Image Processing & Computer Vision

Project Proposal

Prepared By

Name	**Section**	**Bench No.**
Radwa Ahmed	1	30
Iten Emad	1	19
Mohamed Saad	2	14
Essam Wisam	2	1

Under the supervision of
Dr. Mayada Hadhoud
Eng. Youssef Ghattas

November 2021

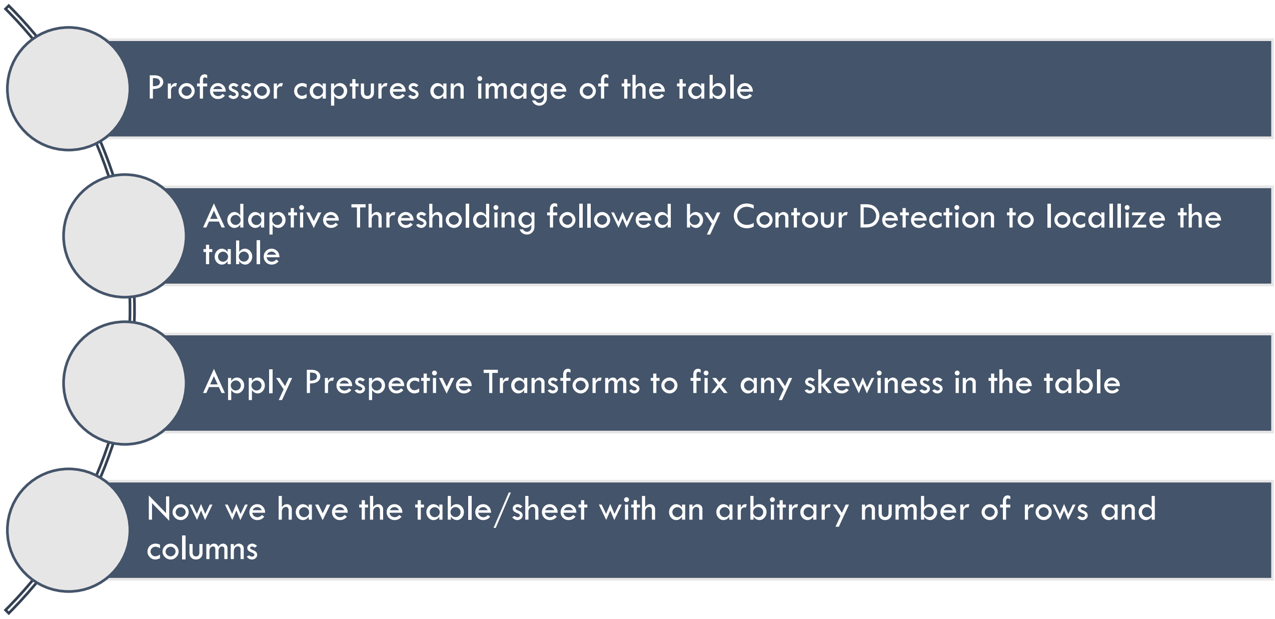
Project Idea & Need

Countless hours are spent by our beloved TAs and professors in grading papers when their valuable time should go to research instead. The point is that many aspects of the grading process are boring tasks that can be just automated. Thus, we decided that for a project we will propose an automatic grader. It's enough for the professor to just do the on-paper grading and leave us the rest and if their exam is MCQ then there is no need for them to see any of the papers in the first place.

In more detail, the idea is simply a grader that takes a captured photo of the handwritten graded results for each question in an exam for each student (in a table form with possible numeric grades or A, B, C, D, E, F) and then produce a computerized version of the table that can be passed directly to the students. Calling names and grades in front of everyone is no longer needed. This is only one module in our project. The other module is a bubble sheet corrector for MCQ exams. It's okay if no scanner is available because a camera shot from the phone should be more than enough.

Informative Block Diagram

Table/Bubble Sheet Detection (PageExtractor.py)



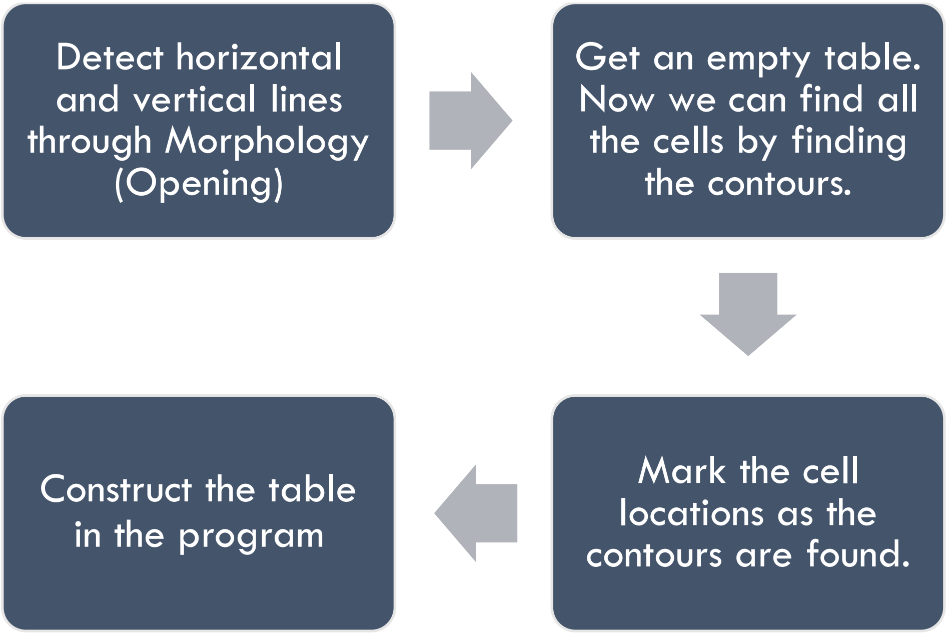
Professor captures an image of the table

