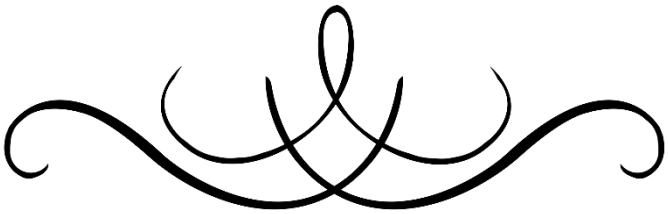


ANSWERS EXTRACTOR

DECEMBER 12, 2019

AIN SHAMS UNIVERSITY – FACULTY OF ENGINEERING

FACULTY OF ENGINEERING

AIN SHAMS UNIVERSITY

CSE 365: Computer Vision

Answers Extractor

Submitted by:

Abdelrahman Amr Issawi	aid-issawi@hotmail.com	16P6001@eng.asu.edu.eg
Abdelrahman Ibrahim ELGhamry	Ghamry98@hotmail.com	16P3043@eng.asu.edu.eg
Hossam ELDin Khaled	hossampen97@gmail.com	16P3025@eng.asu.edu.eg

Submitted to:

Dr. Mahmoud Khalil,

Eng. Amr Yassin.

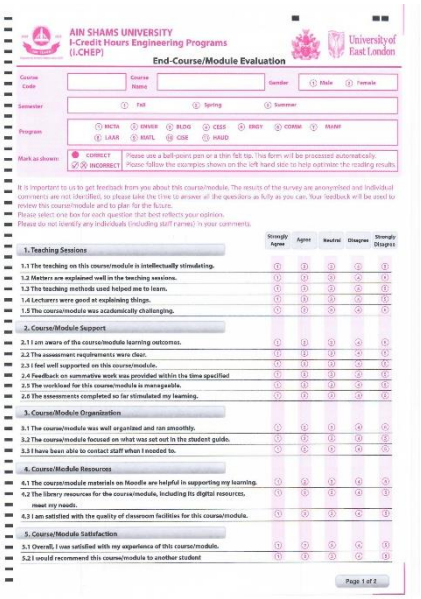
DECEMBER 12, 2019

A REPORT FOR COMPUTER VISION COURSE CODDED CSE365 WITH THE REQUIREMENTS OF AIN SHAMS UNIVERSITY

Table of Contents

1. INTRODUCTION

The goal of the project is to implement an algorithm which takes an input image, then apply some computer vision tools in order to recognize the answers, and finally the algorithm exports a text file as an output.


AIN SHAMS UNIVERSITY
I-Credit Hours Engineering Programs
(I.CHEP)
University of East London
End-Course/Module Evaluation
Course Code
Course Name
Gender
Male
Female
Semester
Fall
Spring
Summer
Program
MCTA
ENVEE
BLDG
CESS
ENGY
COMM
MANF
LAAR
MATL
CISE
MAUD
Mark as shown:
CORRECT
INCORRECT
Please use a ball-point pen or a thin felt tip. This form will be processed automatically.
Please follow the examples shown on the left hand side to help optimize the reading results.
It is important to us to get feedback from you about this course/module. The results of the survey are anonymised and individual comments are not identified, so please take the time to answer all the questions as fully as you can. Your feedback will be used to review this course/module and to plan for the future.
Please select one box for each question that best reflects your opinion.
Please do not identify any individuals (including staff names) in your comments.
Strongly Agree
Agree
Neutral
Disagree
Strongly Disagree
1. Teaching Sessions
1.1 The teaching on this course/module is intellectually stimulating.
1.2 Matters are explained well in the teaching sessions.
1.3 The teaching methods used helped me to learn.
1.4 Lecturers were good at explaining things.
1.5 The course/module was academically challenging.
2. Course/Module Support
2.1 I am aware of the course/module learning outcomes.
2.2 The assessment requirements were clear.
2.3 I feel well supported on this course/module.
2.4 Feedback on summative work was provided within the time specified
2.5 The workload for this course/module is manageable.
2.6 The assessments completed so far stimulated my learning.
3. Course/Module Organization
3.1 The course/module was well organized and ran smoothly.
3.2 The course/module focused on what was set out in the student guide.
3.3 I have been able to contact staff when I needed to.
4. Course/Module Resources
4.1 The course/module materials on Moodle are helpful in supporting my learning.
4.2 The library resources for the course/module, including its digital resources, meet my needs.
4.3 I am satisfied with the quality of classroom facilities for this course/module.
5. Course/Module Satisfaction
5.1 Overall, I was satisfied with my experience of this course/module.
5.2 I would recommend this course/module to another student
Page 1 of 2

The given form contains 22 questions as shown, the first 3 questions are asking about the student's gender, semester, and program. The rest 19 questions are divided into 5 categories MCQ, each one has an answer from the following:

- Strongly Agree,
- Agree,
- Neutral,
- Disagree,
- Strongly Disagree.

The report shows our implementation, steps we followed to get the right answers, the features we added, a detailed user guide, and at the end we are running multiple test cases.

2. BRIEF DESCRIPTION

1.1 Implementation

There are 5 python files, code.py, rotate.py, answers.py, scale.py, and image.py.
The **code.py** is the main file calling the other functions, it gets the test sample name and reads the image using the **imread()** function. Then, the image is passed to **scaleImg()** function defined in **scale.py** to scale the image to a defined scale of 1654 x 2338 and then the image is passed to **invRotation()** function in **rotate.py** where the function:

- Scale the image,
- Convert it to grey scale image,
- Detect edges by **Canny** function,
- Get the straight lines using **HoughLinesP,**
- Get the rotation angle from the first detected line,
- Get image center to rotate around it,
- Get the rotation matrix using **getRotationMatrix2D(),**
- Wrap the image with the rotation matrix using **wrapAffine()**,
- Return the result inverse rotated image.

