**Status: Submitted**

**Approved By: Management representative**

**Released By: QMS Documentation Controller**

Note: Replace the above contents with project specific information

**Internal / Confidential / Strictly Confidential**

**Revision / Document History**

| **Ver.** | **Date** | **Changed by** | **Modifications** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**List of Abbreviations**

1. OCR : Optical Character Recognition
2. AWS : Amazon Web Services

**Table of Contents**

1. Introduction 2

2. Overview 2

2.1. Cautions & Warnings 2

2.2. Copyright Information 2

2.3. License, Liability and Warranty Information 2

3. Product Overview 2

3.1. Package details 3

3.2. Required hardware/environment and software specifications 3

4. Getting Started/Installation 3

5. Using the Product/System/Subsystem 3

5.1. Function/Feature 4

6. Uninstallation/Decommissioning 5

6.1. Exiting the System 5

7. Troubleshooting & Support 5

7.1. Error Messages 5

7.2. Support 6

# Introduction

This is user manual developed by Bosch for the project of Intel named Intel OCR

# Overview

This user manual contains details of the developed end product for the project and will help user to operate the web app developed and obtain results

## Cautions & Warnings

**Important Instructions:**

**Read Carefully:**

Instruction 1: Template to be scanned should be first printed in A4 size of DPI = 400 for best results. Solution will work for other DPI setting as well but aspect ratio of A4 size is the minimum requirement

Instruction 2: There should exist little space between each character and exponentials should be written properly, not farther away from the parent line

Instruction 3: There should exist little space between consecutive lines in any workspace for better recognition

Instruction 4: Rough work should not be done in a workspace which will lead to false detection as a character

## Copyright Information

This project is open source and readily available on github.com

## License, Liability and Warranty Information

The support period after the delivery of the final content for this project is 45 days after formal acceptance by Intel.

# Product Overview

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

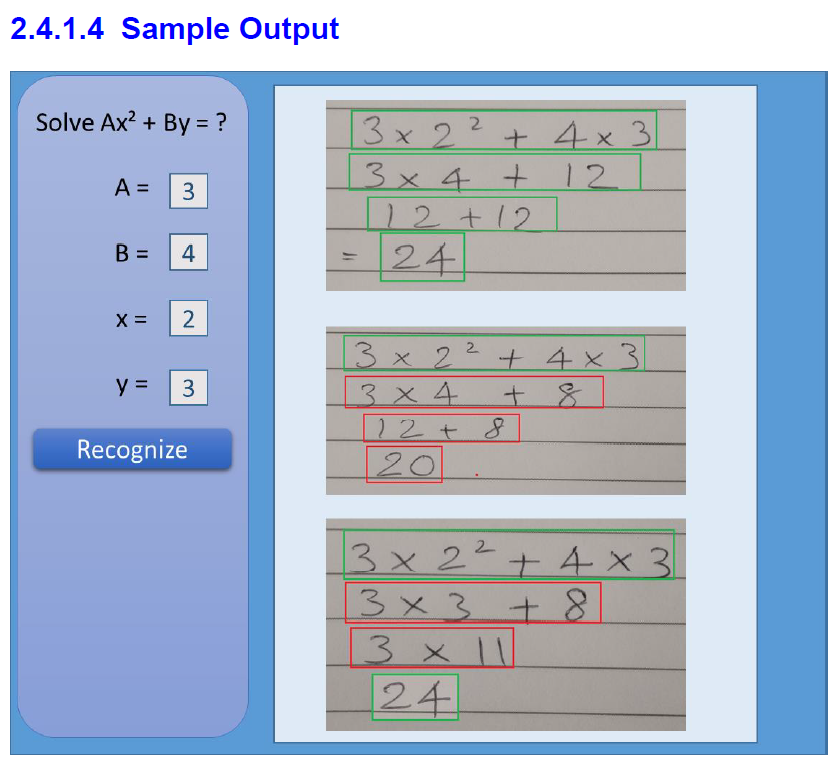


Figure 1 Sample Image Output

## Package details

This section will the details of the python based package delivered to Intel

The package will contain :

1. Docs  
   1.1 User Guide

1.2 PPTs

1. Codes
   1. Data
   2. Logs
   3. Models
   4. Notebooks
   5. Output
   6. UI
   7. Data\_annotater.py
   8. Utils.py
   9. Testing.py
   10. Requirement\_pip.txt

## Required hardware/environment and software specifications

Computer -

CPU - Intel Core i5

RAM - 4 GB RAM ( 8 GB Recommended ),

HDD – 50 GB

AWS Server –

on AWS EC2 Machine with Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz

# Getting Started/Installation

This section deals with the installation instructions that should be considered to start the system/product including the sequence from initiation through exit, installation environment needed.

