|  |  |
| --- | --- |
|  |  |
| *Optical Mark Recognition* |  |
|  |  |
|  | June 22, 2021Ahmed Ahmed ElrewaidyAhmed Nabil LotfyEslam Khalid ElfauomyAkram Hesham RagabBasma Mohamed Ahmed |

### TABLE OF CONTENT:

* Grabbing and sorting the contours
* Filtering only circles
* Threshold and making bubbles
* Comparing answers
* Renaming files with student name and grade

# Grabbing and sorting the contours:

In this section we are interested in the contours in our questions table but sorted this time. So, we grab them first then we sort them and store them in a variable.

# Filtering only circles:

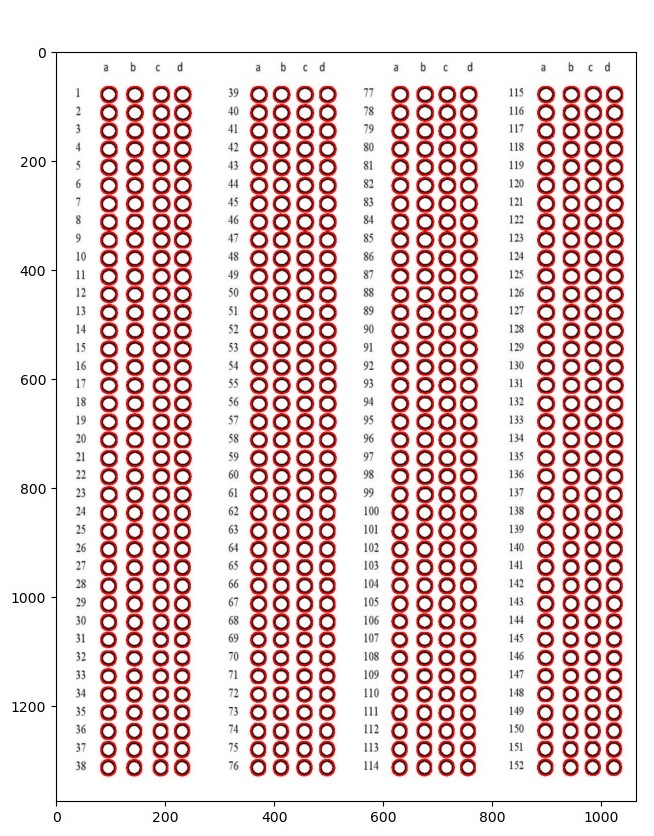
 Now that is a very critical step as its parameters are so sensitive and need a careful tuning. There is a trick implemented to overcome this problem in the 150 questions model, and we will discuss its details later.

Fig 1 Circles contours

Magic done! We finally reached a goal, we successfully and perfectly captured our 608 circles for our 152 questions. Note that when we say circles, we don’t really mean circles, it’s more of an oval, which is a good time to answer the question “why don’t we just use circle detection? “. The algorithms available for circle detection need almost perfect circles which makes it nearly impossible usable for our task.

# threshold and making bubbles:

Now got our contours or ovals detected and sorted, we need to know which one is answered and which is not. We can do this by applying threshold first and then counting the non-zero pixels inside of each oval. The more the oval gets a value, the more likely this oval is the answer for this question. This step also holds an important hyperparameter to tune. The figures below show an example.

