. Store that characters into a text file. The text file is a common file to all images. The the text file is passed to the main program that contains a library to solve all mathematical problems. Check each character from the text file and identifies the operation on it. Separate the operations and operands on the text file. Then call function inside the library corresponding to the operation. Perform the operation and find the result. When the user need to view the output on screen then click the view result button. The application PicMath will print the output on screen. If the user need to view the solution finding part then a step-by-step solution will be provided on the screen with animated texts. If there is multiple procedures to solve a single problem then all possible procedures will be provided by this application.

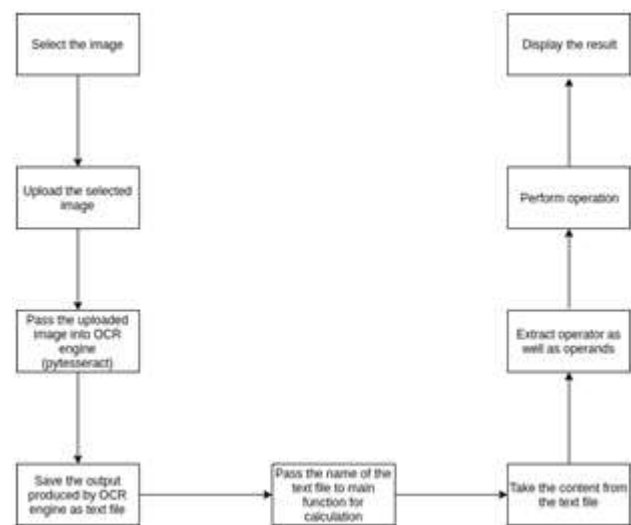


Fig. 1: The overall working of the application

## III. IMPLEMENTATION

### 3.1 Data Preprocessing

Quality of the original inputted image is a crucial area here. If the quality is good, it will be possible to achieve good OCR results. But the input is not clear originally, then the engine results will most likely include errors. The better the quality of the input, the simpler it is to identify characters from the image, the higher the accuracy of OCR engine will be. For the OCR engine to distinguish a character from the background the image should have:

- sharp character borders
- high contrasts
- well aligned character
- negligible pixel noise possible

The first and basic step for having accurate character recognition it should ensure a good quality input image. Check that the paper documents aren't damaged, wrinkled, discolor or colorless print.

Scaling to the right size: input should be undergoes correct size scaling usually is of more than 30px (pixel) and less than 140px. Increase the contrast: Increase the contrast while taking the photo or making the text as image. Binarize the image: convert the multicolored image to a black and white image to reduce image size. Tesseract automatically do the binarization itself. Remove noise: pixel noise is a result of poor scanning or poor quality so try to reduce that initially itself. De skew: simply rotation, De-skewing the image to make it in the correct format and correct shape. After performing the data preprocessing perfectly then pass the image into the OCR engine. Then the OCR engine pytesseract will provide highly accurate result for each character recognition. Then make those characters to a text file.

### 3.2 Performing the operations

The text file is the input to the core functioning area of this application. Pass the text file to the core program code is the next process. The core program code identifies each character and separate the operations and operands in it. Store that on two separate lists for further requirements. Check each operators on the list. If the operator list contains only single operation then run the code to solve that problem. Otherwise go to area where the code solve BODMAS problems. Store the result. When the user need to display the result the inform the application to display the result on screen.

### 3.3 Test the application

For testing this application created as many images as possible that contains different types of mathematical equations and problems. The images have extensions .png,

.jpg, .jpeg. Most of the images are created as a screen shot of file extension .png. The mathematical problems includes: simple arithmetic (addition, subtraction, multiplication, division, modulus), percentage calculation, multiple operations on multiple numbers (BODMAS), single operation multiple times, operations on decimal numbers without limiting decimal places, trigonometric identities, logarithms, exponents, squares and square roots, relational operations, etc.

## IV. RESULT

The result of this application is highly correct and accurate. All the operations provided as input executed correctly with a few Milli seconds of time. One of the most significant of this application is that there is no limit for the number of digits can the problem have. Even though the number exceeds 17 digits the solution is perfect than a calculator.

Result of some of the operations are given:



fig.2: solution of a BODMAS problem



fig.3: solution of percentage problem



fig. 4: solution of a relational operation

## V. CONCLUSION

Mathematics is one of the rigid subjects in school times, that's why owning a calculator seems like a necessity for students. The difficulty of typing the lengthiest problem on calculator is the idea behind PicMath. PicMath is an application that can read and solve mathematical expressions using camera on your device in real time. It is the best application to students those who have access to a guide and difficulty with solving mathematical problems. Parents can also use the PicMath application when help to solve problems to their children. PicMath currently supports basic arithmetics, factorials, decimal numbers, percentage calculation, multiple operation on multiple numbers, single operation multiple

times on multiple numbers, and several functions like basic trigonometric identities, simple logarithms, exponents, square roots, etc.

### REFERENCES

- [1] S.V. Rice, F.R. Jenkins, T.A. Nartker, “The Fourth Annual Test of OCR Accuracy”, Technical Report 95-03, Information Science Research Institute, University of Nevada, Las Vegas, July 1995.
- [2] R.W. Smith, “The Extraction and Recognition of Text from Multimedia Document Images”, PhD Thesis, University of Bristol, November 1987.
- [3] R. Smith, “A Simple and Efficient Skew Detection Algorithm via Text Row Accumulation”, Proc. of the 3rd Int. Conf. on Document Analysis and Recognition (Vol. 2), IEEE 1995, pp. 1145-1148.
- [4] Nicolas D. Jimenez, Lan Nguyen, “Mathematical formulas Recognition of Handwritten Mathematical Symbols with PHOG features”