Optica

By:
Faiez Malik 19i2194
Syed Abubakr 19i2199

Optical-Mark-Recognition

This project conatains a OMR program using openCV and matplotlib. Main concepts of Image processing that were used are canny egde detection, adaptive thresholding and image contouring. All of these functionalities are present in the openCV library of python.

File Structure

The repo contains an image folder that are the dataset for the program. The program works on the mentioned dataset only. There is a .py, .ipynb that gives the same output. The difference is just in extension and that the py file contains tkinter for GUI and py file conatins a driver function. There is also an executable file in the 'Executable-app' folder. ** Note: ** The exe file contains path for the background image that is located in the root directory of the folder.

how to use (.ipynb)

Place an image in the same directory of the code. Give the path of the image in the second bloq and start the process.Run the code using Visual Studio Code with a openCV and a python interpreter installed.

how to use (.py)

The py file contains the tkinter library and has a GUI. Hence running the code simple with VS code would be self-explanatory

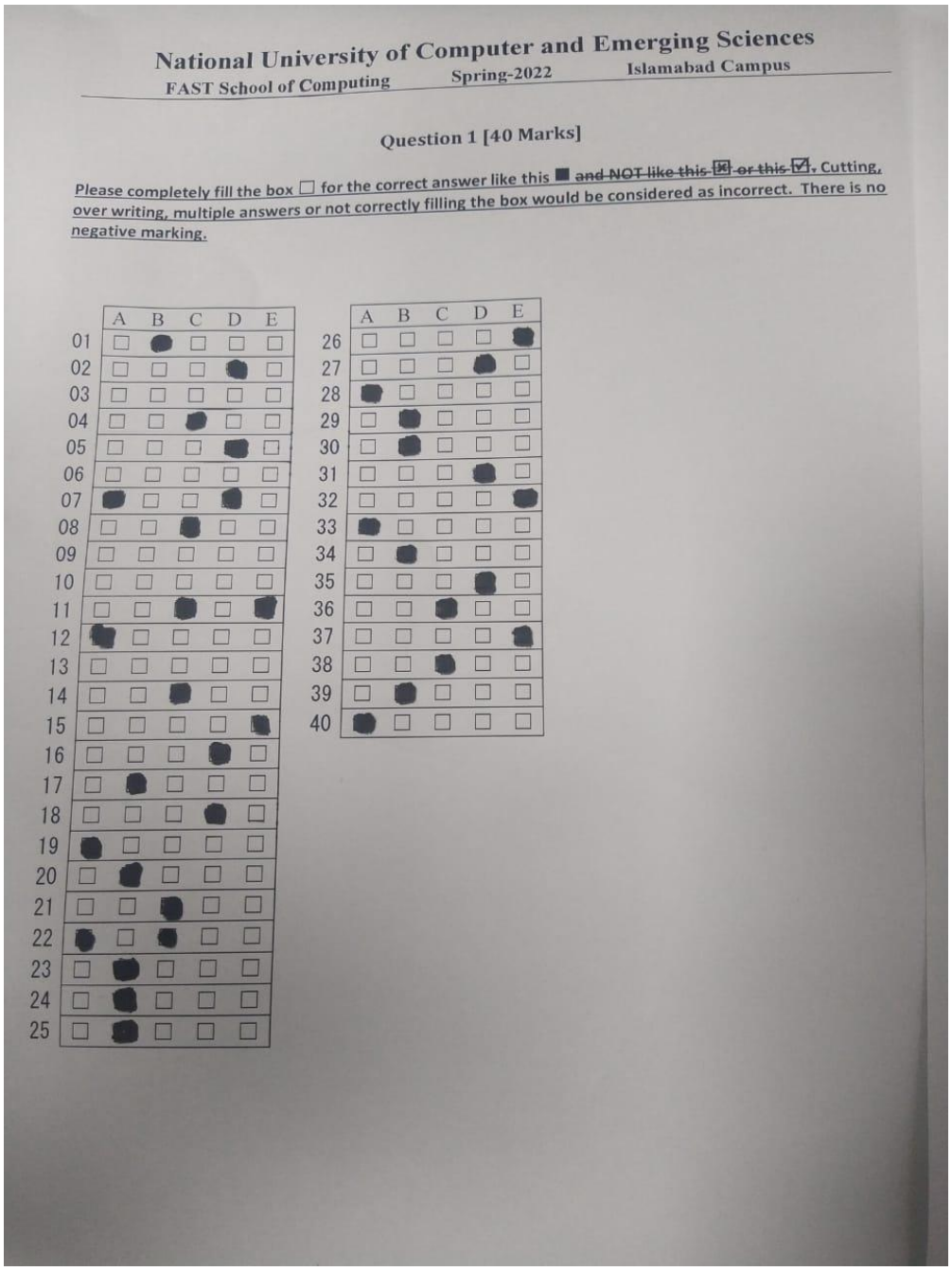
how to use (.exe)