The rotated image is saved to be displayed later, then **image.py** is used to get the locations of the circles,

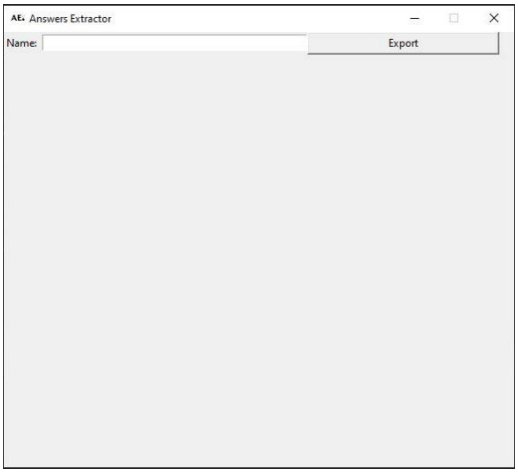
- Firstly, by applying **threshold()** on the image to show the circles only,
- Invert the image to make the circles in white,
- Apply opening on the image to eliminate the noise,
- Apply connected components **connectedComponentsWithStats()**,
- Return the centroids list.

The returned centroids list is casted “converted” into an array using **np.array()**.
Finally, the array of centroids is passed to the **answers.py** where we check the coordinates of each returned centroid, there are defined 8 lists for the sections: gender, semester, program, question 1, question 2, question 3, question 4, and question 5.
Each question is initialized to “Not solved”, the function **getAnswers()** loops to check the coordinates of the centroids and change the pre-defined variables into the right values and finally return the 8 lists containing the 22 answers.

The 8 returned lists are now ready to be written in a text file, a file is created, and saved after it is filled with the required data. And the text file is automatically initialized whenever the user clicks on the export button.

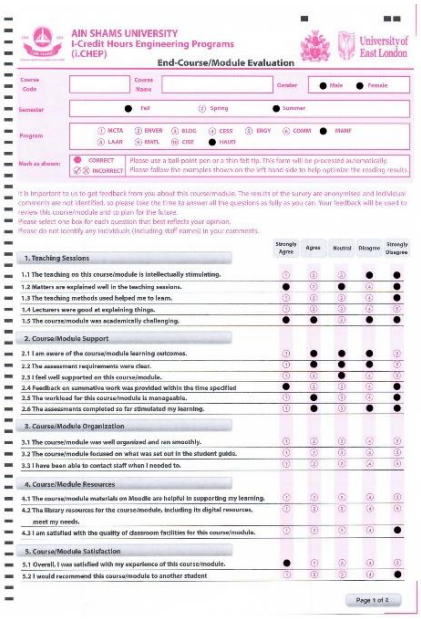
1.2 Added features

- GUI is implemented to take the test sample name in a specified form and displays the rotated enhanced image and the answers as well.
  The GUI is implemented using tkinter library in python to use the labels, buttons, entry fields, and adding an icon to the window. Also, PIL library is used to display images in the GUI, firstly, we save the inversely rotated image then we read it enhanced to be displayed in the GUI.



- Scaled images are handled and the answers are recognized as expected.

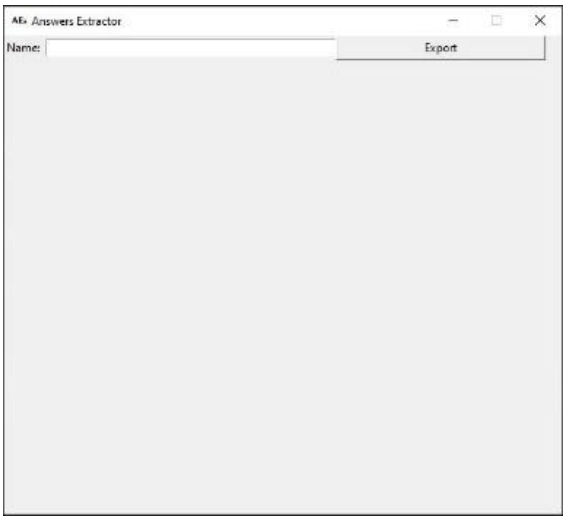
- The cases of an empty question or a question with more than one answer are handled, the first case is handled by giving an initial value, the second case is handled by a counter associated with the answers list, it's added and initialized to zero, and whenever the counter value exceeds "1" it means that the question has more than one answer, so this question gets the answer "Invalid Answer".
  If this case was not handled, then the question's answer would be the right most answer. EX: if the user selected Agree and Neutral the answer will be Neutral, but this is not acceptable.



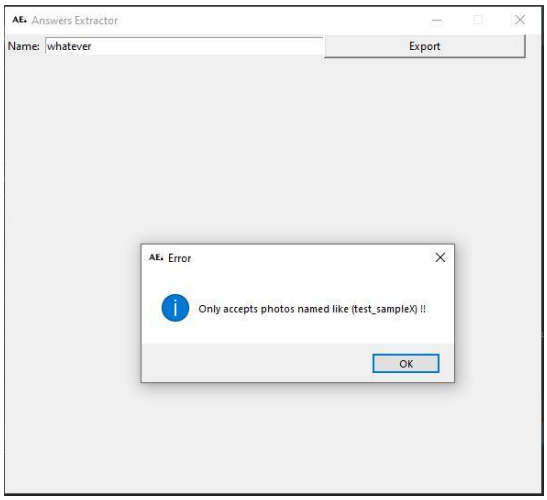
As shown any question has a counter bigger than one got the value "Invalid Answer", and any unanswered question got the value "Not solved".