Adaptive Thresholding followed by Contour Detection to locallize the table

Apply Prespective Transforms to fix any skewiness in the table

Now we have the table/sheet with an arbitrary number of rows and columns

Cell Detection (Cell Extractor.py)



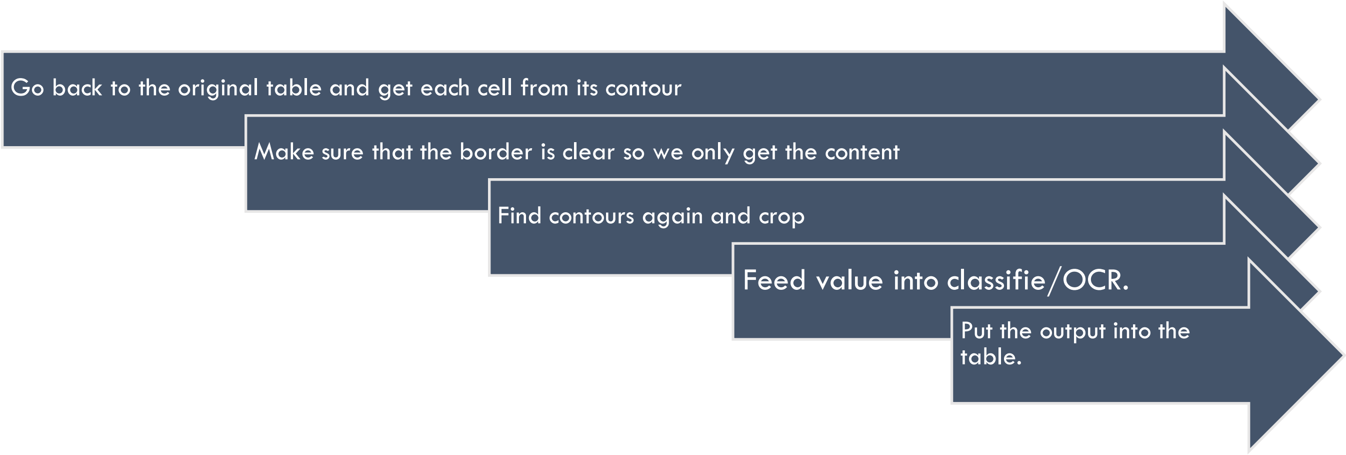
Detect horizontal and vertical lines through Morphology (Opening)

Get an empty table. Now we can find all the cells by finding the contours.

Mark the cell locations as the contours are found.

Construct the table in the program

Cell Content Extraction & Output (CellScanner.py)



Go back to the original table and get each cell from its contour

Make sure that the border is clear so we only get the content

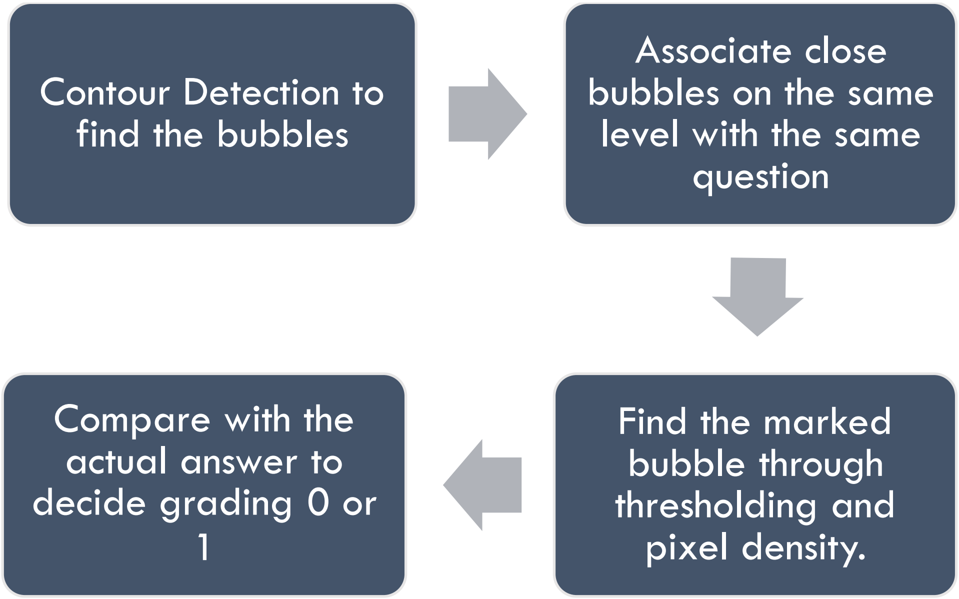
Find contours again and crop

Feed value into classifie/OCR.

Put the output into the table.

For some outputs (e.g. stacked lines) further image processing will be done before passing the result into the table. Assuming that the distance between the lines is zero their number can be naively computed without the need of murphology. For the student ID, contours will be found for each digit before passing the result to the classifier

Bubble Detection (Bubble Sheet.ipynb)



Contour Detection to find the bubbles

Associate close bubbles on the same level with the same question

Find the marked bubble through thresholding and pixel density.

Compare with the actual answer to decide grading 0 or 1

Non-primitve Functions

- OpenCV's findContours( )
- FNN Classifier
- Tesseract OCR or similar

Scientific Papers/References

[1] *Suzuki, S. and Abe, K., Topological Structural Analysis of Digitized Binary Images by Border Following. CVGIP 30 1, pp 32-46 (1985)*

This is the algorithm behind OpenCV's findContours.

Depending on the classifier and other functions that we may use, we might include more research papers.

Thank you.