

# Project Documentation

**Project Title:-** Text Recognition

## Team Members:-

- Arpit Agarwal
- Jaskeerat Singh
- Vikrant Singh
- Piyush Singla

## Abstract:-

We wanted to make a project capable of reading text from an image and subsequently process the data extracted. In our case, Image is a printed system of linear system of equation in two variables, we take a photo of these equations and display it on the screen along with the solution of the above equations. It have various applications in everyday life like extracting valuable information from visiting cards, ATM cards etc., getting information directly from application forms. It saves a lot of time and manual labour.

## MOTIVATION:-

As we can see in our daily lives, people take images of some documents when they have no other source to take that document with them, but later they have to read each and every word from it. So we thought to make a project in which we just take an image and process it to extract the text present in the image. It saves a lot of time to read the text from an image.

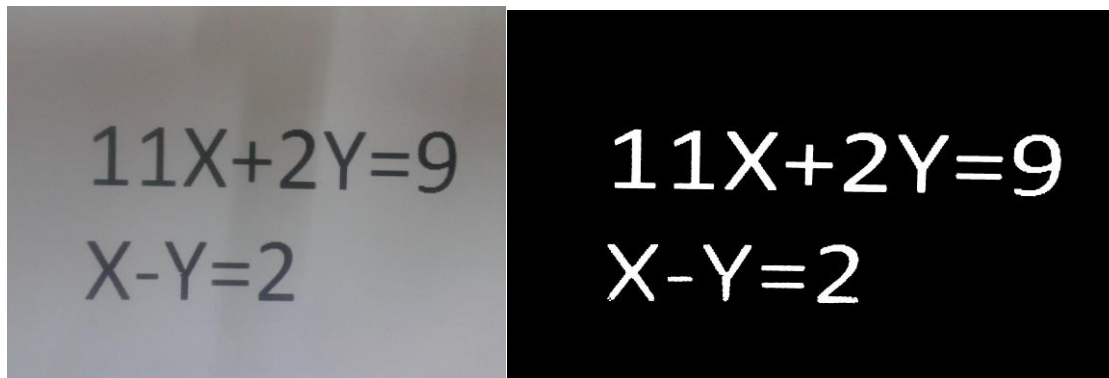
## SOFTWARE REQUIRED:-



We used opencv library 2.1 configured with DevC++4.9 .

## IMPLEMENTATION:-

To start with our project , first we loaded the RGB image and then we converted the image into a greyscale image and finally converted it into a binary image

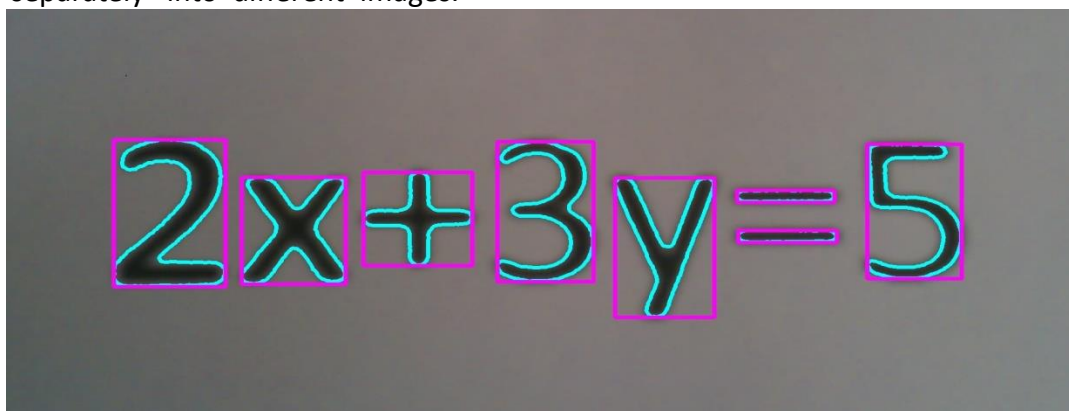


GrayScale Image

Binary Image

but the image that was coming contained a lot of noise .To reduce the noise we used morphological operations of dilation and erosion .The result was the kind of the above binary image that we got.

The next task was of cropping the image .This was done using the method of contours . A contour is a list of points that represent , in one way or another , a curve in an image. This representation can be different depending on the circumstance at hand .Contours are represented by sequences in which every entry in the sequence encodes information about the location of the next point in the curve. Then we made bounding boxes around the contours and all the contours were saved separately into different images.