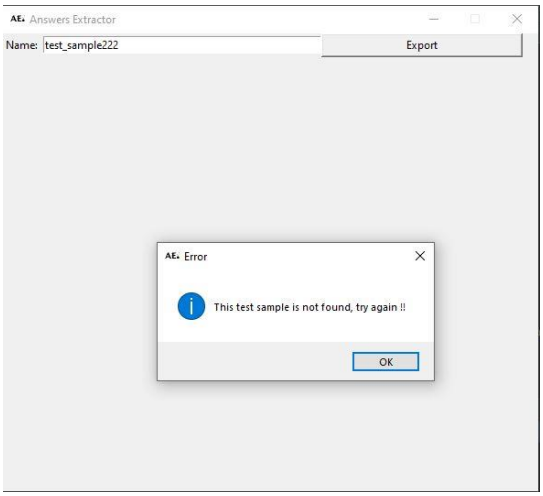
```
0.0
['Invalid Answer', 2]
['Invalid Answer', 2]
['Invalid Answer', 2]
[['Invalid Answer', 2], ['Invalid Answer', 3], ['Strongly Disagree', 1], ['Not Solved', 0], ['Invalid Answer', 4]]
[['Invalid Answer', 3], ['Invalid Answer', 3], ['Neutral', 1], ['Invalid Answer', 2], ['Invalid Answer', 2], ['Invalid Answer', 3]]
[['Not Solved', 0], ['Not Solved', 0], ['Not Solved', 0]]
[['Not Solved', 0], ['Not Solved', 0], ['Strongly Disagree', 1]]
[['Strongly Agree', 1], ['Strongly Disagree', 1]]
```

3. USER GUIDE



GUI opens once you run the executable file or run the program from the PyCharm. You are required to enter the test sample name in the right form. Ex: “test_sample(#no)” and an existed image with this name must be found in the folder named **CSE365_test_cases_project_1 folder.** If the 2 conditions are satisified the program will run successfully, in opposite, if one of the conditions is violated one of the following errors messages will show up.

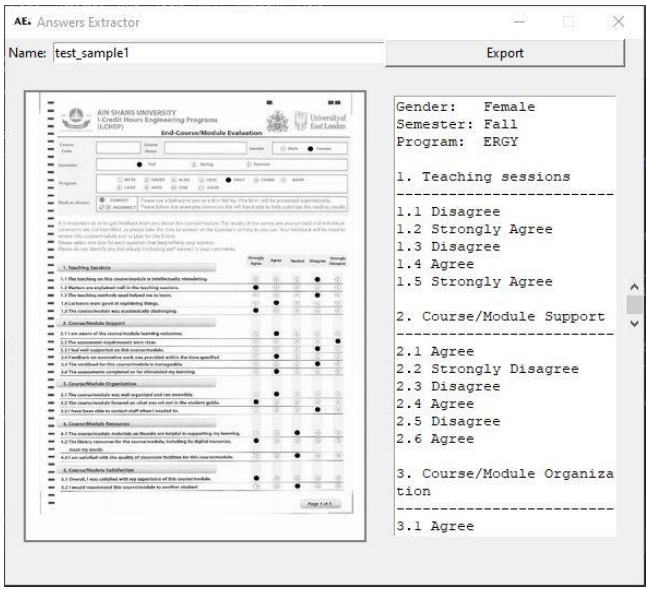


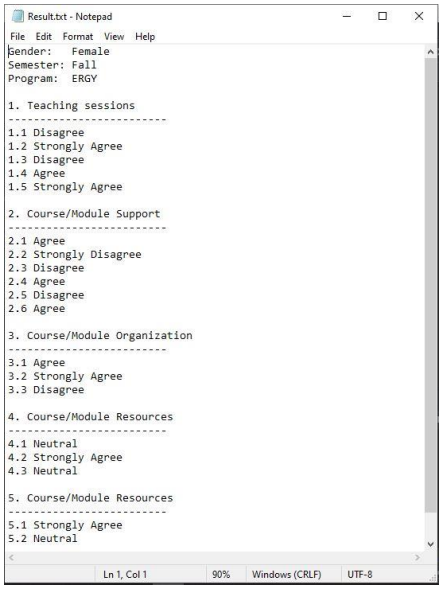


4. TEST CASES

4.1 Test Case 1

Suppose you entered “test_sample1” which exists in the specified folder.

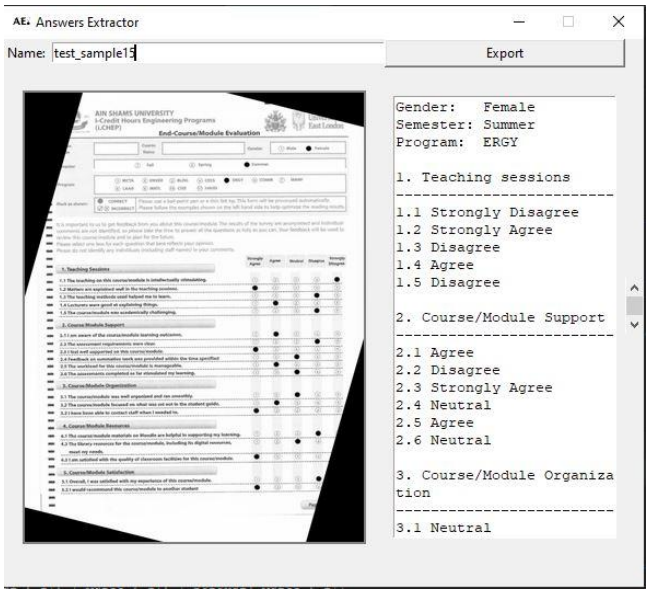


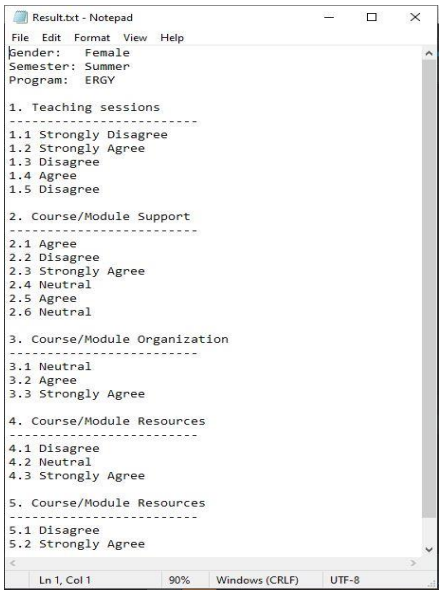


The program recognized the answers as expected, exported the text file. In addition, it displayed the results in a text area where you can scroll to see full results.

