

Intel Ocr

User Guide



November 12, 2019

BOSCH Group

7th Block, Koramangala, Bangalore (IN)

# Index

|  |  |
| --- | --- |
| 1. Problem Statement | 1 |
| 1. Work Flow | 2 |
| 1. Web Application | 3 |
| 1. Color Coding | 6 |
| 1. Features | 6 |
|  |  |
|  |  |
|  |  |

1

# Problem Statement

The overall idea behind this project is to develop an computer vision algorithm along with solution package for recognizing and digitizing steps of solving a mathematical equation written by freehand on a paper, validating the steps and final answer of the recognized handwritten lines by maintaining the context

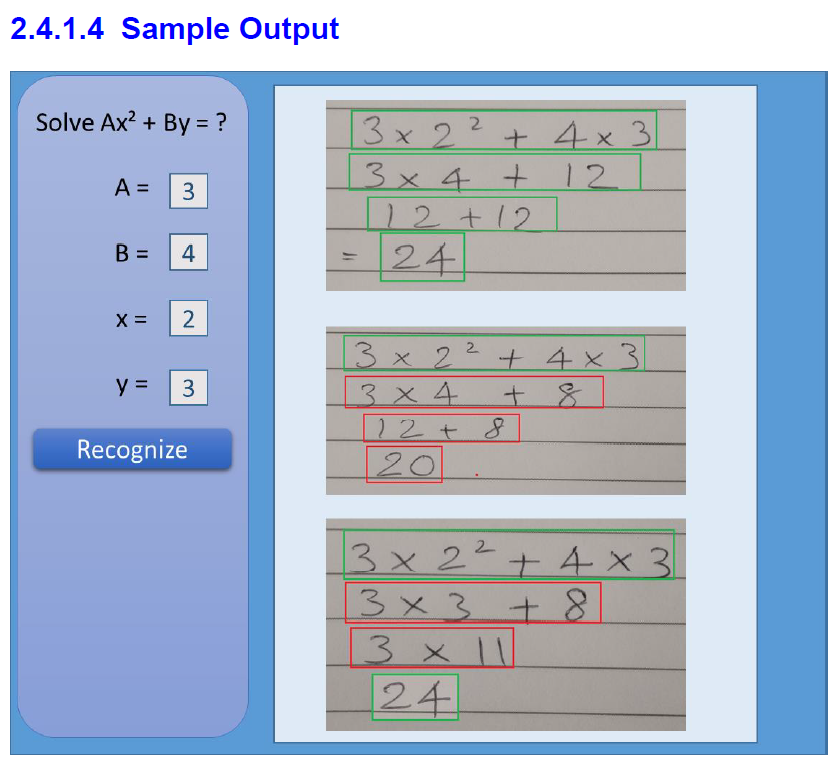


Figure 1 Sample Image Output

2

# Work Flow

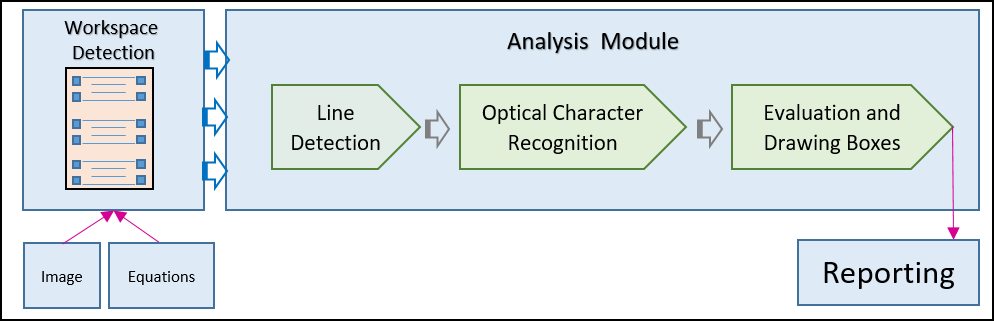


Figure 2 Workflow of the application

Figure 2 describes the overall workflow of the application. Input required for application to run is the handwritten images containing mathematical texts and values of the variables A, B, X, & Y based on the equation :

(1)

After a valid image followed by required variables are uploaded following steps are performed in sequence :

