Proposal for Intel

OCR using computer vision

Bijon Guha (RBEI/EDS2)

2019

Project Overview and Details

This section outlines the details related to the project background, proposed scope and perceived technical approach towards attaining the end objective of optical character recognition of mathematical equation using computer vision

## 2.1 Background

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

## 2.2 Project Scope and Details

The following are the broad modules which will be catered to, towards the realization of the end objective of the project:

1. Workspace Detection using valid markers in the sheet
2. Detecting and localizing each single lines
3. Perform Optical Character Recognition in each detected line
4. Evaluating each line and Providing feedback in terms of red/green bounding box drawn across it where green represents correct and red represents wrong

* **Estimated Duration**: 2 months from kick-off (timeline to be frozen upon acceptance of proposal)
* **Tentative start date**: ~~<blank>~~
* **Location**: Delivered from Robert Bosch Engineering and Business Solutions, Bangalore
* **Travel**: Currently no travel is planned for the associates involved in the project. (Required travel may be planned on a need basis and billed at actuals)

## 2.3 Functional Scope

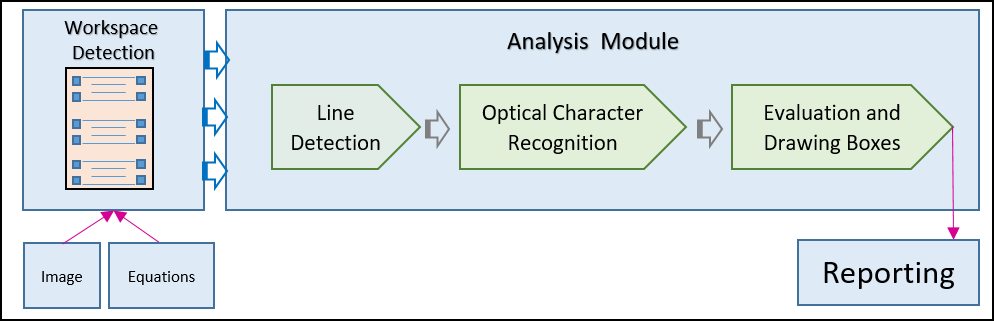
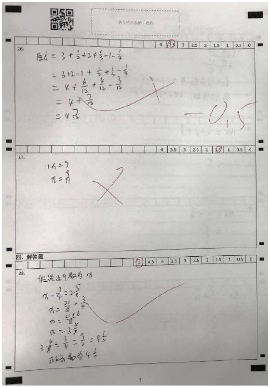
The primary functional blocks of the project are outlined in the block diagram below:   
  


Figure Workflow

As shown the overall solution can be divided into two parts, i.e ‘Workspace Detection’ module and ‘Analysis Module’.

Workspace Detection module is responsible for detecting multiple workspaces in a   
given sheet of paper using pre-defined markers as shown in *image : 1* *Sheet*

Analysis module is responsible for detecting and localizing characters in lines in any   
given single workspace, and mathematically analyzing them and then drawing red,   
green lines depending upon their correctness

Image : Sheet

## 3. Technical Scope

This section deals with Data requirements, Development, testing, quality and deployment approach

## 3.1 Data Requirements

1. Clear scanned image
2. Values of all variables

## 3.2 Approach

This section deals with the approach to generate data, training, testing, and quality

## 3.2.1 Data Generation

Data Generation will take place in the following two phases :-

1. Electronic data generation using publicly available digits and symbols datasets
2. Manually data generation by writing by Bosch employees

## 3.2.2 Training, testing, and Quality

From the complete dataset, 75% data will be used for training, 5% for validation, and 20% for testing.  
Confusion matrix will be used for quality estimation

## 4. Technical Assumption and Dependencies

* Algorithm will be developed based on the images of the publicly available datasets and images captured by Computer Vision lab set up at RBEI premises, there may be a need to validate and fine tune the same for a production environment
* There should be proper workspace detection markers as shown in the sample image of the sheet to identify workspace in every sheet
* There should be sufficient gap between the digits. If two or more digits are separated by space, they will be considered as a single integer
* Multiplication operator is ‘x’ or ‘()’. No division operator will be used.
* Color of the ink is always black
* Only whole numbers are considered for the detection of digits

## 5. Success Measures and Acceptance Criteria

Following will be measured

* Intermediate steps are correctly recognized
* Final answer is correctly recognized.

An accuracy level of >90% should be achieved by the solution where accuracy level is defined in terms of confusion matrix.

## 6. Deliverables

Following are minimum deliverables.

* Sample application in which take image input and provide image output with colored bounding boxes with Right or Wrong indication. Application should also generate detailed log file.
* Model files and weights.
* Training Files and Test Results
* User guide along with steps for integrating in a different application software.

All parts will be hosted on GitHub publicly