4.2 Test Case 2

Suppose you entered “test_sample15” which exists in the specified folder.

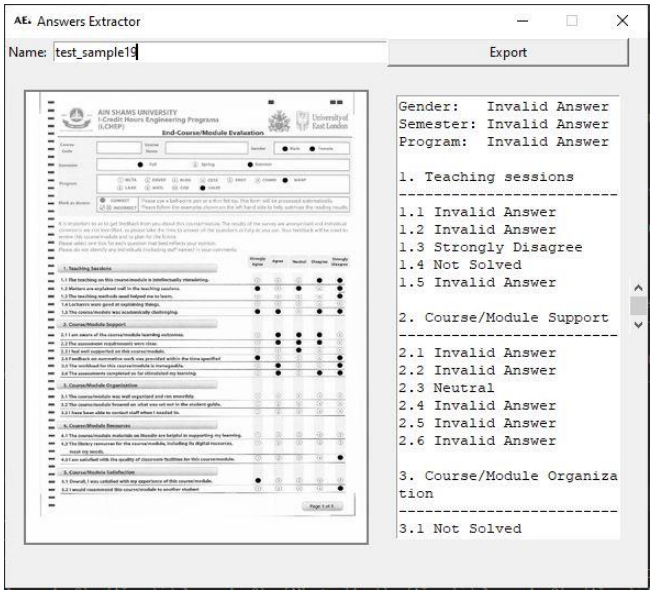


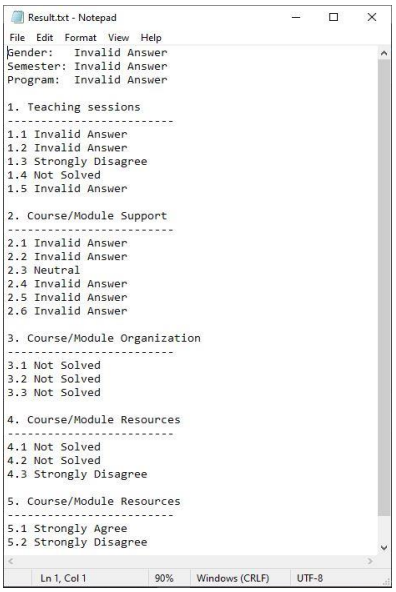


The program recognized the **rotation angle**, inversed it, and recognized the answers as expected, exported the text file. in addition, it displayed the results in a text area where you can scroll to see full results.

4.3 Test Case 3

Suppose you entered “test_sample19” which exists in the specified folder.

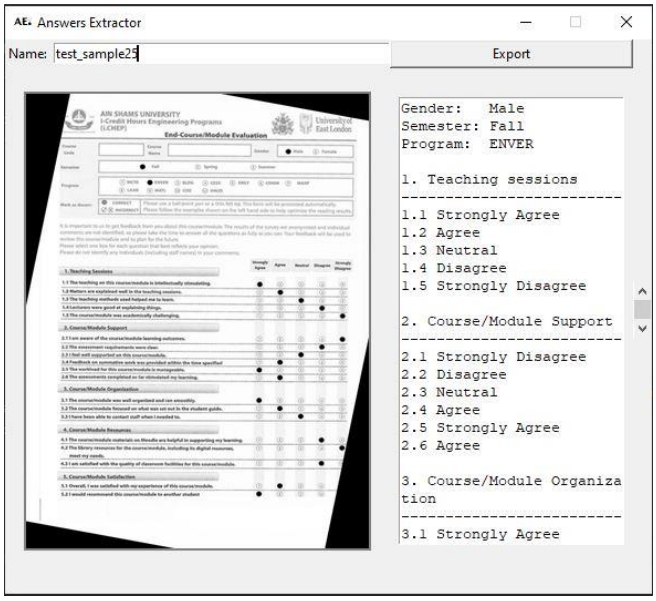


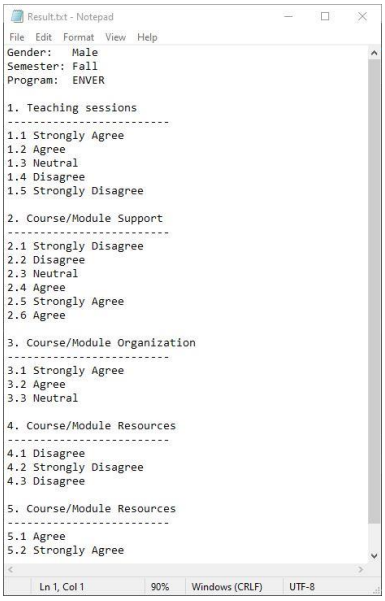


The program recognized the valid answers as expected, detected the invalid and unsolved answers as well, exported the text file. In addition, it displayed the results in a text area where you can scroll to see full results.

4.4 Test Case 4

Suppose you entered “test_sample25” which exists in the specified folder.





The original image is scaled to 500x707 the program rescaled the image, inversed the rotation, detected the valid answers as expected, and exported the text file. In addition, it displayed the results in a text area where you can scroll to see full results.