1. Workspace Detection
2. Line Detection
3. Character Detection
4. Character Recognition and Box coloring

3

# Web Application

Following section describes the web application which is used to run the python code. Steps of running the web application is enlisted below

**Step 1: Open this link :** [**http://13.233.58.89:9444/#/dashboard**](http://13.233.58.89:9444/#/dashboard)

On clicking this link, a web based application will open in your web browser which is Intel OCR application for analyzing handwritten mathematical equation, evaluate them and give color coded output

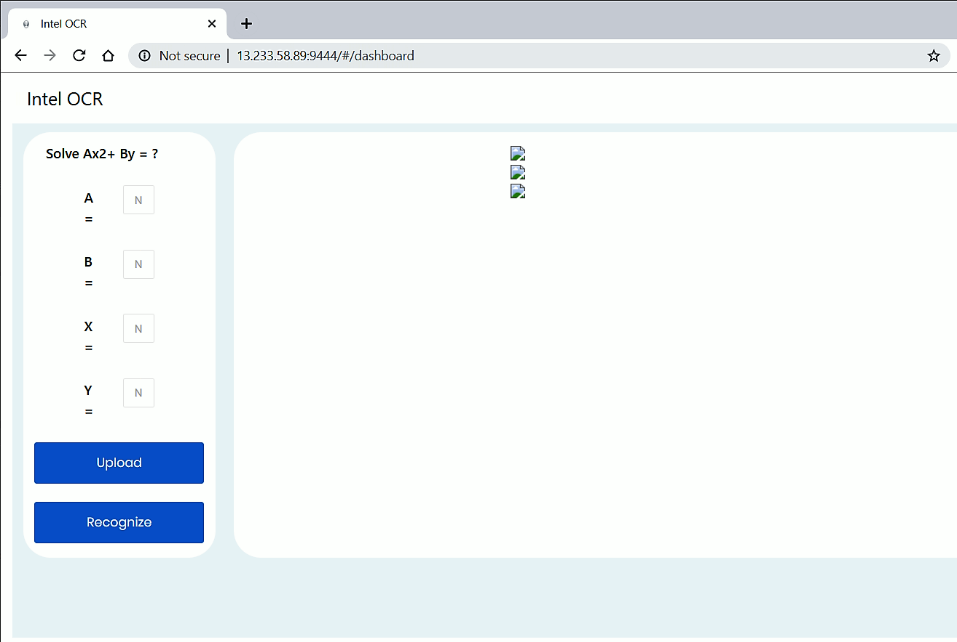


Figure 3 Web Page of Web application

**Step 2: Click on Upload, Select Image to upload**

Second step is to upload an image which has to be analyzed using Intel OCR webapp. For this purpose click on upload, on clicking the upload button File Explorer should open, post which you have to select the required image and note down values of A,B,X,Y variables