The exe file is converted from the py file through pyinstaller. The output should be same. Note: The exe will give an error if the bgf.jpg img is not placed in the said directory.

Dataset:

Considering we recognized the problem in our university, we used another university's pre-existing dataset of multiple choice questions answer sheets to use as our input images. We would use these images as our sample input and write our program in such a way that it can use these images and produce the desired output.

The sample dataset for some of the pictures can be seen below:



National University of Computer and Emerging Sciences

FAST School of Computing Spring-2022 Islamabad Campus

Question 1 [40 Marks]

Please completely fill the box ☐ for the correct answer like this ■ ~~and NOT like this ☒ or this ☑~~, Cutting, over writing, multiple answers or not correctly filling the box would be considered as incorrect. There is no negative marking.

	A	B	C	D	E
01	☐	■	☐	☐	☐
02	☐	☐	☐	■	☐
03	☐	☐	☐	☐	☐
04	☐	☐	■	☐	☐
05	☐	☐	☐	■	☐
06	☐	☐	☐	☐	☐
07	■	☐	☐	■	☐
08	☐	☐	■	☐	☐
09	☐	☐	☐	☐	☐
10	☐	☐	☐	☐	☐
11	☐	☐	■	☐	■
12	■	☐	☐	☐	☐
13	☐	☐	☐	☐	☐
14	☐	☐	■	☐	☐
15	☐	☐	☐	☐	■
16	☐	☐	☐	■	☐
17	☐	■	☐	☐	☐
18	☐	☐	☐	■	☐
19	■	☐	☐	☐	☐
20	☐	■	☐	☐	☐
21	☐	☐	■	☐	☐
22	■	☐	■	☐	☐
23	☐	■	☐	☐	☐
24	☐	■	☐	☐	☐
25	☐	■	☐	☐	☐

	A	B	C	D	E
26	☐	☐	☐	☐	■
27	☐	☐	☐	■	☐
28	■	☐	☐	☐	☐
29	☐	■	☐	☐	☐
30	☐	■	☐	☐	☐
31	☐	☐	☐	■	☐
32	☐	☐	☐	☐	■
33	■	☐	☐	☐	☐
34	☐	■	☐	☐	☐
35	☐	☐	☐	■	☐
36	☐	☐	■	☐	☐
37	☐	☐	☐	☐	■
38	☐	☐	■	☐	☐
39	☐	■	☐	☐	☐
40	■	☐	☐	☐	☐

National University of Computer and Emerging Sciences

FAST School of Computing Spring-2022 Islamabad Campus

Question 1 [40 Marks]

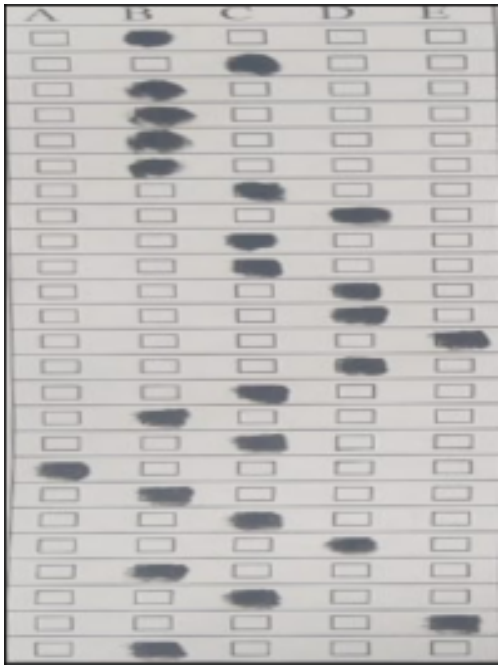
Please completely fill the box ☐ for the correct answer like this ■ ~~and NOT like this ☒ or this ☑~~, Cutting, over writing, multiple answers or not correctly filling the box would be considered as incorrect. There is no negative marking.