5. CODES

Code.py

```
import cv2
import numpy as np
from tkinter import *
from tkinter import messagebox
import os
import answers
from PIL import ImageTk, Image
import image
import rotate
import scale


def buttonClick():
    # read original image.
    name = e1.get()
    # handle wrong name
    try:
        if str(name).startswith("test_sample") and e1.get() != "":
            imgOriginal = cv2.imread("CSE365_test_cases_project_1/" + name +
".jpg", 1)
            imgOriginal = scale.scaleImg(imgOriginal)
            img = rotate.invRotation(imgOriginal)
            cv2.imwrite("CSE365_test_cases_project_1/rotatedImage.png", img)

            myImg = Image.open("CSE365_test_cases_project_1/rotatedImage.png")
            myImg = myImg.resize((310, 410), Image.ANTIALIAS)
            photo = ImageTk.PhotoImage(myImg)
            l2 = Label(bg="grey", image=photo)
            l2.image = photo
            l2.grid(row=2, column=0, columnspan=2)

            # getting the circles location
            centroids, cirlesNumber, labeledImg = image.getLabels(img)
            # cv2.imshow("check", labeledImg)

            # set answers
            centroids = np.array(centroids)
            gender, semester, program, firstQ, secondQ, thirdQ, fourthQ, fifthQ =
answers.getAnswers(centroids)
            # set the output text file data
            fileName = "Result.txt"
            myFile = open(fileName, 'w')
            myFile.write("Gender:    " + str(gender[0]) + "\n"
                         "Semester:  " + str(semester[0]) + "\n"
                         "Program:   " + str(program[0]) + "\n\n"
                         "1. Teaching sessions" + "\n"
                         "-------------------------" + "\n"
                         "1.1 " + str(firstQ[0][0]) + "\n"
                         "1.2 " + str(firstQ[1][0]) + "\n"
                         "1.3 " + str(firstQ[2][0]) + "\n"
                         "1.4 " + str(firstQ[3][0]) + "\n"
                         "1.5 " + str(firstQ[4][0]) + "\n\n"
                         "2. Course/Module Support" + "\n"
                         "-------------------------" + "\n"
                         "2.1 " + str(secondQ[0][0]) + "\n"
                         "2.2 " + str(secondQ[1][0]) + "\n"
                         "2.3 " + str(secondQ[2][0]) + "\n"
                         "2.4 " + str(secondQ[3][0]) + "\n"
                         "2.5 " + str(secondQ[4][0]) + "\n"
                         "2.6 " + str(secondQ[5][0]) + "\n\n"
                         "3. Course/Module Organization" + "\n"
                         "-------------------------" + "\n"
                         "3.1 " + str(thirdQ[0][0]) + "\n"
                         "3.2 " + str(thirdQ[1][0]) + "\n"
```

```
                        "3.3 " + str(thirdQ[2][0]) + "\n\n"
                        "4. Course/Module Resources" + "\n"
                        "-------------------------" + "\n"
                        "4.1 " + str(fourthQ[0][0]) + "\n"
                        "4.2 " + str(fourthQ[1][0]) + "\n"
                        "4.3 " + str(fourthQ[2][0]) + "\n\n"
                        "5. Course/Module Resources" + "\n"
                        "-------------------------" + "\n"
                        "5.1 " + str(fifthQ[0][0]) + "\n"
                        "5.2 " + str(fifthQ[1][0]) + "\n\n"
                        )
            myFile.close()
            myFile = open(fileName, 'r')
            content = myFile.read()
            text1 = Text(window, height=25, width=25)
            text1.grid(row=2, column=2, rowspan=30)
            text1.insert(INSERT, content)
            sb1 = Scrollbar(window)
            sb1.grid(row=1, column=3, rowspan=10)
            text1.configure(yscrollcommand=sb1.set)
            sb1.configure(command=text1.yview)
            window.update()
            myFile.close()
            os.startfile(fileName)

            print(gender)
            print(semester)
            print(program)
            print(firstQ)
            print(secondQ)
            print(thirdQ)
            print(fourthQ)
            print(fifthQ)

            cv2.waitKey()
        else:
            messagebox.showinfo("Error", "Only accepts photos named like
(test_sampleX) !!")

    except:
        messagebox.showinfo("Error", "This test sample is not found, try again !!")

# set the GUI
window = Tk()
window.geometry("585x500")
window.title("Answers Extractor")
window.iconbitmap("icon (1).ico")
window.resizable(False, False)
sys.path.append('/usr/local/lib/python2.7/site-packages')

l1 = Label(window, text="Name: ")
l1.grid(row=0, column =0)

title_text=StringVar()
e1 = Entry(window, textvariable=title_text, width=50)
e1.grid(row=0, column=1)

b1 = Button(window, text="Export", width=30, command=buttonClick)
b1.grid(row=0, column=2)

l3 = Label(window, text="")
l4 = Label(window, text="")
l3.grid(row=1, column=0)

window.mainloop()
cv2.waitKey()
```

Scale.py

```
import cv2

def scaleImg(img):
    imgOriginal = img
    dimensions = imgOriginal.shape

    height = imgOriginal.shape[0]
    width = imgOriginal.shape[1]
    print('Image Height       : ', height)
    print('Image Width        : ', width)

    if width == 1654 and height == 2338:
        return imgOriginal
    else:
        dim = (1654, 2338)
        imgOriginal = cv2.resize(imgOriginal, dim, interpolation = cv2.INTER_AREA)
        dimensions = imgOriginal.shape
        height = imgOriginal.shape[0]
        width = imgOriginal.shape[1]
        print('Image Height       : ', height)
        print('Image Width        : ', width)
        return imgOriginal

cv2.waitKey()
```

Rotate.py

```
import cv2
import numpy as np
import math


def invRotation(image):
    # scale image
    image = cv2.resize(image, (0, 0), fx=0.3, fy=0.3)

    # convert into grey scale img
    image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    # detect edges, used default Sobel-kernel
    img_edges = cv2.Canny(image, 100, 100, apertureSize=3)
    # Run Hough on edge detected image
    lines = cv2.HoughLinesP(img_edges, 1, math.pi / 180.0, 100, minLineLength=100,
maxLineGap=5)

    angle = 0

    # get the rotation angle of the line
    for x1, y1, x2, y2 in lines[0]:
        cv2.line(image, (x1, y1), (x2, y2), (255, 0, 0), 3)
        angle = math.degrees(math.atan2(y2 - y1, x2 - x1))
    print(angle)

    image_center = tuple(np.array(image.shape[1::-1]) / 2)
    rot_mat = cv2.getRotationMatrix2D(image_center, angle, 1.0)
    result = cv2.warpAffine(image, rot_mat, image.shape[1::-1],
flags=cv2.INTER_LINEAR)
    return result
```