This project was developed using Anaconda software and tested on AWS. For getting started with the installation process following steps should be followed:

1. Create a separate anaconda / python virtual environment
2. Clone the project from github
3. Activate the virtual environment
4. Run the command in anaconda prompt >> pip install requirements\_pip.txt

Installation on Windows (standalone setup)

Python and Node.JS installation

**Python 3**:

1. Download python3 from official download page (<https://www.python.org/downloads/>)
2. Once the download is complete, install python by launching the setup file and follow the on-screen instructions for installation
3. After the installation is complete, verify it by typing the command – **python --version**

****

**Node.JS**:

1. Download Node.JS from official website - <https://nodejs.org/en/>

2. Once the download is complete, please follow the on-screen installation setup steps.

3. After the installation is complete, verify the installation by typing the command

**node –v**



Creating the Virtual Environment (**Python**)

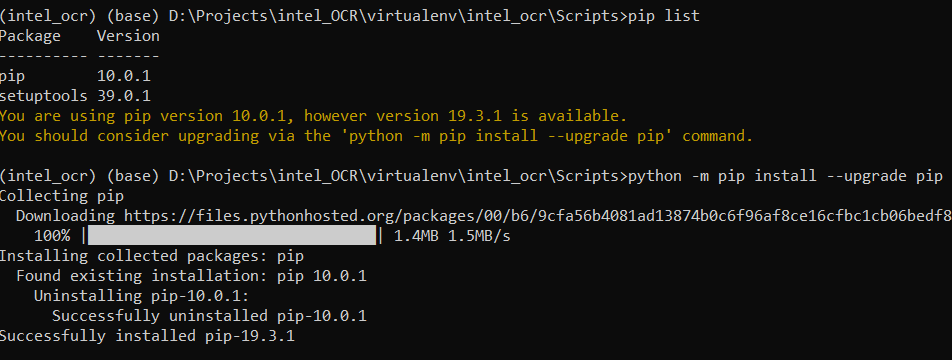
1. Virtual Environment is pre-installed in Python 3
2. We can manually install it with pip if it doesn’t exist using - **pip install virtualenv**
3. Create a new environment by using the below command
4. Activate the environment by using the command

**<venv>\Scripts\activate.bat - <venv> virtual environment name is intel\_ocr**

**command - intel\_ocr\Scripts\activate.bat**

****

1. Update pip before installing the dependencies



Enabling Proxy Connection – **Python**

* If pip install package-name command fails, most likely user is running behind a proxy network, to allow package downloads proxy settings need to be updated by using the below steps
* **Windows 10** –

1. Click on windows icon
2. Go to settings- find environment variables in search box
3. Click on Edit environment variables for your account
4. New window open showing the available environment variables
5. Click on New.. and edit variable name and variable value with values given below and click ok.

http\_proxy = [“http://proxy](http://proxy/)link/”

https\_proxy = [“http://proxy](http://proxy/)link/”

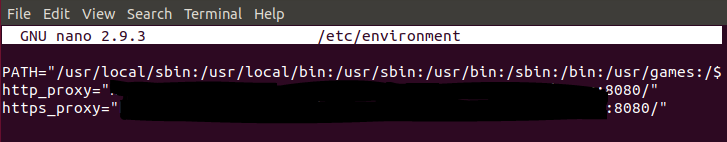
ftp\_proxy = [“http://proxy](http://proxy/)link/”

socks\_proxy = [“http://proxylink/](http://proxylink/)”

1. Exit the command prompt and relaunch the command prompt for changes to get affected.

* **Ubuntu -**

1. Open the terminal
2. Type sudo nano /etc/environment ( Admin Mode – enter password )
3. Add new lines below the PATH variables



1. Enter http\_proxy , https\_proxy , socks\_proxy, ftp\_proxy for allowing package downloads behind the proxy.
2. Save the file and exit the terminal and relaunch the terminal for changes to get affected.