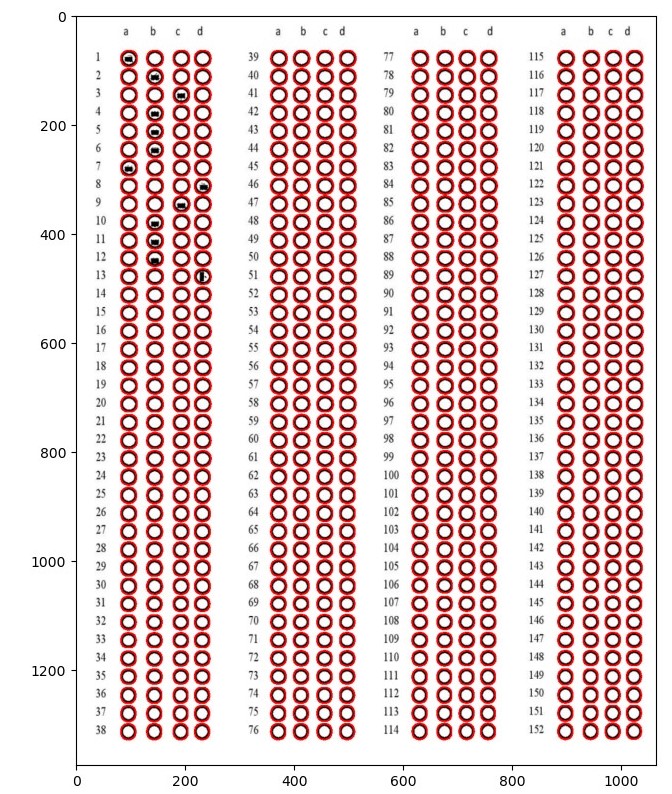


Fig 2 bubbled questions

Well it’s a little distorted filling to the circles but it will deliver the idea. The bubbled oval now has a bigger value than the non-bubbled ones.

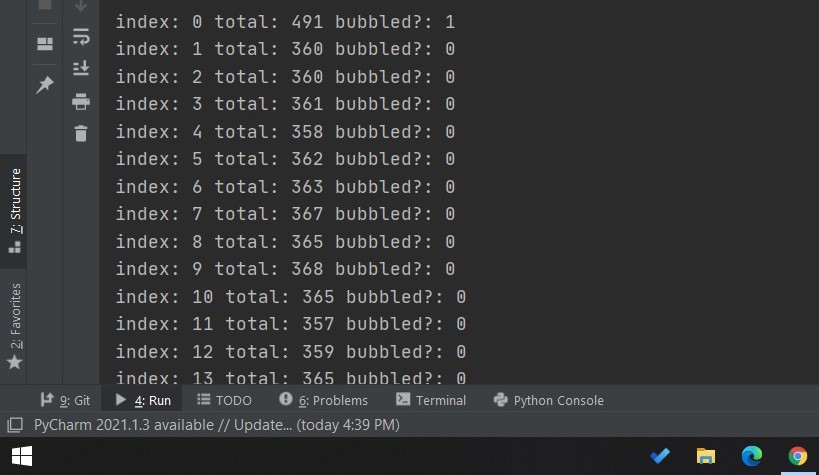


Fig 3 bubbled or not results.

In the figure above in the very first index we have the value 1, which indicates that there is a bubble in this place, which matches the figure 2.13.

There is an even better way to view the answers as if you are looking directly to the paper. Doing some logic with code, we can get this result in the figure below.

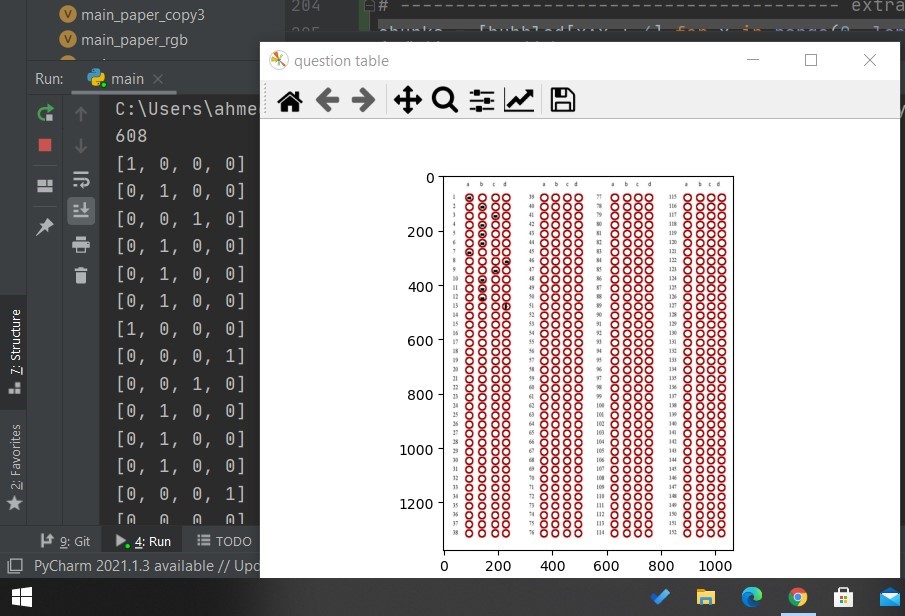


Fig 4 a nicer presentation of the bubbles.

