Image-based Calculator for Handwritten Mathematical expressions

Nilotpal Saikia[1], Nikhil D[2] and Hemanth Kumar[3]
Indian Institute of Technology Hyderabad, Kandi,Sangareddy, India, 502285
[1]ee16mtech11011@iith.ac.in , [2] ee15mtech11021@iith.ac.in, [3]ee16mtech11008@iith.ac.in

Abstract— **This report introduces an “Image based Calculator” that can recognize and evaluate handwritten mathematical expressions from an image. The difficulty of hand-written expressions recognition is due to variation of characters and poor alignment of text line. The application is able to solve expressions that include digits, the basic arithmetic operators (addition, subtraction, multiplication and division.**

Keywords— Image segmentation; contours; bounding box; neural netwokrs;convolution neural networks;

I. Introduction

The motivation for this project comes from the time and effort of entering equations into a calculator or computer. When doing computational work, a person has to go through the painstaking and time-consuming process of entering equations into a calculator or scientific computing program even after writing it all out by hand. Also, many potential users of these resources are unfamiliar with the syntax of these programs and cannot take advantage of the more advanced functions. Presently available mobile apps require the user to enter information on the screen instead of on handwritten paper.

In this project, we limited the initial scope of the application to solve expressions that include digits, the basic arithmetic operators (addition, subtraction, multiplication and division. Subsequently it can be extended to include more complex operators.

II. System Overview

The basic steps for acquiring and solving an equation from an image are mentioned below:

1) *Capture image*
2) *Convert to grayscale*
3) *Binarize*
4) *Segment characters*
5) *Character Recognition*
6) *Calculation*

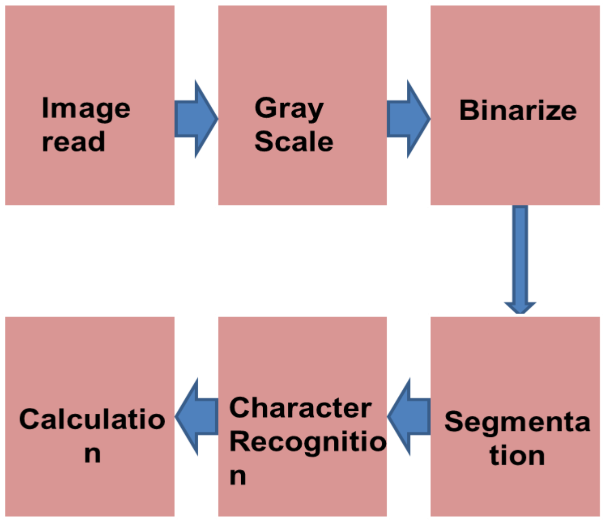

Fig. 1: Block diagram of the project

III. Image Capture

The equation is to be written on a white page preferably with a ball pen.

IV. Sementation

The objective of preprocessing is to take the captured image and segment the characters. The image is converted to grayscale. The image is then binarized using thresholding value of 127. This binarized image is then inverted and segmented using the findContours OpenCV function. To remove noise, only contours that have either a width or height of 10 pixels are kept. From these valid contours, any contours that a r e completely bounded by the bounding boxes of other valid contours are removed. This process ensures that inner contours (such as the circle inside a 6) are not classified.

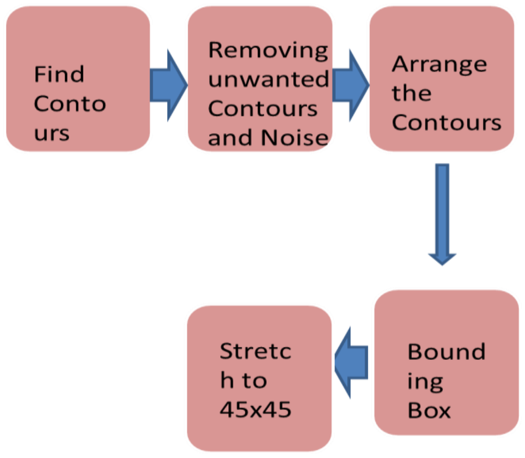

Fig 2: Segmentation Block diagram

` V. Recognition

Different classifiers have been tried for recognition of the handwritten characters. Their classification results are listed in the table below:

S. No.	Classifier Used	Recognition rates within its own database	Recognition rates outside its own database
1.	Multi Layer Perceptron	80%	Very Poor
2.	Radial Basis Neural Network	95%	Very Poor
3.	CNN Using Tensor Flow Inception	98%	Good

Table 1: Results of different classifires

So it is the CNN using inception module of Tensor Flow which gave good recognition performance.

