IOSR Journal of Engineering (IOSRJEN)
e-ISSN: 2250-3021, **p**-ISSN: 2278-8719
Vol. 3, Issue 5 (May. 2013), ||V1 || PP 42-44

Linear equation solver in Android using OCR

Amey Chavan[1], Asad Naik[2]
Department of Information Technology, Trinity College of Engineering, Pune, India
Department of Information Technology,Trinity College of Engineering, Pune, India

Abstract: In this paper we demonstrate an Android application that takes a camera image of a simultaneous linear equation and displays the solution. Application can solve hand written or computer printed equations. In this paper we demonstrate method for Optimizing Image for better accuracy of OCR.

Keywords: *Android, Equation solver, math solver, OCR, Text Recognisation.*

I. INTRODUCTION

Mathematics is very important part of a person's life. The person has to know many formulae to solve equation. And has to depend on the assistive device like calculator or programs, but the equations need to be entered in a specific format. Our goal is to develop a mobile based application that bridges the gap between technology and the traditional pen and paper approach by new technology. Nowadays all people carry Smartphone with and android dominating Smartphone market. The idea of solving math equations is quite an intriguing. There are many app that allows user to solve equation by entering the variable and operator in predefine format. Hence we proposed a project that will allow user to just capture image containing the linear equations and solve it.

II. LITERATURE SURVEY

2.1 OCR Basics

The steps of OCR, as described by Garrisetal (1998) are as such:

1. Preprocessing (including line isolation)
2. Character segmentation
3. Segment reconstruction
4. Character recognition
5. Word and phrase reconstruction

Optical Character Recognition or (OCR) software uses machine learning methods to convert digital images of text into text files that may be easily modified with text editing software. usually the text are in ASCII or UTF-8 Format. OCR was basically used for Digitalization of Printed Data. Specifically, OCR software must adjust the effects of poor lighting, low-quality images and Rectify the mistake automatically with minimum human intervention.

2.2 TEXT RECOGNITION

Text recognition is divided into two separate classes -computer printed expressions and hand written expressions. Computer printed expressions are processed with Tesseract [2],a free Optical Character Recognition (OCR) engine originally developed by HP Labs and now maintained by Google. The Tesseract Shows significant results up to 95% accuracy with a proper image**.**

III. OVERVIEW

The general approach for solving an image is as follows:

1) Capture image.
2) Corp Image
3) Process Image
4) Recognize text.
5) Prompt for user confirmation.
6) Solve equation.

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

3.1 Capture Image

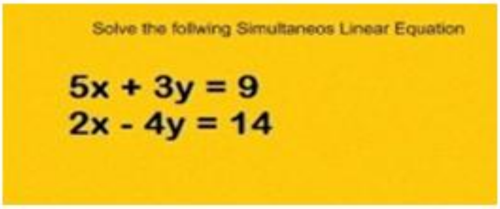
The ocr technology is depend on the user to function properly, the user must capture photo with proper angle to avoid distortion of image[3] and image must be minimum of 400 x400 pixels.

3.2 crop image

Since the background will add noise and will lessen the accuracy of ocr engine, the crop feature is added to system to allow user to corp the photo contain the text only.

3.3 Image Processing

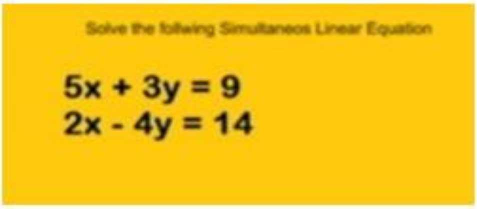
A Gaussian blur is the result of blurring an image by a Gaussian function. It is a widely to reduce image noise and reduce detail .Gaussian blur can be used in order to obtain a smooth grayscale digital image. We have include blur algorithm to reduce the noise of the photo Binarization and segmentation are crucial steps in identifying the regions of interests. Otsu's adaptive thresholding was used to threshold the image**.** Color images can also be thresholded. One approach is to designate a separate threshold for each of the RGB components of the image and then combine them with an AND operation [4]. Otsu's Algorithm is used to capture histogram shape-based image thresholding. The algorithm assumes that the image to be threshold contains two classes of pixels the threshold coverts the image into two bit image just containing white and black pixel