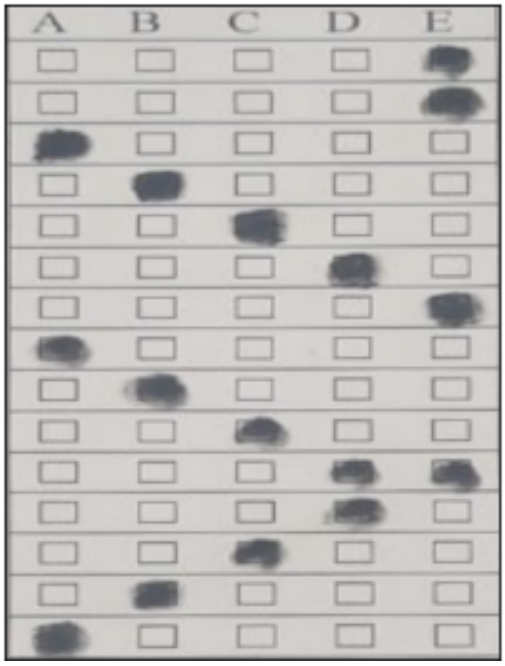
	A	B	C	D	E
01	☐	■	☐	☐	☐
02	☐	☐	■	☐	☐
03	☐	■	☐	☐	☐
04	☐	■	☐	☐	☐
05	☐	■	☐	☐	☐
06	☐	■	☐	☐	☐
07	☐	☐	■	☐	☐
08	☐	☐	☐	■	☐
09	☐	☐	■	☐	☐
10	☐	☐	■	☐	☐
11	☐	☐	☐	■	☐
12	☐	☐	☐	■	☐
13	☐	☐	☐	☐	■
14	☐	☐	☐	■	☐
15	☐	☐	■	☐	☐
16	☐	■	☐	☐	☐
17	☐	☐	■	☐	☐
18	■	☐	☐	☐	☐
19	☐	■	☐	☐	☐
20	☐	☐	■	☐	☐
21	☐	☐	☐	■	☐
22	☐	■	☐	☐	☐
23	☐	☐	■	☐	☐
24	☐	☐	☐	☐	■
25	☐	■	☐	☐	☐

	A	B	C	D	E
26	☐	☐	☐	☐	■
27	☐	☐	☐	☐	■
28	■	☐	☐	☐	☐
29	☐	■	☐	☐	☐
30	☐	☐	■	☐	☐
31	☐	☐	☐	■	☐
32	☐	☐	☐	☐	■
33	■	☐	☐	☐	☐
34	☐	■	☐	☐	☐
35	☐	☐	■	☐	☐
36	☐	☐	☐	■	■
37	☐	☐	☐	■	☐
38	☐	☐	■	☐	☐
39	☐	■	☐	☐	☐
40	■	☐	☐	☐	☐

Methodology:

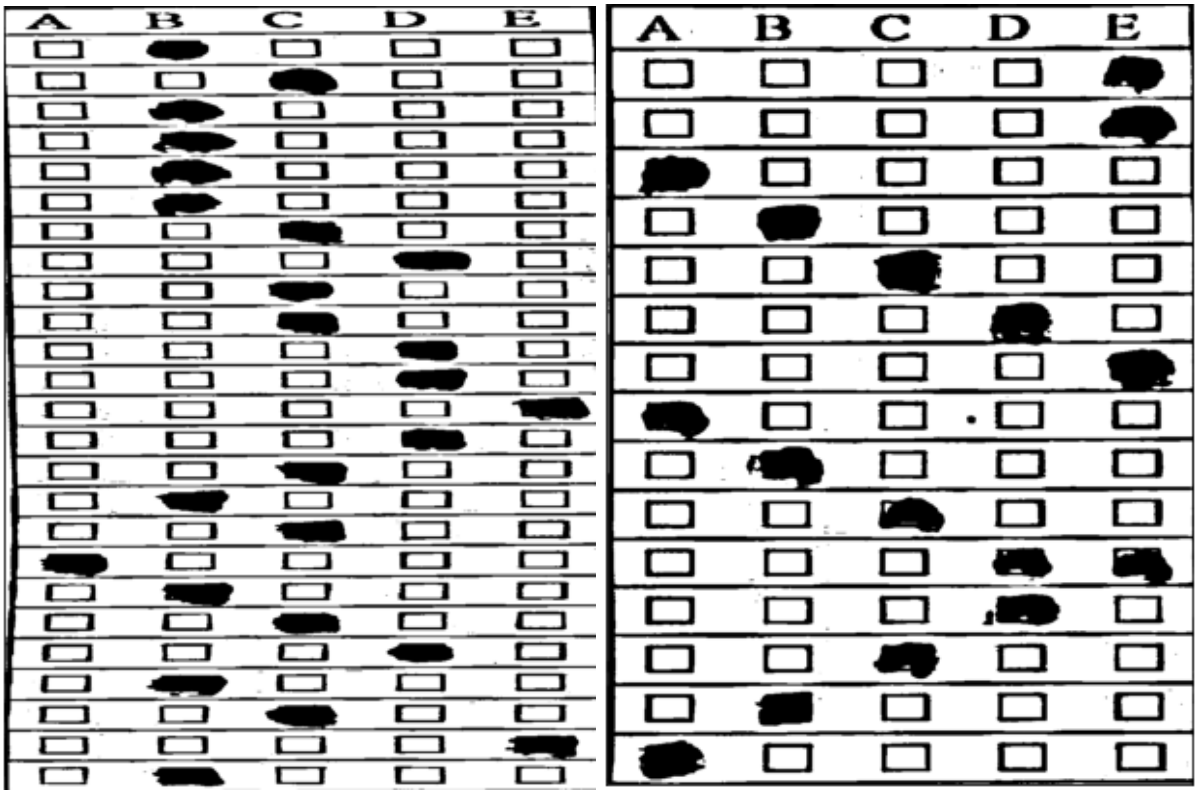
Our methodology is somewhat simple in this regard. We can clearly spot the answer panes in as the biggest rectangle in the picture. We just need to extract this and then count the choice that the student has made. First We extract the images by applying edge detection followed by contours detection. We get 2 separate images by warping the perspective to get a **birds eye view** of both the panes for the selection. The panes after warping the perspective are shown below:





It can clearly be seen that one pane has more congested rows than the other. This i due to one pane being smaller than the other. This problem is latr on catered by splitting the first pane in 26 rows(keeping in mind the header row for A B C D E) and splitting the second pane by 16 rows.