Enabling Proxy Connection – **Node.JS**

* **Windows 10 -**

1. User needs to create a file called .npmrc in the user’s root directory

( C:\Users\\*username\*\.npmrc )

image001

1. Edit the file with any text editor and enter

the below commands.

registry=https://registry.npmjs.org;

proxy= “http://proxylink/”

http-proxy= “http://proxylink/”

https-proxy= [“http://proxylink/](http://proxylink/)”

1. Restart the command prompt for changes to get affected.

* **Ubuntu**

1. Open User’s home directory by entering the below command in terminal

> $HOME

1. Create a new file called .npmrc by using the below command

> touch .npmrc

1. Open the file using any text editor ( nano, vi, gedit e.t.c) and enter the below commands and save the file.

registry=https://registry.npmjs.org;

proxy= “http://proxylink/”

http-proxy= “http://proxylink/”

https-proxy= [“http://proxylink/](http://proxylink/)”

4. Restart the terminal for changes to get affected.

Installing dependencies – **Python**

1. Activate the environment by using the command

**Windows :**

**<venv>\Scripts\activate.bat - <venv> virtual environment name is intel\_ocr**

**intel\_ocr\Scripts\activate.bat**

output : (intel\_ocr) (base) D:\Projects\intel\_OCR\virtualenv>

**Ubuntu :**

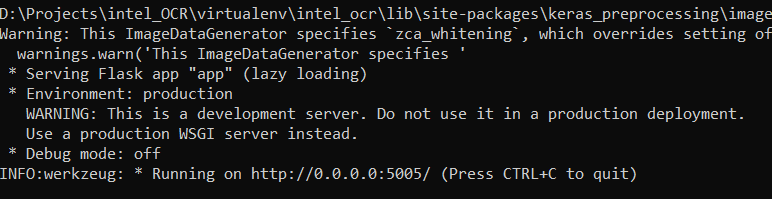
**source env/bin/activate**

output : (env) testuser@localhost:~/python-environment$

1. Install the required dependencies by navigating to the application folder.
2. Find the requirements.txt file/
3. Make sure the virtualenv is active before running the below command

**pip install –r requirements.txt**

1. After the installation is complete, verify the packages by typing the command pip list
2. Run the app.py file in the application folder

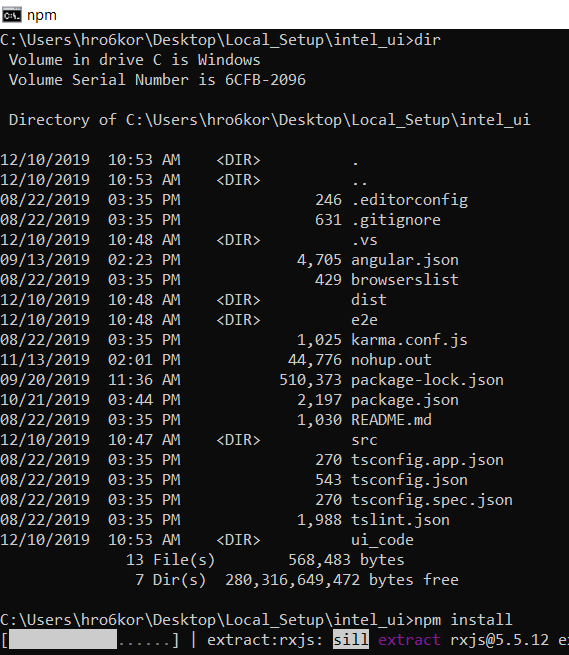


1. API server will run on port 5005.
2. We can test the API’s by using the swagger documentation link

<http://localhost:5005/api/documentation>

Installing dependencies – **Node.JS**

1. Browse to the project folder



1. Confirm whether node is present or not by running the command ( **node -v** )
2. Make sure package.json file in present in the application folder
3. Run the command ( inside project folder )

**npm install**

1. All the required packages will be auto downloaded
2. To run the server, we need to execute the command

**ng serve** or **npm start**

1. After the server has started, we need to open a browser and enter the below URL

<http://localhost:9444/> ( please make sure the API server is running in the backend