Image.py

```
import cv2
import numpy as np

def getLabels(img):
    # threshold, and inverse the photo
    th, ciclesImg = cv2.threshold(img, 30, 255, cv2.THRESH_BINARY)
    ciclesImg = 255 - ciclesImg

    # obtaining answers' circles alone without any noise
    SE = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (5, 5))
    ciclesImg = cv2.morphologyEx(ciclesImg, cv2.MORPH_OPEN, SE)
    # cv2.imshow("detected1", ciclesImg)

    # connected components to obtain centroid coordinates
    result = cv2.connectedComponentsWithStats(ciclesImg, 4, cv2.CV_32S)
    number = result[0]
    centroids = result[3]

    # counter = 1
    # for i in centroids[1:]:  # in the normal case the first centroid represents
the background.
    #     temp = tuple([int(i[0]), int(i[1])])  # get the x, y coordinates
    #     cv2.putText(ciclesImg, str(counter), temp, cv2.FONT_HERSHEY_SIMPLEX, 0.5,
(100,100,100), 2)
    #     # print(str(counter) + str(temp))
    #     counter += 1
    return centroids, number - 1, ciclesImg
```

Answers.py

```
def getAnswers(centroids):
    # setting defaults
    gender = ["Not Solved", 0]
    semester = ["Not Solved", 0]
    program = ["Not Solved", 0]
    firstQ = [["Not Solved", 0], ["Not Solved", 0], ["Not Solved", 0], ["Not
Solved", 0], ["Not Solved", 0]]
    secondQ = [["Not Solved", 0], ["Not Solved", 0], ["Not Solved", 0], ["Not
Solved", 0], ["Not Solved", 0],
               ["Not Solved", 0]]
    thirdQ = [["Not Solved", 0], ["Not Solved", 0], ["Not Solved", 0]]
    fourthQ = [["Not Solved", 0], ["Not Solved", 0], ["Not Solved", 0]]
    fifthQ = [["Not Solved", 0], ["Not Solved", 0]]

    # getting exact values, check for invalid answers.
    for i, j in centroids[1:]:
        # print( str(i) + "," + str(j))
        if 85 < j < 95:
            if 370 < i < 385:
                gender[0] = "Male"
                gender[1] += 1
            elif 410 < i < 425:
                gender[0] = "Female"
                gender[1] += 1
            if gender[1] > 1:
                gender[0] = "Invalid Answer"

        elif 108 < j < 115:
            if 160 < i < 175:
                semester[0] = "Fall"
                semester[1] += 1
            elif 243 < i < 250:
                semester[0] = "Spring"
                semester[1] += 1
            elif 323 < i < 330:
```

```
                semester[0] = "Summer"
                semester[1] += 1
            if semester[1] > 1:
                semester[0] = "Invalid Answer"

        elif 134 < j < 139:
            if 134 < i < 142:
                program[0] = "MCTA"
                program[1] += 1
            elif 174 < i < 180:
                program[0] = "ENVER"
                program[1] += 1
            elif 213 < i < 219:
                program[0] = "BLDG"
                program[1] += 1
            elif 255 < i < 260:
                program[0] = "CESS"
                program[1] += 1
            elif 293 < i < 300:
                program[0] = "ERGY"
                program[1] += 1
            elif 335 < i < 340:
                program[0] = "COMM"
                program[1] += 1
            elif 370 < i < 389:
                program[0] = "MANF"
                program[1] += 1

        elif 144 < j < 152:
            if 134 < i < 142:
                program[0] = "LAAR"
                program[1] += 1
            elif 173 < i < 180:
                program[0] = "MATL"
                program[1] += 1
            elif 213 < i < 219:
                program[0] = "CISE"
                program[1] += 1
            elif 253 < i < 260:
                program[0] = "HAUD"
                program[1] += 1
            if program[1] > 1:
                program[0] = "Invalid Answer"

        elif 290 < j < 295:
            if 330 < i < 342:
                firstQ[0][0] = "Strongly Agree"
                firstQ[0][1] += 1
            elif 360 < i < 372:
                firstQ[0][0] = "Agree"
                firstQ[0][1] += 1
            elif 390 < i < 402:
                firstQ[0][0] = "Neutral"
                firstQ[0][1] += 1
            elif 420 < i < 432:
                firstQ[0][0] = "Disagree"
                firstQ[0][1] += 1
            elif 450 < i < 462:
                firstQ[0][0] = "Strongly Disagree"
                firstQ[0][1] += 1
            if firstQ[0][1] > 1:
                firstQ[0][0] = "Invalid Answer"

        elif 300 < j < 310:
            if 330 < i < 340:
                firstQ[1][0] = "Strongly Agree"
                firstQ[1][1] += 1
            elif 360 < i < 372:
```

```
                firstQ[1][0] = "Agree"
                firstQ[1][1] += 1
            elif 390 < i < 402:
                firstQ[1][0] = "Neutral"
                firstQ[1][1] += 1
            elif 420 < i < 432:
                firstQ[1][0] = "Disagree"
                firstQ[1][1] += 1
            elif 450 < i < 462:
                firstQ[1][0] = "Strongly Disagree"
                firstQ[1][1] += 1
            if firstQ[1][1] > 1:
                firstQ[1][0] = "Invalid Answer"

        elif 313 < j < 321:
            if 330 < i < 342:
                firstQ[2][0] = "Strongly Agree"
                firstQ[2][1] += 1
            elif 360 < i < 372:
                firstQ[2][0] = "Agree"
                firstQ[2][1] += 1
            elif 390 < i < 402:
                firstQ[2][0] = "Neutral"
                firstQ[2][1] += 1
            elif 420 < i < 432:
                firstQ[2][0] = "Disagree"
                firstQ[2][1] += 1
