Equation Solving Using Image Processing

Archit Tiwari[1], Ayush Tiwari[2], Abhidarsh Mishra[3], Sachin B. Wakurdekar[4]
[1,2,3]Student, Bharati Vidyapeeth (Deemed to be) University College of Engineering, Pune
[4]Asst. Prof. Bharati Vidyapeeth (Deemed to be) University College of Engineering, Pune

***Abstract* - In this article we show the Android system mobile app equation solver that takes a photo with the camera. Equation and present the solution. The program is capable of solve simple arithmetic equations (addition, subtraction, and multiplication) and systems of linear equations up to two. handwritten or computerized variants.**

I.INTRODUCTION

Mathematical equations are a very widespread part of student life. In general, you have to use a calculator or a computer. Solve complex equations. Depending on the auxiliary device, Equations must be entered in a specific format. Our objective you are developing an app for mobile devices that fills the gap between technique and the traditional pencil and paper approach intuitive way. User can take a photo with the camera. For a mathematical expression, either on a computer calligraphy or handwritten text, and the solution will be calculated. It is displayed on the mobile device. First, we limit the scope of the task to a simple solution arithmetic equations, quadratic equations and systems linear equations for two variables. Mobile applications to solve such equations exist. However, these applications basically predefined equation templates used by users enter numbers in static text boxes. Although restricted interface provides powerful input capture, it's bulky. For the user, who must click multiple times to enter a file equation. It will be the most user-friendly method take a photo of the written expression.

II. OVERVIEW

The Process for solving equation from an image is as follows:

1. Capture image
2. Crop Image
3. Preprocessing
4. Recognize text
5. Prompt for user confirmation
6. Solve equation

The system contains in 3 phases the first phase contains image capturing and cropping, the second phase contain the image processing and text recognisation and third part contain parser and solver.

2.1. Image Capture

In the mobile application the user has two options. first - "printed solution" and second - "handwritten solution" that captures a file Image and begin two different python scripts on the server the image is reduced to 640 x 480 pixels from its original size to hurry up sending to the server.

2.2. Crop Image

Since the background can add noise and scale back the accuracy of the reader, the crop perform is supplemental to the system permits the user to make a picture containing solely text.

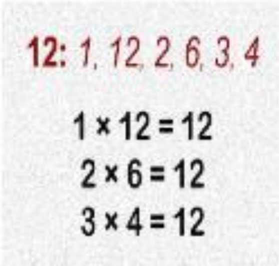
2.3. Preprocessing

Once you upload an image of the math expression to a file system, the image is pre-processed. Preprocessing step includes image enhancement and grayscale conversion.

This is the step makes the image ready to be processed correctly. Eliminates the chances of minor errors in the OCR process. For this, several filters are used. In this next, filters are used to improve the recognition rate.

i. RGB grayscale (dimensional reduction) Since the input color image has a 3D pixel value (RGB). Therefore, the matrix obtained for this input color image is three-dimensional matrix.

It is very difficult to make the image 3D matrix processing technology, and therefore smoother and more convenient image processing gray format. The gray image has a 2D pixel value which falls between [0, 255] depending on the gray value of the pixel.



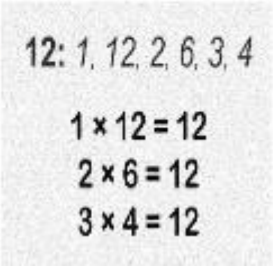
12: 1, 12, 2, 6, 3, 4

1 × 12 = 12

2 × 6 = 12

3 × 4 = 12

Input image



12: 1, 12, 2, 6, 3, 4

1 × 12 = 12

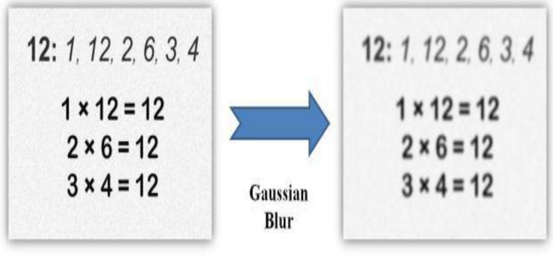
2 × 6 = 12

3 × 4 = 12

Gray Scale image

ii.Gaussian blur filter

Once you have a grayscale image, filters are applied to it image smoothing. Smoothing, also called darkening, is simple the image processing process is often used much. There are reasons to use this filter, but here is a guide. Filters are used to reduce noise.



