Credit Hours System

Cairo University
Faculty of Engineering

GRADES AUTO FILLER
TEAM 13

Submitted to:

- Dr. Mayada Hadhoud
- Eng. Youssef Ghattas
- Eng. Ahmed Maher

Submitted by:

Karim Yasser Ahmed	1152092
Khaled Sameh Mohamed	1153073
Ahmed Hussein	1152259
Mina Ashraf Louis	1152050

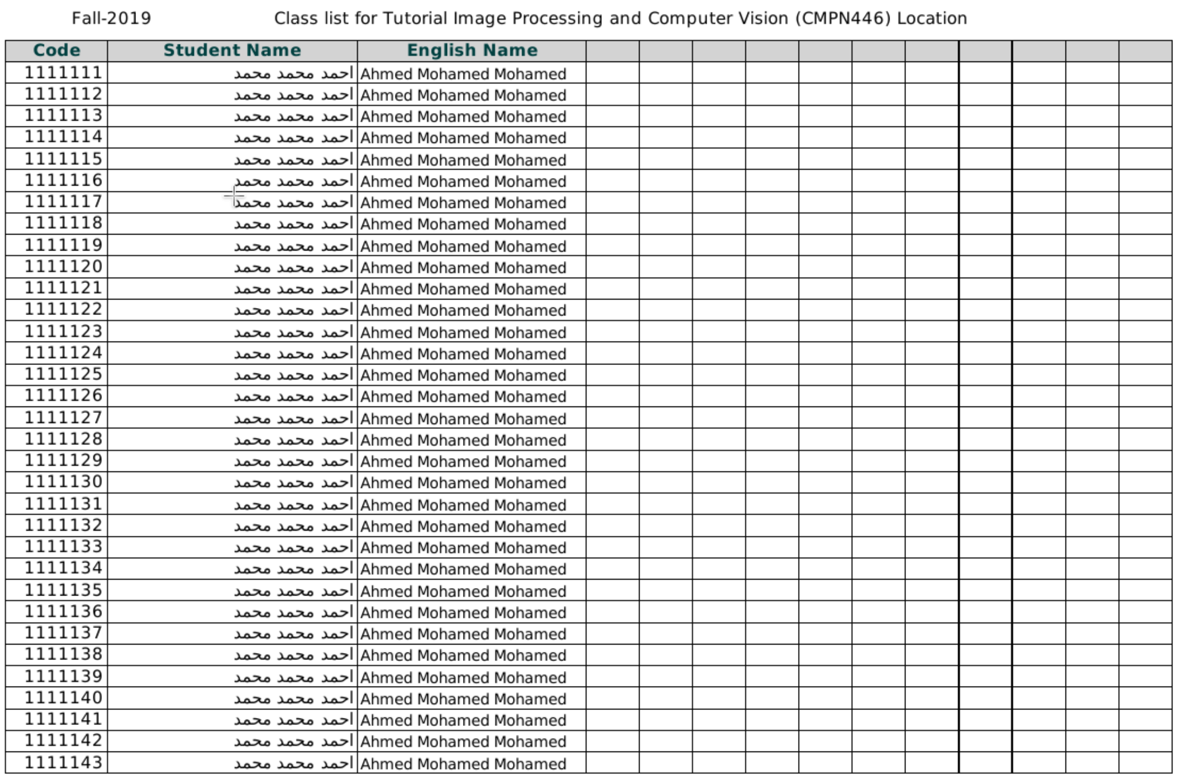
Table of Contents

Grades auto filler

This project is an assistant to TAs and Professors in our department. It should provide an easy way to fill the grades electronically and it should be able to correct MCQ bubble sheet exams automatically.

Module 1 (Grades sheet)

Ordinary grades sheet: in this mode you are required to work on a printed paper of grades' sheet like the following:



Fall-2019 Class list for Tutorial Image Processing and Computer Vision (CMPN446) Location

Code	Student Name	English Name											
1111111	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111112	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111113	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111114	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111115	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111116	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111117	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111118	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111119	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111120	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111121	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111122	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111123	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111124	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111125	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111126	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111127	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111128	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111129	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111130	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111131	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111132	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111133	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111134	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111135	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111136	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111137	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111138	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111139	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111140	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111141	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111142	احمد محمد محمد	Ahmed Mohamed Mohamed											
1111143	احمد محمد محمد	Ahmed Mohamed Mohamed											

This sheet will be filled by TAs or Professor with corresponding grade for each column for each student. Then, a picture taken by mobile camera will be the input to the required system.