Note that in the figure above we can now look at the results as well as the image and see that both are identical (one for each bubbled oval and zero for non-bubbled oval).

# Comparing Answers:

Now that we can identify bubbles, we have the ability to set answer model which is fig 2.1 for example, and start comparing all our sheets with it, if the question bubbles are identical to the answer model, we raise the grade by one.

# Renaming files with student name and grade:

Now that we corrected all our sheets, we can rename the sheet with the name of the student and the student grade as the new file name. In the figure below there is an example.

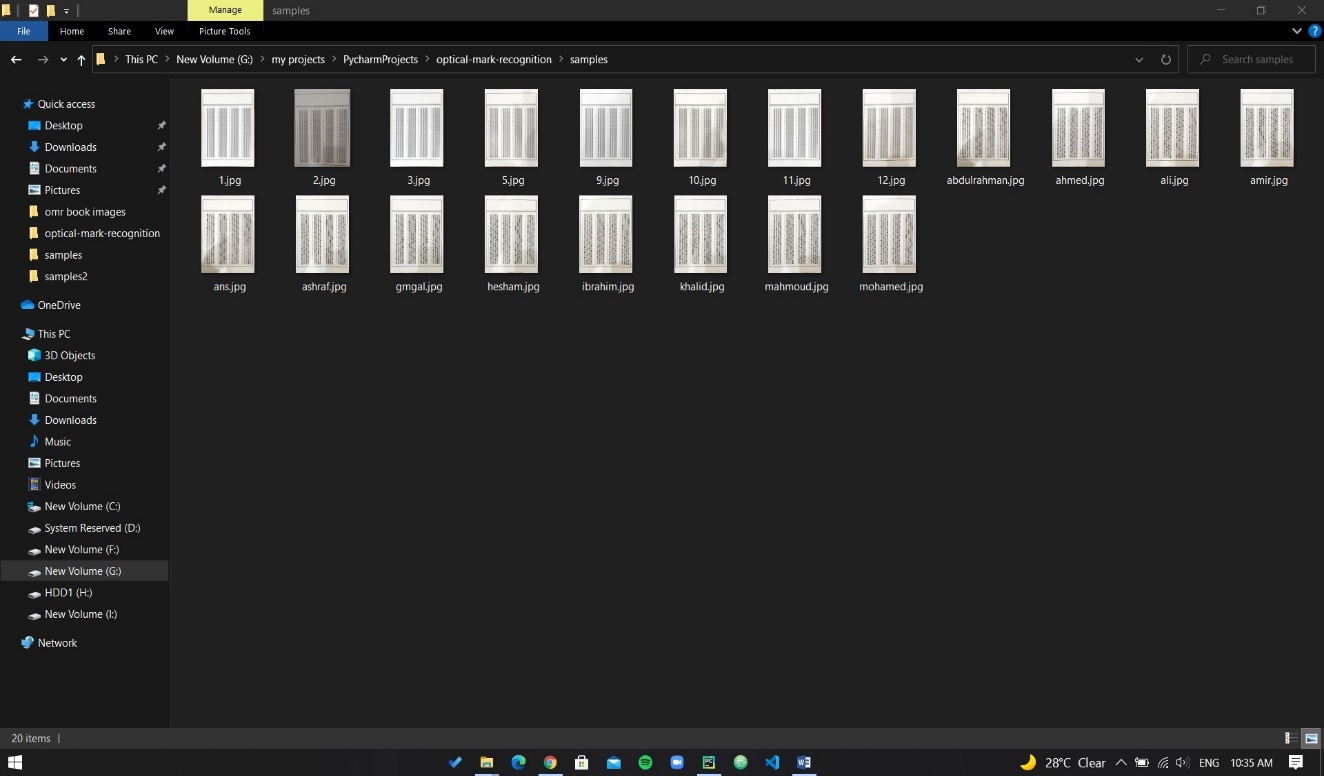


Fig 5 samples on project directory

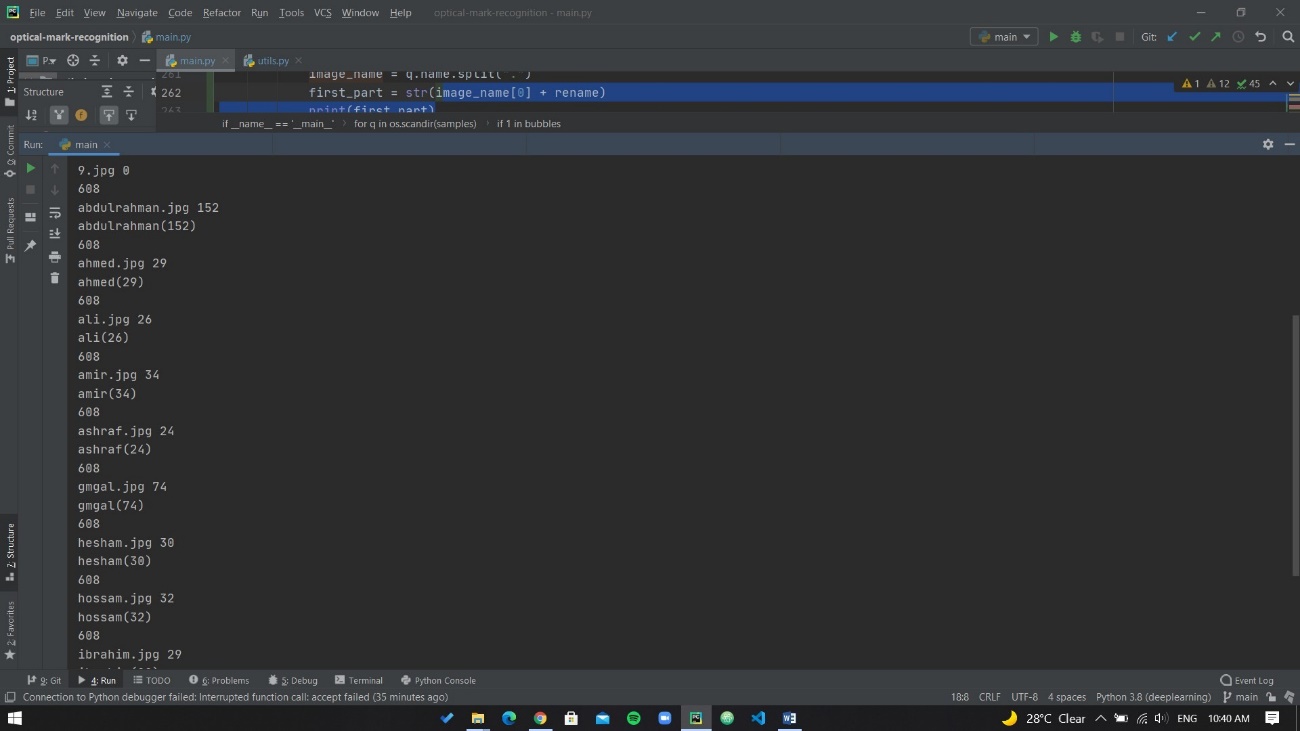


Fig 6 System Output

While system is generating its output, it’s also renaming each sheet and saving it in the project directory as in the figure below.

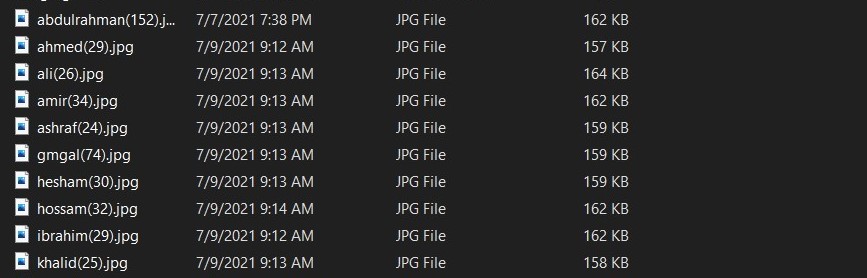


Fig 7 Files Renamed