



Project Report

Image Processing and Computer Vision

CMPN446 | Fall 2022

Cairo University – Faculty of Engineering

Credit-Hours System

Submitted to:

Eng. Ahmed Yousry

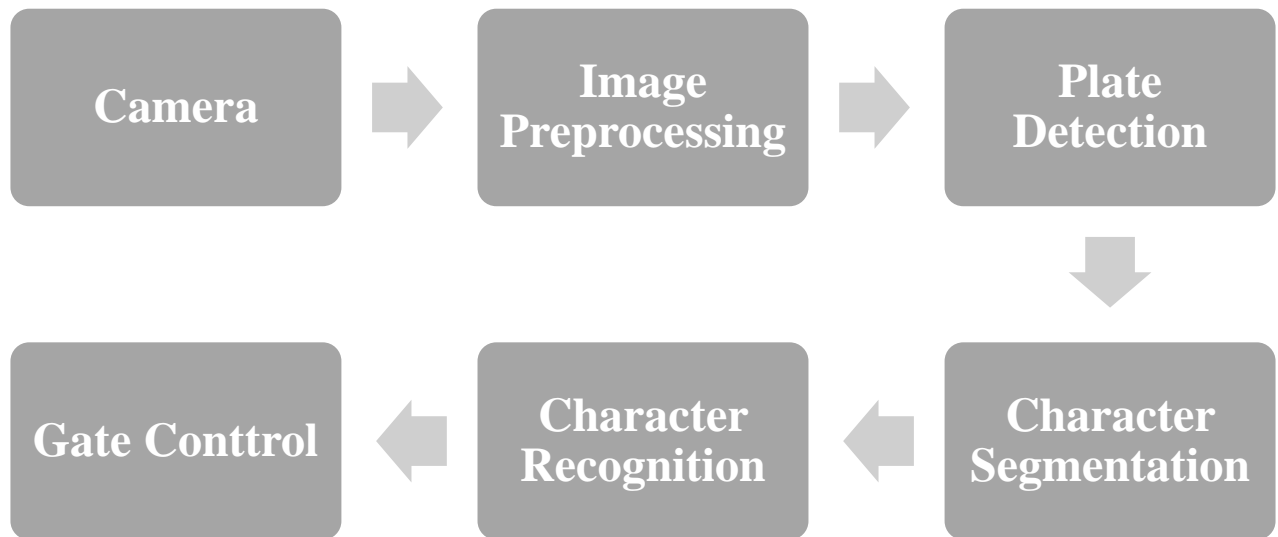
Team Members

<i>Name</i>	<i>ID</i>	<i>Major</i>
Abdallah Wael	1180376	CCE
Abdallah Ahmed	1180316	CCE
Karim Khalifa	1180478	MEE
Youssef Tarek	1180361	MEE

Project Idea and Need

The idea is that we will create an automatic plate number reader that detects the car plate number. It can be used in gates for mall to check the time this plate number spent in the parking. It can also be used for a company's or compound's parking to allow specific cars' plate number to enter the parking. This will save the time wasted in long car queues.

Block Diagram



Block Description

1. Camera

Output the image

2. Image Preprocessing

- 1) RGB to gray conversion
- 2) Noise reduction
- 3) Edge detection
- 4) Morphological operation
- 5) Filling holes
- 6) Masking
- 7) Opening and closing

3. Plate Detection

- 1) Contour finding to detect rectangle in the image
- 2) Plate dimension check

4. Character Segmentation

- 1) The plate is divided into three parts. Upper part, left part (contains numbers) and right part (contains characters)
- 2) Characters and numbers are cut into blocks with fixed size.

5. Character Recognition

These blocks are matched with characters stored in the database (alphabets and numbers).

6. Gate control

Responsible for opening the gate if it matches a plate number in the database.

Libraries used

- OCR detection (Easy OCR)
- Contour finding (OpenCV)

Performance and Accuracy:

Actually, we faced a plethora of problems in the matching and segmentation throughout the project.

The main problem is that the linkage between the phase of the plate detection and the phases of the segmentation and matching. We tried to use some models to help us with the matching stuff.

Coming to accuracy, due to the lack of good quality and actually a data set for Arabic plates were not easily available. We tried to capture as much as possible images from the street to help us, but it was not an easy as thought. So accuracy can be said to be around 70 percent. Or in other word more data set must be available to have a good sensible value for the accuracy.

Algorithms:

We used gaussian filter, morphological operations. We used to find contours. Edge detection. Segmentation to get plate and characters.

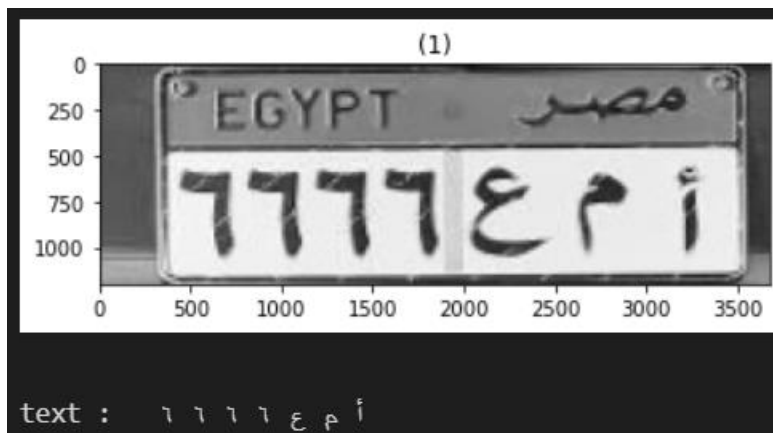
Work Division:

1. Abdallah Wael and Abdallah Ahmed worked on plate detection and post processing.
2. Youssef Tarek and karim Khalifa worked in character segmentation and character recognition.

Conclusion:

In a nutshell, we really struggled to get a good dataset online, so we needed to get the dataset manually by taking photos from the street. The quality wasn't as good as the English data so when it reached the character recognition it wasn't very accurate. As discussed with you, we took your advice, and we were committed to the Arabic plates project and spent a great effort to reach both the dataset and achieve a high accuracy and added to this it adding a model was not a big push. Hope you understand our situation and hope we will be rewarded on the effort spent.

Output Screenshots:



References

- 1) https://www.researchgate.net/publication/259143749_Automated_new_license_plate_recognition_in_Egypt
- 2) https://www.researchgate.net/publication/338645726_Egyptian_License_Plates_Recognition_System_Using_Morphological_Operations_and_Multi_Layered_Perceptron
- 3) <https://arxiv.org/ftp/arxiv/papers/2107/2107.11640.pdf>
- 4) <https://thescipub.com/pdf/jcssp.2014.961.969.pdf>
- 5) <https://www.kaggle.com/datasets/andrewmvd/car-plate-detection>
- 6) <https://universe.roboflow.com/yousef-gamal/egypt-car-plate/dataset/1>
- 7) <https://universe.roboflow.com/yousef-gamal/egypt-car-plate/dataset/2>
- 8) <https://universe.roboflow.com/yousef-gamal/egypt-car-plate/dataset/4>
- 9) <https://discordapp.com/channels/1041029676690329700/1058780615874449438/1058780764168261644>
- 10) <https://discordapp.com/channels/1041029676690329700/1058780615874449438/1058780645523996702>