The next task was to identify the characters in the image. We used fuzzy recognition system in which every character has a unique signature and signature is decided by checking the value of the pixels at the edge centres and the corners of

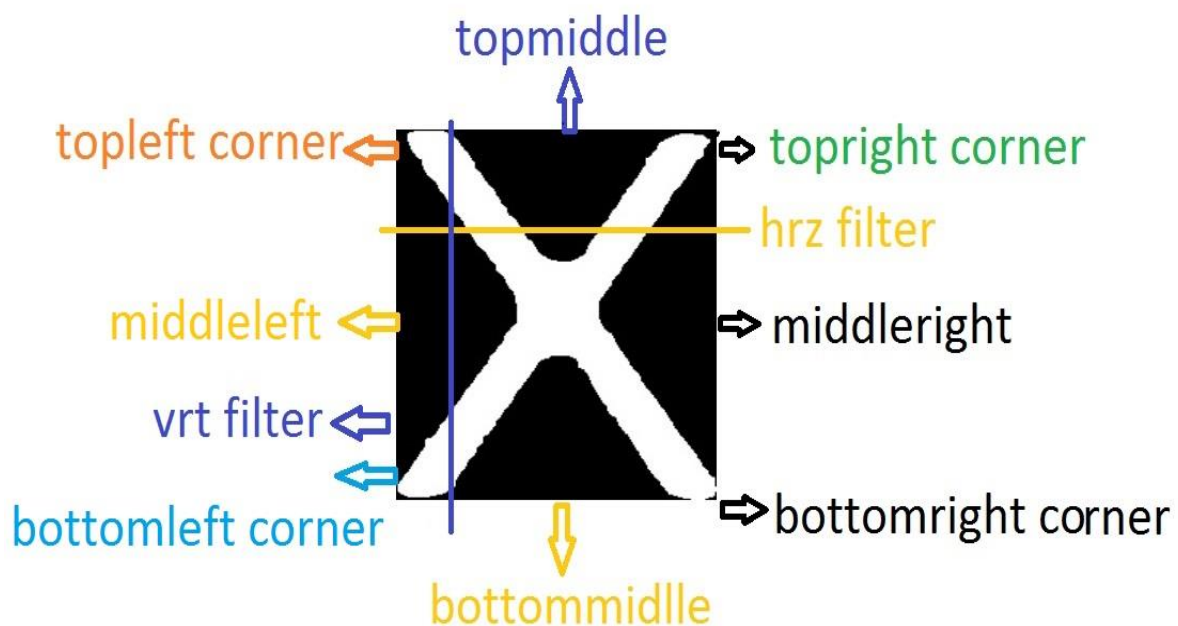
the image and also horizontal and vertical filters. For example, the signature of 'X' is{ Topleft corner(1), TopRight Corner(1), BottomRight Corner(1), BottomLeft Corner(1), Middle Left(0), MiddleRight (0),TopMiddle(0), BottomMiddle(0), Vertical filter(0) ,Horizontal filter(0) }, where 0 stands for that particular region is black and 1 stands for white. We also use the number of holes in the character to divide all the alphabets in three groups. For example

no. holes=0(C,E,F,G,H,I,J,K,L,,M,N,S,T,U,V,W,X,Y,Z,1,2,3,5,7,)

no. holes=1(A,D,O,P,Q,R,0,4,6,9)

no. holes=2(B,8)

then find the highest match with given signatures.



The Final work that was left was to solve the system of linear equations in two variables. This task was very simple and was done by writing a simple C code and finally solutions were displayed on the computer screen.

**Load the original image( RGB image) into the code.**

```
graph TD; A[Load the original image( RGB image) into the code.] --> B[Convert that image into Grayscale image and Grayscale Image into binary using cvThreshold function and filter the binary image using morphological functions like cvErode,cvDilate,cvSmooth .]; B --> C[Crop the image using method of contours.By using cvFindContours store the structure of contours for every character.]; C --> D[Using that structure of contours of each character find the bounding box of each character (Length and bradth of box were parallel to the X-axis and Y-axis) by passing pointer to that structure of corresponding character to cvBoundingRect function.]; D --> E[To recognise the image use the Fuzzy Recognition System. Till now every character is stored in bounding box. Find the signature corresponding to each character . Compare this signature from stored signatures. Character would be decide by the Highest Matching]; E --> F[TEXT RECOGNITION DONE];
```

Convert that image into Grayscale image and Grayscale Image into binary using cvThreshold function and filter the binary image using morphological functions like cvErode,cvDilate,cvSmooth .

Crop the image using method of contours.By using cvFindContours store the structure of contours for every character.

Using that structure of contours of each character find the bounding box of each character (Length and bradth of box were parallel to the X-axis and Y-axis) by passing pointer to that structure of corresponding character to cvBoundingRect function.

To recognise the image use the Fuzzy Recognition System. Till now every character is stored in bounding box. Find the signature corresponding to each character . Compare this signature from stored signatures. Character would be decide by the Highest Matching

**TEXT RECOGNITION DONE**

## **UTILITIES:-**

This project has many utilities in our day-to-day life like it can be used as a doc to pdf converter, as scanner to read the handwritten text and convert it into typed text, in banking purposes and many other applications listing a few of them above.

## **FURTHER IMPROVEMENTS:-**

It can be further extended to identify handwritten text from the images and also an Android App can also be made based on the same principle .

## **A WORD OF THANKS:-**

We extend a special word of thanks to Shivendu Bhushan for helping us throughout the project and providing time-to-time guidance to us.