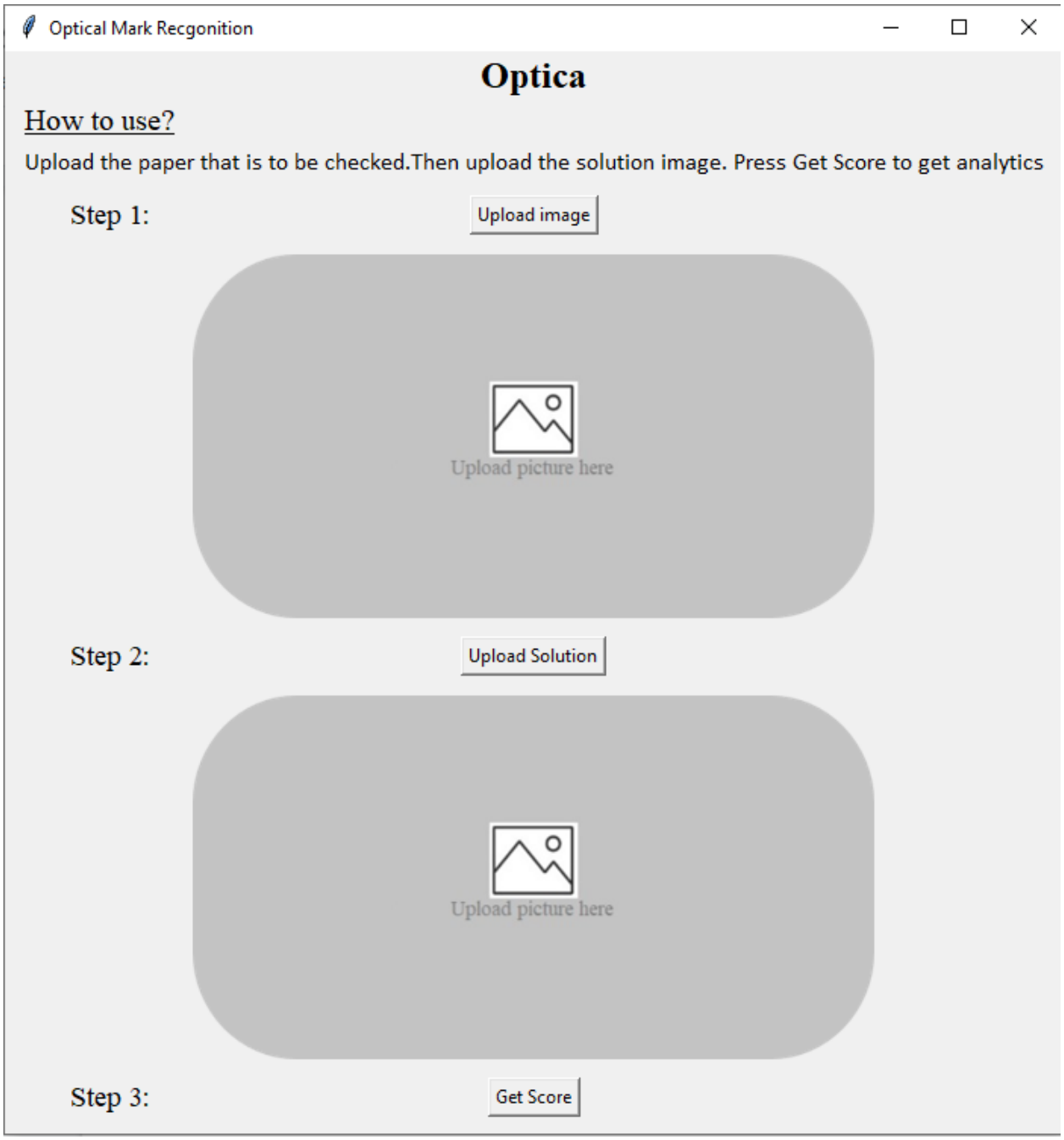
After obtaining the images that are shown above, Local adaptive gaussian thresholding is applied to get a binary image so that reading the choice of the student is made easier. The thresholding is applied to further clarify the image and remove any noise in it as well. The said images are shown below:



Lastly, we split each row and column and store that in an array iteratively. Hence after every 25th index of the array, column B starts and after every 52nd index column C as so on… We took advantage of this and counted the number of pixels in each box and traversed the array from 5 index simultaneously. The value that shows the number of black pixels greater than the set threshold is counted as being selected as shown in the example below.

Black: 2490 Black: 2360 Black: 2558 Black: 4244

Frontend:


Optical Mark Recgonition
Optica
How to use?
Upload the paper that is to be checked.Then upload the solution image. Press Get Score to get analytics
Step 1:
Upload image
Upload picture here
Step 2:
Upload Solution
Upload picture here
Step 3:
Get Score

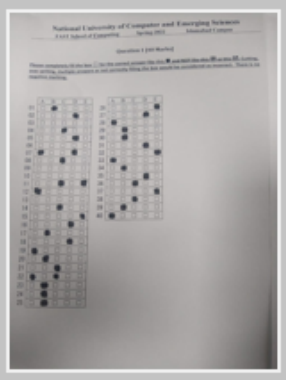
Optical Mark Recgonition

Optica

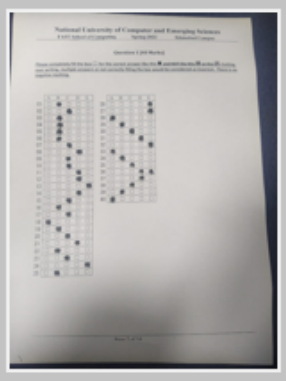
How to use?

Upload the paper that is to be checked.Then upload the solution image. Press Get Score to get analytics

Step 1: Upload image



Step 2: Upload Solution



Step 3: Get Score

Marks: 12

Multiple Seletions: 3

Empty MCQ's: 5

Thank You !!