Neither the paper nor the photo will be in a perfect state. You should deal with:

1. Different angles of capturing (Skewing, orientation, scale) but no upside down.
2. Different ink colors (or clear pencils).

3. Different format for the sheet (for example: Different sizes for rows and columns and so on).
4. Different hand-writing filling.
5. Different number of students.
6. And so on.

Hint: this should be easy as you are always sure there will be a table.

Your output should be an excel sheet that contains a sheet similar to the one (or you can take the original sheet and just fill it whatever easier for you).
The following data should be converted to text:

1. Printed Student ID: (should be implemented using (1) already-made OCR - (2) features + classifier. User should choose method 1 or 2 before processing).
2. The following written symbols (no ocr can be used):

   a. ✓ (output should be 5)
   b. □ (output should be 0)
   c. **-** (output should be 0)
   d. **Empty cell** (output should be empty cell)
   e. **Stacked Vertical lines |||** in the cell (but they won't be perfectly vertical as they are hand-written) [output should be i where i is the number of lines]
   f. **Stacked Horizontal lines** - in the cell (but they won't be perfectly horizontal as they are hand-written) [output should be (5 - i) where i is the number of lines].
   g. **?** (output should be empty cell with red background color).

   ⇒ a,b,c,d should be as accurate as possible.
3. Numeric written values (should be implemented using (1) already-made OCR - (2) features + classifier. User should choose method 1 or 2 before processing).

The project structure consists on several modules including:

1. Segmentation functions
2. Read and extract paper from image
3. Extract the table
4. Extract student details
5. Extract ids & convert to text
6. Get grades cells count
7. Arrange grades contours & extract them
8. Distribute data per student
9. Classifier functions
10. Training classifier section
11. Writing results to excel sheet

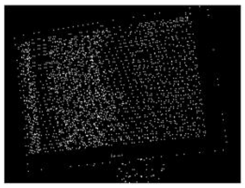
** MOST OF PREVIOUS MODULES INCLUDE PREPROCESSING**

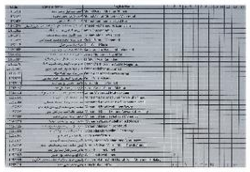
Used Algorithms:

1. Otsu Thresholding
2. Four Point Transform
3. findContours & contourArea using OpenCV
4. GaussianBlur
5. Canny for Edge Detection
6. Bounded Rotate Image from IMUTILS
7. Bounding Rectangle and Triangle using OpenCV
8. Image_to_string from Pytesseract to extract IDs
9. Built in Sorting & Reverse list algorithms
10. KNN Classifier built by us
11. Euclidean Distance Calculation built by us
12. The Classification Algorithms:
    a. For the right mark, square and minus sign we use the KNN classifier based on minimum bounding rectangle and triangle and using a training set
    b. For The Horizontal Lines, Vertical Lines, Empty, and Question Mark we use the characteristics of the histogram projection to detect the symbol.

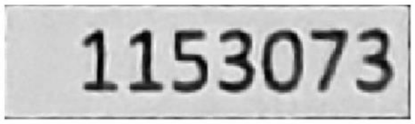
Project Stages Results:

1. Read and extract paper from image & Extracting the table

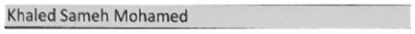




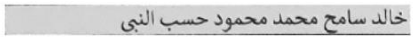
2. Extract student details


1153073

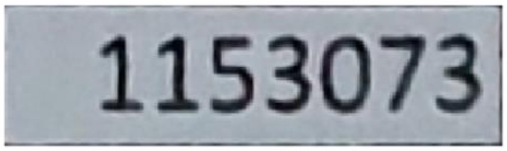
```
contour  79 =  55468.5
```


Khaled Sameh Mohamed

```
contour  80 =  40869.0
```

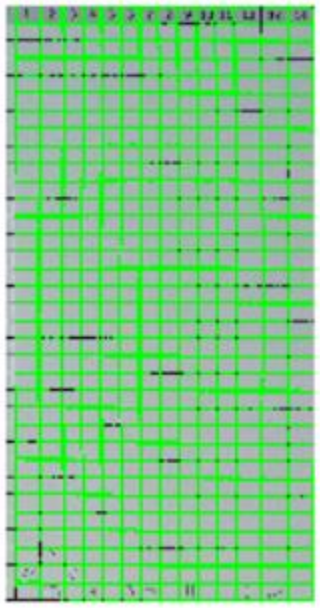

خالد سامح محمد محمود حسب النبي

3. Extract ids & convert to text


1153073

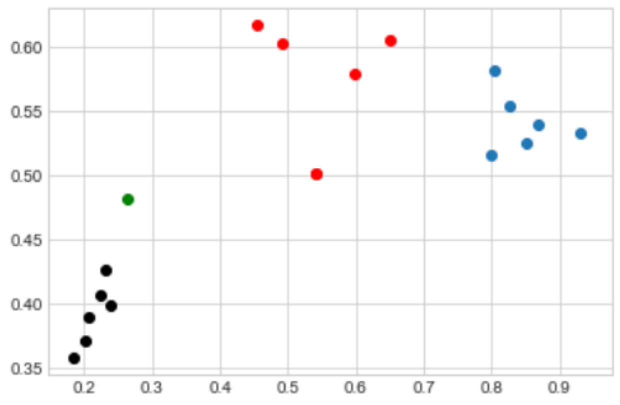
```
ID =>  1153073
```

4. Arrange grades contours & extract them

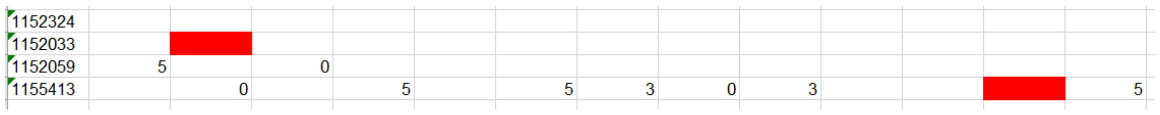


5. Training classifier section

```
[<matplotlib.lines.Line2D at 0x1d00040e978>]
```



6. Writing results to excel sheet



1152324													
1152033		🟥											
1152059	5		0										
1155413		0		5		5	3	0	3			🟥	5

Work Division

Ahmed Hussein:
60% of Segmentation, 20% Classification, Writing To Excel Sheet
Karim Yasser:
40% of Segmentation, 80% Classification

Accuracy & Performance

1. Printed Student ID: Using Tesseract the output is almost 99% correct
2. The following written symbols:
    a. ✓ (output should be 5) =95% correct
    b. □ (output should be 0) =95% correct
    c. **-** (output should be 0) =95 % correct
    d. **Empty cell** =98% correct
    e. **Stacked Vertical lines** =100% correct
    f. **Stacked Horizontal lines** =100% correct
    g. **?** =100% correct
3. Numeric written values = Failure
   Unfortunately, we got a library to detect handwritten numbers but we couldn't use it till now. The Reference to it:
   https://github.com/pavitrakumar78/Python-Custom-Digit-Recognition

Limitations

1. The paper must be full in the photo, meaning the 4 corners must be included
2. The character in the grading box MUST NOT touch the lines
3. The algorithm works in Landscape orientation and only if it is rotated, portrait orientation, 90 degrees anti-clockwise but not clockwise.
4. Sometimes the detection reads the rubbed symbol from paper

References

1. openCV documentation
2. python documentation
3. Imutils documentation
4. XLS documentation
5. Plot documentation
6. Stack overflow
7. https://www.pyimagesearch.com/2014/08/25/4-point-opencv-getperspective-transform-example/
8. https://github.com/pavitrakumar78/Python-Custom-Digit-Recognition