before trying out the application from

1. Setup is complete.

# Using the Product/System/Subsystem

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

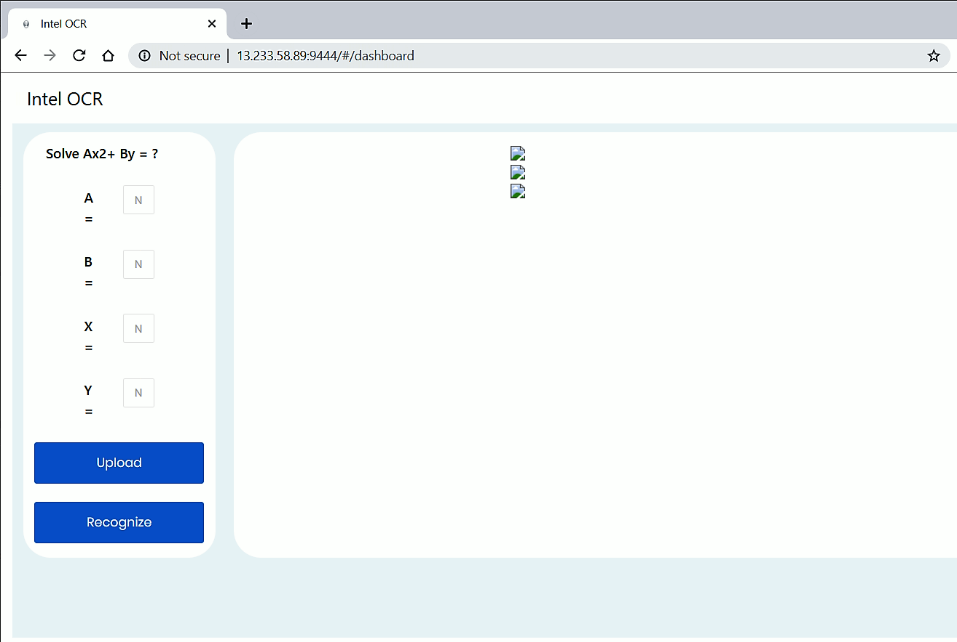
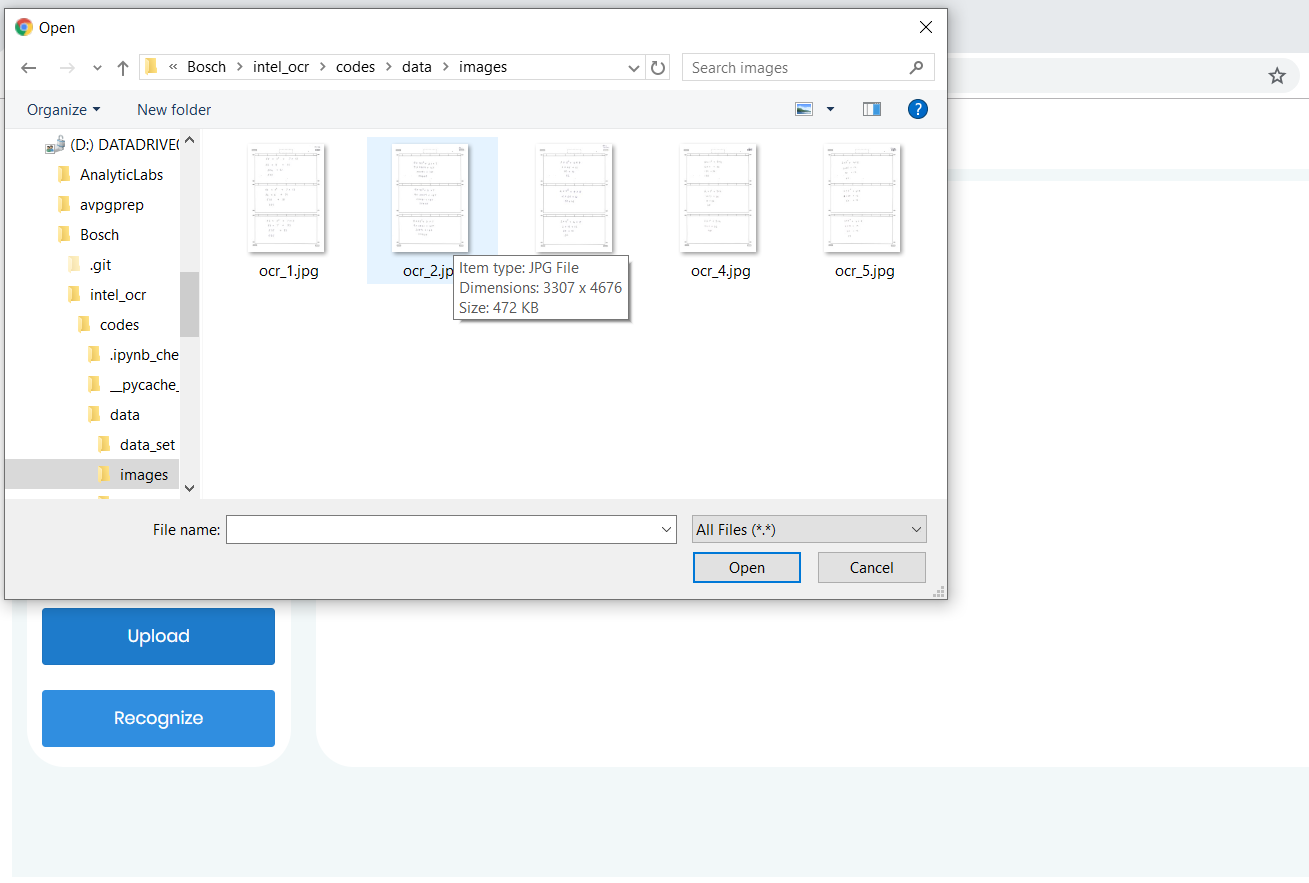