VI. Google Net

Google made the use of multi layered CNN simplified with the introduced of the Inception module. GoogLeNet is a 22 layer CNN and was the winner of ILSVRC 2014 with a top 5 error rate of 6.7%. This was one of the first CNN architectures that really strayed from the general approach of simply stacking conv and pooling layers on top of each other in a sequential structure. The authors of the paper also emphasized that this new model places notable consideration on memory and power usage.

VII. Inception Module

When we first take a look at the structure of GoogLeNet, we notice immediately that not everything is happening sequentially, as seen in previous architectures. We have pieces of the network that are happening in parallel.

This box is called an Inception module.

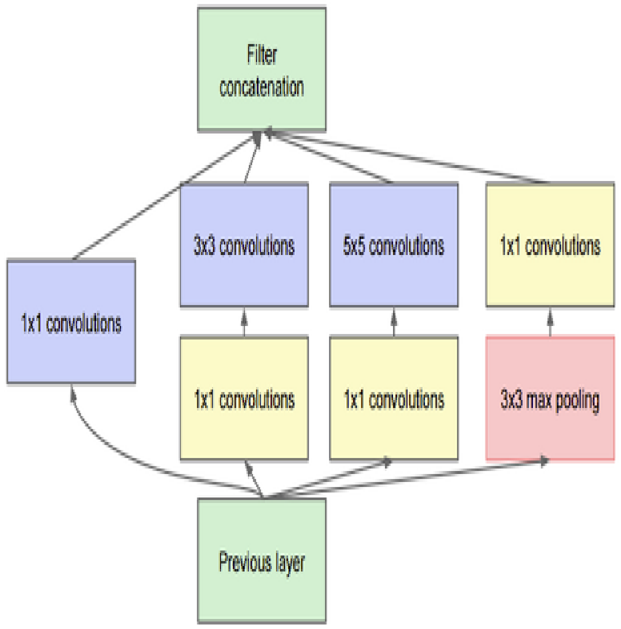

Fig 3: Inception module

The bottom green box is our input and the top one is the output of the model (Turning this picture right 90 degrees would let one visualize the model in relation to the last picture which shows the full network). Basically, at each layer of a traditional ConvNet, we have to make a choice of whether to have a pooling operation or a conv operation (there is also the choice of filter size). What an Inception module allows us to do is perform all of these operations in parallel.

VIII. Training

Kaggle dataset which consists of (45x45) .jpg images of handwritten digits and mathematical operators is used for training the CNN. It contains around 10,000 samples for each symbol.

A sample dataset is shown here:

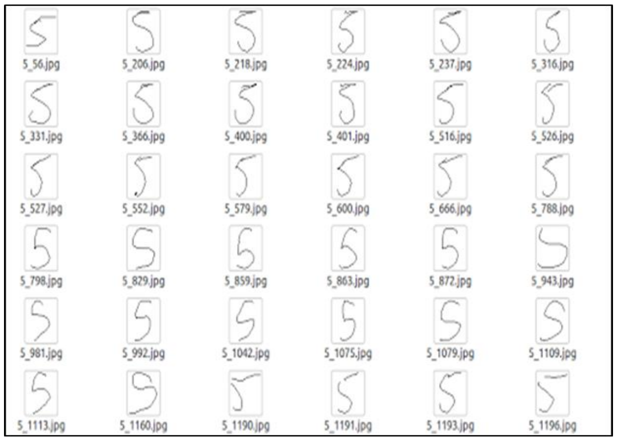

Fig 4: A sample dataset for the digit '5'

IX. RESULTS

Some images of handwritten equations were tested and it successfully gave the correct result.

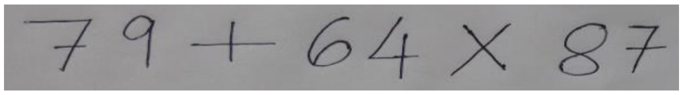

Fig:5: Test image of a Handwritten expression

X. CONCLUSION

In this paper we have discussed the implementation of a Image based calculator. We have illustrated the basic building blocks of the project. Improvement in recognition of 1 and / is required. Inclusion of special characters (log, ln, x, y, derivatives, integrals) can be considered in the next version

ACKNOWLEDGMENT

The authors wish to acknowledge deep sense of gratitude to Dr. Sumohana Channappayya, Associate Professor, IIT-Hyd, for his encouragement and support to carry out this work. The authors express their gratitude to CPS Lab, IIT-Hyd for allowing us to use their GPUs for training the CNNs.

REFERENCES

[1] http://yann.lecun.com/exdb/mnist/

[2] http://mccormickml.com/2013/08/15/radial-basis-function-network-rbfn-tutorial/

[3] https://www.kaggle.com/xainano/handwrittenmathsymbols

[4]http://www.kdnuggets.com/2016/09/9-key-deep-learning-papers-explained.html/2