            elif 450 < i < 462:
                firstQ[2][0] = "Strongly Disagree"
                firstQ[2][1] += 1
            if firstQ[2][1] > 1:
                firstQ[2][0] = "Invalid Answer"

        elif 326 < j < 335:
            if 330 < i < 340:
                firstQ[3][0] = "Strongly Agree"
                firstQ[3][1] += 1
            elif 360 < i < 372:
                firstQ[3][0] = "Agree"
                firstQ[3][1] += 1
            elif 390 < i < 402:
                firstQ[3][0] = "Neutral"
                firstQ[3][1] += 1
            elif 420 < i < 432:
                firstQ[3][0] = "Disagree"
                firstQ[3][1] += 1
            elif 450 < i < 462:
                firstQ[3][0] = "Strongly Disagree"
                firstQ[3][1] += 1
            if firstQ[3][1] > 1:
                firstQ[3][0] = "Invalid Answer"

        elif 338 < j < 348:
            if 330 < i < 342:
                firstQ[4][0] = "Strongly Agree"
                firstQ[4][1] += 1
            elif 360 < i < 372:
                firstQ[4][0] = "Agree"
                firstQ[4][1] += 1
            elif 390 < i < 402:
                firstQ[4][0] = "Neutral"
                firstQ[4][1] += 1
            elif 420 < i < 432:
                firstQ[4][0] = "Disagree"
                firstQ[4][1] += 1
            elif 450 < i < 462:
                firstQ[4][0] = "Strongly Disagree"
                firstQ[4][1] += 1
```

```
            if firstQ[4][1] > 1:
                firstQ[4][0] = "Invalid Answer"

        elif 373 < j < 383:
            if 330 < i < 342:
                secondQ[0][0] = "Strongly Agree"
                secondQ[0][1] += 1
            elif 360 < i < 372:
                secondQ[0][0] = "Agree"
                secondQ[0][1] += 1
            elif 390 < i < 402:
                secondQ[0][0] = "Neutral"
                secondQ[0][1] += 1
            elif 420 < i < 432:
                secondQ[0][0] = "Disagree"
                secondQ[0][1] += 1
            elif 450 < i < 462:
                secondQ[0][0] = "Strongly Disagree"
                secondQ[0][1] += 1
            if secondQ[0][1] > 1:
                secondQ[0][0] = "Invalid Answer"

        elif 386 < j < 392:
            if 330 < i < 342:
                secondQ[1][0] = "Strongly Agree"
                secondQ[1][1] += 1
            elif 360 < i < 372:
                secondQ[1][0] = "Agree"
                secondQ[1][1] += 1
            elif 390 < i < 402:
                secondQ[1][0] = "Neutral"
                secondQ[1][1] += 1
            elif 420 < i < 432:
                secondQ[1][0] = "Disagree"
                secondQ[1][1] += 1
            elif 450 < i < 462:
                secondQ[1][0] = "Strongly Disagree"
                secondQ[1][1] += 1
            if secondQ[1][1] > 1:
                secondQ[1][0] = "Invalid Answer"

        elif 397 < j < 403:
            if 330 < i < 342:
                secondQ[2][0] = "Strongly Agree"
                secondQ[2][1] += 1
            elif 360 < i < 372:
                secondQ[2][0] = "Agree"
                secondQ[2][1] += 1
            elif 390 < i < 402:
                secondQ[2][0] = "Neutral"
                secondQ[2][1] += 1
            elif 420 < i < 432:
                secondQ[2][0] = "Disagree"
                secondQ[2][1] += 1
            elif 450 < i < 462:
                secondQ[2][0] = "Strongly Disagree"
                secondQ[2][1] += 1
            if secondQ[2][1] > 1:
                secondQ[2][0] = "Invalid Answer"

        elif 407 < j < 416:
            if 330 < i < 342:
                secondQ[3][0] = "Strongly Agree"
                secondQ[3][1] += 1
            elif 360 < i < 372:
                secondQ[3][0] = "Agree"
                secondQ[3][1] += 1
            elif 390 < i < 402:
```

```
                secondQ[3][0] = "Neutral"
                secondQ[3][1] += 1
            elif 420 < i < 432:
                secondQ[3][0] = "Disagree"
                secondQ[3][1] += 1
            elif 450 < i < 462:
                secondQ[3][0] = "Strongly Disagree"
                secondQ[3][1] += 1
            if secondQ[3][1] > 1:
                secondQ[3][0] = "Invalid Answer"

        elif 420 < j < 428:
            if 330 < i < 342:
                secondQ[4][0] = "Strongly Agree"
                secondQ[4][1] += 1
            elif 360 < i < 372:
                secondQ[4][0] = "Agree"
                secondQ[4][1] += 1
            elif 390 < i < 402:
                secondQ[4][0] = "Neutral"
                secondQ[4][1] += 1
            elif 420 < i < 432:
                secondQ[4][0] = "Disagree"
                secondQ[4][1] += 1
            elif 450 < i < 462:
                secondQ[4][0] = "Strongly Disagree"
                secondQ[4][1] += 1
            if secondQ[4][1] > 1:
                secondQ[4][0] = "Invalid Answer"

        elif 431 < j < 440:
            if 330 < i < 342:
                secondQ[5][0] = "Strongly Agree"
                secondQ[5][1] += 1
            elif 360 < i < 372:
                secondQ[5][0] = "Agree"
                secondQ[5][1] += 1
            elif 390 < i < 402:
                secondQ[5][0] = "Neutral"
                secondQ[5][1] += 1
            elif 420 < i < 432:
                secondQ[5][0] = "Disagree"
                secondQ[5][1] += 1
            elif 450 < i < 462:
                secondQ[5][0] = "Strongly Disagree"
                secondQ[5][1] += 1