Grayscale Image Blurred Image

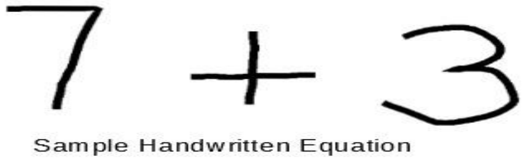
2.4. Recognize Text

Text recognition is split into 2 separate categories - digital written expressions and handwritten expressions. Digital written expressions are processed with Tesseract, Handwritten expressions are processes with a Support-Vector Machine prediction model.

In order to enhance detection accuracy the quantity of characters within the information are restricted to digits (0 - 9), 2 variables (a, b), and a set of mathematical operators (+,-, x, and =).These characters are ample for composing each quadratic equations and also the majority of a system of linear equations in 2 variables. The lexicon is often simply swollen to include a lot of characters within the future.

Tesseract worked well for printed text, with the

The detection rate is better than 87% for lines within your database. However, the detection rates for handwritten text are below 53%, probably due to a size difference in writing and a shortage matches the sources in your database. Machine learning algorithm. Instead, it is based on Support Vector Machines.



Handwritten Equation & Separate counter symbol [1]

Separate Contour of each digit/symbol

SVM is a supervised learning method that analyzes data recognize patterns. It is often used for classification and regression analysis. The prediction model is created using libSVM, a set of tools for SVM training and modelling. It was developed at the National Taiwan University.

2.5. User Confirmation and Equation Solver

Even for a system with a single character recognition rate 95%, the probability of discovering a set of equations with the number 12 characters are just over 50% perfectly correct [2] that it is extremely important to provide the user with a means of confirmation or to edit the recognized text. . When confirmed, express it is sent back to the server where it is resolved.

After the equation is confirmed by the user, the equation is converted into a readable expression by inserting the missing operators. For example, the equation "2 + 5 = 7". Then the resulting solution is sent back to android device.

III. FUTURE SCOPE

The prediction rate of the SVM model is highly dependent on a training data set. Currently, only the training dataset contains our handwriting. Ideally, a larger training data set of various individuals, to accommodate them for long writing styles. It will be the extension so the system trains on the go. Every time the user sends a confirmation of an equation, which corresponds to 64 dimensional vectors can be added to detected characters to an existing training data set.

This would allow training the dataset grows quickly and efficiently without the need to do separate training[3].

IV. CONCLUSION

In this article, we describe an Android-based system for capture and dissolve an image containing mathematical material expression. The system consists of two parts: Android application to capture and display the camera image the recognized text and the solution, and the server to perform image processing, text recognition, and equation solving algorithms.

Camera image for a mathematical expression sent from your Android phone to the server, where it is dual with MSER. For printed text, distort perspective fixed with RANSAC, text recognition is done with Tesseract OCR Engine. For handwritten text, the SVM prediction model trained with libSVM is used in performance text recognition. The detected text is sent back to Android User confirmation device. Proven equation then it is resolved on the server and the solutions it is displayed on the Android device.

REFERENCES

[1] Handwritten Equation Solver in Python https://www.geeksforgeeks.org/handwritten-equation-solver-in-python/

[2] Optical Recognition of Handwritten Digits Data Set. Machine Learning Repository, University of California at Irvine. Available: http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits

[3] Aman Sikka and Benny Wu," Camera Based Equation Solver for Android Devices"