4

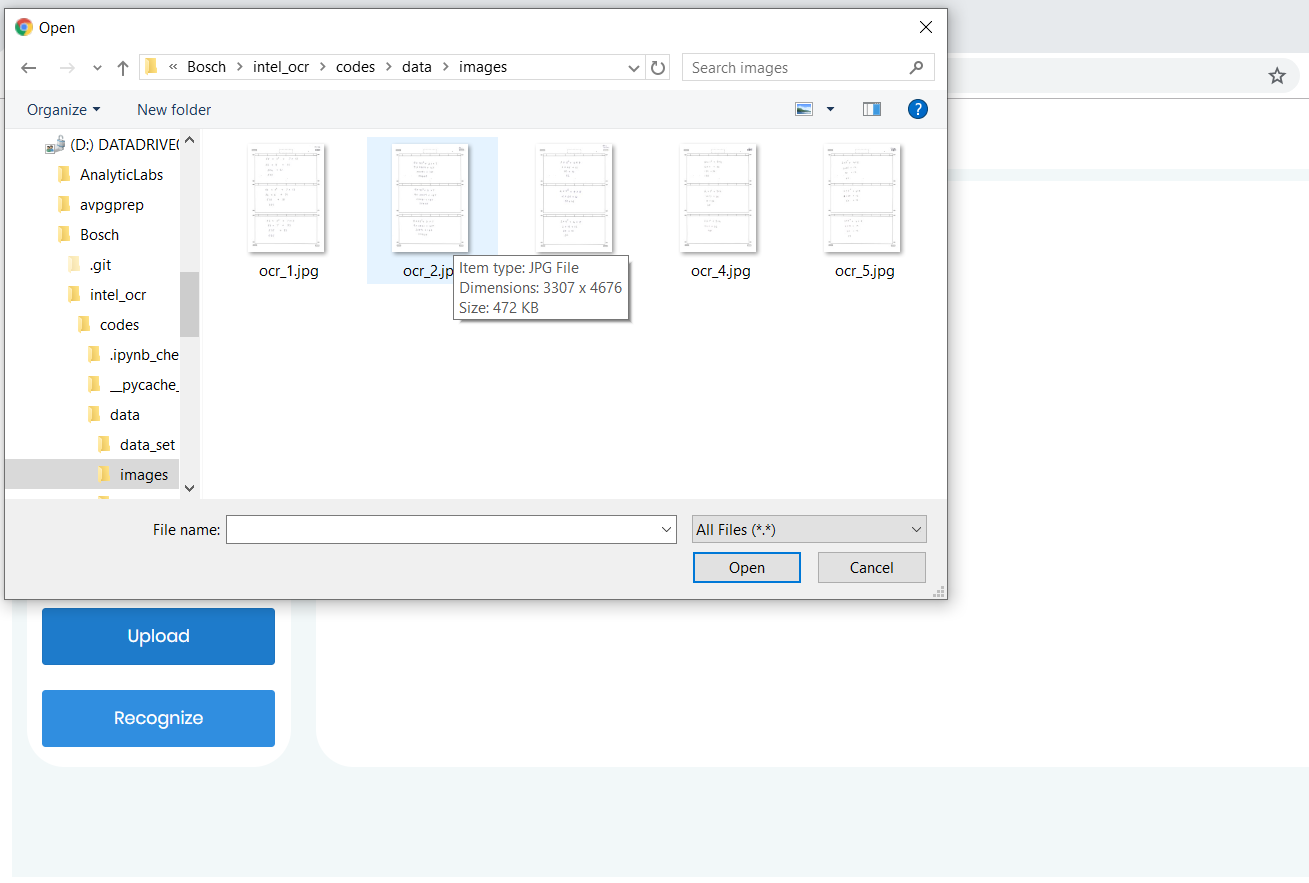


Figure 4 Describes step 2

**Step 3: Fill Values of A, B, X, Y and then click Recognize**

Now fill the values of A, B, X and Y in the corresponding boxes which belong to the corresponding image uploaded. After image upload and values insertion click on Recognize button.

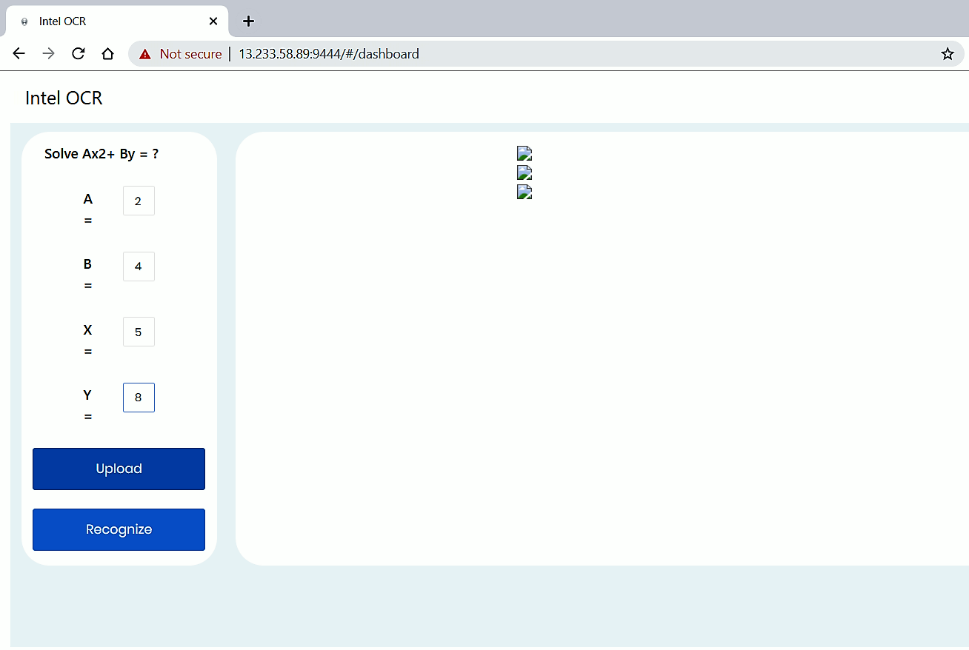


Figure 5 Describes step 3

5

**Step 4: Get answer**

After clicking the Recognize button, wait for 5-10 seconds for results to appear. This waiting time may vary depending on the image size and network speed