            if secondQ[5][1] > 1:
                secondQ[5][0] = "Invalid Answer"

        elif 468 < j < 475:
            if 330 < i < 342:
                thirdQ[0][0] = "Strongly Agree"
                thirdQ[0][1] += 1
            elif 360 < i < 372:
                thirdQ[0][0] = "Agree"
                thirdQ[0][1] += 1
            elif 390 < i < 402:
                thirdQ[0][0] = "Neutral"
                thirdQ[0][1] += 1
            elif 420 < i < 432:
                thirdQ[0][0] = "Disagree"
                thirdQ[0][1] += 1
            elif 450 < i < 462:
                thirdQ[0][0] = "Strongly Disagree"
                thirdQ[0][1] += 1
            if thirdQ[0][1] > 1:
                thirdQ[0][0] = "Invalid Answer"
```

```
        elif 480 < j < 489:
            if 330 < i < 342:
                thirdQ[1][0] = "Strongly Agree"
                thirdQ[1][1] += 1
            elif 360 < i < 372:
                thirdQ[1][0] = "Agree"
                thirdQ[1][1] += 1
            elif 390 < i < 402:
                thirdQ[1][0] = "Neutral"
                thirdQ[1][1] += 1
            elif 420 < i < 432:
                thirdQ[1][0] = "Disagree"
                thirdQ[1][1] += 1
            elif 450 < i < 462:
                thirdQ[1][0] = "Strongly Disagree"
                thirdQ[1][1] += 1
            if thirdQ[1][1] > 1:
                thirdQ[1][0] = "Invalid Answer"

        elif 492 < j < 500:
            if 330 < i < 342:
                thirdQ[2][0] = "Strongly Agree"
                thirdQ[2][1] += 1
            elif 360 < i < 372:
                thirdQ[2][0] = "Agree"
                thirdQ[2][1] += 1
            elif 390 < i < 402:
                thirdQ[2][0] = "Neutral"
                thirdQ[2][1] += 1
            elif 420 < i < 432:
                thirdQ[2][0] = "Disagree"
                thirdQ[2][1] += 1
            elif 450 < i < 462:
                thirdQ[2][0] = "Strongly Disagree"
                thirdQ[2][1] += 1
            if thirdQ[2][1] > 1:
                thirdQ[2][0] = "Invalid Answer"

        elif 528 < j < 536:
            if 330 < i < 342:
                fourthQ[0][0] = "Strongly Agree"
                fourthQ[0][1] += 1
            elif 360 < i < 372:
                fourthQ[0][0] = "Agree"
                fourthQ[0][1] += 1
            elif 390 < i < 402:
                fourthQ[0][0] = "Neutral"
                fourthQ[0][1] += 1
            elif 420 < i < 432:
                fourthQ[0][0] = "Disagree"
                fourthQ[0][1] += 1
            elif 450 < i < 462:
                fourthQ[0][0] = "Strongly Disagree"
                fourthQ[0][1] += 1
            if fourthQ[0][1] > 1:
                fourthQ[0][0] = "Invalid Answer"

        elif 539 < j < 548:
            if 330 < i < 342:
                fourthQ[1][0] = "Strongly Agree"
                fourthQ[1][1] += 1
            elif 360 < i < 372:
                fourthQ[1][0] = "Agree"
                fourthQ[1][1] += 1
            elif 390 < i < 402:
                fourthQ[1][0] = "Neutral"
                fourthQ[1][1] += 1
            elif 420 < i < 432:
```

```
                fourthQ[1][0] = "Disagree"
                fourthQ[1][1] += 1
            elif 450 < i < 462:
                fourthQ[1][0] = "Strongly Disagree"
                fourthQ[1][1] += 1
            if fourthQ[1][1] > 1:
                fourthQ[1][0] = "Invalid Answer"

        elif 564 < j < 574:
            if 330 < i < 342:
                fourthQ[2][0] = "Strongly Agree"
                fourthQ[2][1] += 1
            elif 360 < i < 372:
                fourthQ[2][0] = "Agree"
                fourthQ[2][1] += 1
            elif 390 < i < 402:
                fourthQ[2][0] = "Neutral"
                fourthQ[2][1] += 1
            elif 420 < i < 432:
                fourthQ[2][0] = "Disagree"
                fourthQ[2][1] += 1
            elif 450 < i < 462:
                fourthQ[2][0] = "Strongly Disagree"
                fourthQ[2][1] += 1
            if fourthQ[2][1] > 1:
                fourthQ[2][0] = "Invalid Answer"

        elif 598 < j < 608:
            if 330 < i < 342:
                fifthQ[0][0] = "Strongly Agree"
                fifthQ[0][1] += 1
            elif 360 < i < 372:
                fifthQ[0][0] = "Agree"
                fifthQ[0][1] += 1
            elif 390 < i < 402:
                fifthQ[0][0] = "Neutral"
                fifthQ[0][1] += 1
            elif 420 < i < 432:
                fifthQ[0][0] = "Disagree"
                fifthQ[0][1] += 1
            elif 450 < i < 462:
                fifthQ[0][0] = "Strongly Disagree"
                fifthQ[0][1] += 1
            if fifthQ[0][1] > 1:
                fifthQ[0][0] = "Invalid Answer"

        elif 611 < j < 620:
            if 330 < i < 342:
                fifthQ[1][0] = "Strongly Agree"
                fifthQ[1][1] += 1
            elif 360 < i < 372:
                fifthQ[1][0] = "Agree"
                fifthQ[1][1] += 1
            elif 390 < i < 402:
                fifthQ[1][0] = "Neutral"
                fifthQ[1][1] += 1
            elif 420 < i < 432:
                fifthQ[1][0] = "Disagree"
                fifthQ[1][1] += 1
            elif 450 < i < 462:
                fifthQ[1][0] = "Strongly Disagree"
                fifthQ[1][1] += 1
            if fifthQ[1][1] > 1:
                fifthQ[1][0] = "Invalid Answer"

    return gender, semester, program, firstQ, secondQ, thirdQ, fourthQ, fifthQ
```