Solve the follwing Simultaneos Linear Equation

$5x + 3y = 9$
$2x - 4y = 14$

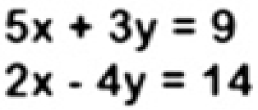
Fig 1) Original image



Solve the follwing Simultaneos Linear Equation

$5x + 3y = 9$
$2x - 4y = 14$

Fig 2) Image after blur



$5x + 3y = 9$
$2x - 4y = 14$

Fig 3) Image after threshold

3.4 Text recognisation

By limiting the number of alphabets to a-z and 0-9 and accuracy was greatly improved. Tesseract performs well for fonts that exist in its database, with accuracy in the range of 85-90%. The notable exceptions were the characters '0',' -', '=', and 'a'. The character '0'was at times mistakenly identified as 'o' and the character '3' was misidentified '8'. The font size played a big role in detection the accuracy was depended of the font size

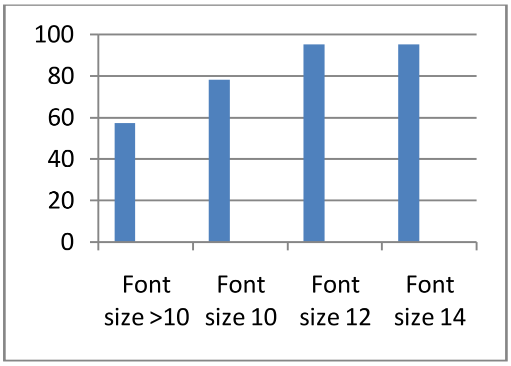


fig 4) Affect of font size on Accuracy

3.5. USER CONFIRMATION AND EQUATION SOLVER

Even for a system with a single character recognition rate of 95% the probability of detecting a set of equations with 12 characters completely correct is only slightly above 50% [1],. It is thus critical to provide the user with a means to confirm or to edit the recognized text. Upon confirmation, parser is started the equations is broken to group a_1x, b_1y and c_1

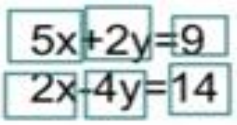


Fig 5) Grouping of variable blocks

And using Cramer Rule $x_i = \det(A_i) / \det(A)$ where A_i is the matrix obtained by substituting for the i^{th} column of A, and similarly y **the equation can be solved**

IV. FUTUERE SCOPE

The prediction rate of the proposed system is highly dependence the training dataset. Currently, the training dataset only contains Computer Fonts. Ideally, a larger training dataset from multiple individuals should be collected, to accommodate wide ranging styles in writing. An extension would be very beneficial for hand written text.

V. CONCLUSIONS

In this paper, we describe an Android-based system for capturing and solving an image containing a mathematical expression. The image is capture, cropped and then blurred to denoise image and then image is biniraize using otsu algorithm. Text recognisation is done by Google Tesseract. The detected text is sent back to the Android device for confirmation by the user. The confirm equation is then solved with parser and cramer rule.

VI. ACKNOWLEDGEMENTS

We like to thank Charulata Dhulekar and Wasim Ansari from Department of Information Technology, for their Expertise in Image processing. Also we like to thank Prof.Bilkis Chandargi for guiding us during Entire Project

REFERENCES

[1]. Optical Recognition of Handwritten Digits Data Set. Machine Learning Repository, University of California

[2]. Tesseract Optical Character Recognition Engine. Available: http://code.google.com/p/tesseract-ocr/

[3]. Sam Tsai, Huizhong Chen, David Chen, Ramakrishna Vedantham, Rade Grzeszczuk, and Bernd Girod, ”Mobile Visual Search Using Image and Text Features”, Proc. 45th Annual Asilomar Conference on Signals, Systems and Computers, November 2011.

[4]. M. Sezgin and B. Sankur (2004). "Survey over image thresholding techniques and quantitative performance evaluation". Journal of Electronic Imaging 13 (1): 146–165.