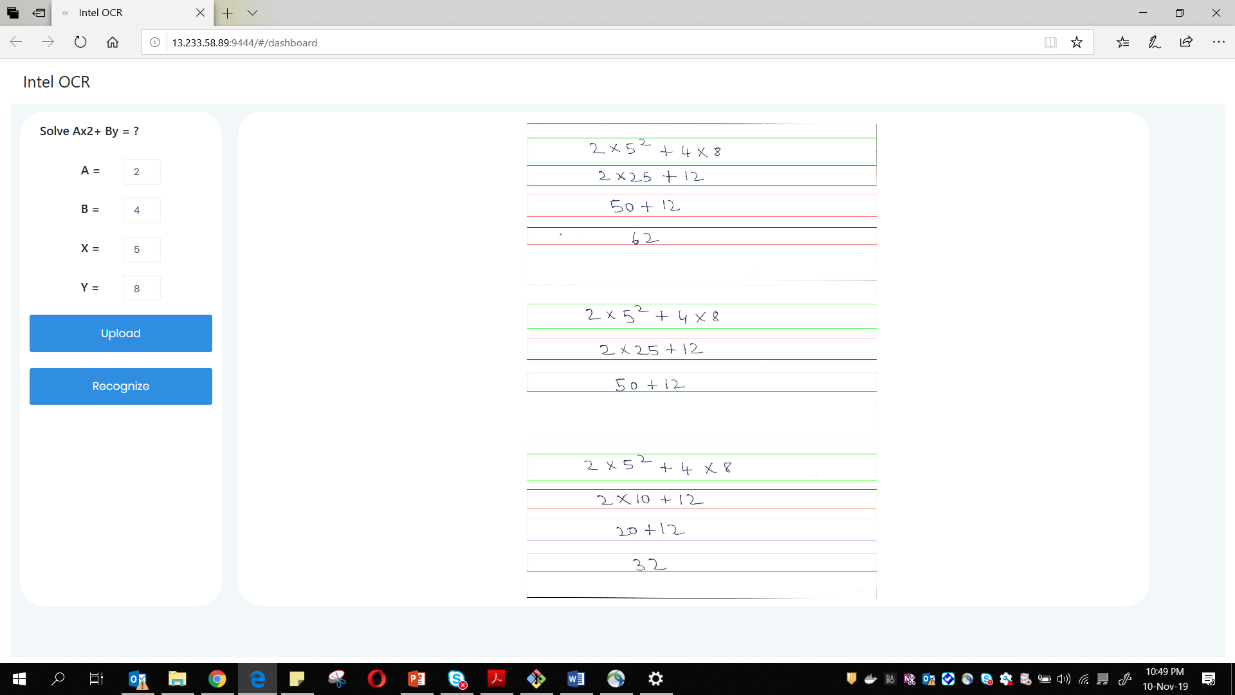


Figure 6 Describes final output

## **About Web Application:**

6

# Color Coding

**Red Box:**Red box signifies that, after analyzing the line its found to be incorrect and there is some mistake in that line

**Green Box:**Green box signifies that, after analyzing the line its found to be correct

**Blue Box:**A line which is enclosed by blue box signifies that the status of line is undetermined and it can be both correct  
or wrong.

# Features

* Accuracy on handwritten images is 98 % for intermediate steps and 93.5 % for Line detection
* Sample application takes images as a input and provides images a output with right, wrong and undetermined colored bounding boxes. Image output is being saved in ‘logs’ folder with respective image name and timestamp. Prediction logs are present in server
* Model file and weights is being saved into ‘models’ folder
* Training files and test results are present in a Jupyter notebook with each steps clearly mentioned
* User guide with steps has been provided
* Hosted solution in Github