Figure 3 Web Page of Web application

**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

After image is selected and open is clicked, name of the image will start displaying below upload button

Filename : ocr\_2.jpg

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.

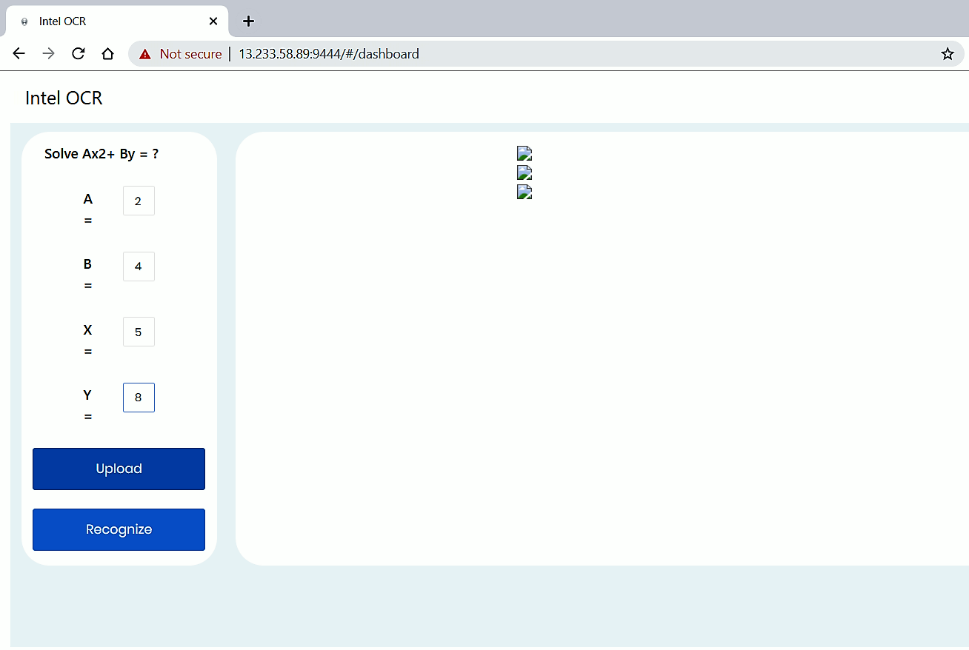


Figure 5 Describes step 3

## Function/Feature

<Describe the specific system function / feature in detail>

Feature 1: Red Box:

Red box signifies that, after analysing the line its found to be incorrect and there is some mistake in that line

Feature 2: Green Box:

Green box signifies that, after analysing the line its found to be correct

Feature 3: 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

# Uninstallation/Decommissioning

This section describes the uninstallation/Decommissioning instructions that should be considered to safely exit/decommission/uninstall the system/product including the sequence from initiation through exit, uninstallation environment needed.

## Exiting the System

Close the browser which you were using to run the web app

# Troubleshooting & Support

<Describe all recovery and error correction procedures, including error conditions that may be generated and corrective actions that may need to be taken. Organize the information in sub-sections as appropriate. >

## Error Messages

< Mention the error messages that a user may receive and the likely cause(s) and/or possible corrective actions for the error>

## Support

<Provide information on how the user can get customer assistance and support (e.g., help desk support, etc.). Include the names of the responsible personnel and telephone numbers, and email addresses who serve as points of contact for support. >

**Template Revision History <To be updated by QMM>**

**Template Revision History**

| **Ver.** | **Date** | **Changed by** | **Modifications** |
| --- | --- | --- | --- |
| 1.0 | 10/11/2019 | Bijon Guha | Initial version |
| 1.1 | 13/11/2019 | Bijon Guha | Modified as per Bosch template requirement |
| 1.2 | 14/11/2019 | Rounak Sharma | Revised for error code |
|  |  |  |  |

<Delete this table when using this template>