

# License Plate Recognition + Interpretation

Sarim Qureshi (squires25), Tosan Egbesemirone (oegbes2), Daniel Levert (dlever2),  
Zakeria Jamal (zmjamal2), Mohammed Akif (makef3)

## Idea + Motivation

License plates are a huge part of identity and car ownership verification. Furthermore, to maintain law and order on the roads, cameras are often situated within locations such as speed cameras, traffic lights, and toll booths to capture images of cars. Therefore, rather than manually processing car information, the process of extracting license plates and getting more information about the associated car can be automated using Python and its powerful libraries.

## Target Audience

This project would likely be of interest to U.S. and State Department of Transportation as well as any other related entities.

## Methodology

This project is product-oriented and uses concepts such as edge-detection, gray-scaling, cropping, and automation testing (not taught in this class) to produce a solution. In this project, we will have a program that takes an input image of a car with a visible license plate. The image recognition program will then utilize the Python OpenCV library to transform the image to better extract the license plate. The program will then create a new image of the rectangular cropped portion that contains the license plate. The program will then use the Python Tesseract OCR library to read the text characters of the license plate image. Upon extracting the license plate's string, the program will then trigger an automation script using the Python Selenium WebDriver library to navigate to [FaxVin](#), which will in turn give us information about the car's make, model, year, VIN number, etc. The program will then finish by outputting the license plate number and the associated vehicle information received from [FaxVin](#).

### Constraints:

- Our inputted image's license plates will be limited to the State of Illinois
- The image containing the license plate must be from an angle where the plate is shown as a parallelogram
- Tesseract OCR is occasionally unable to interpret the text from an image with 100% accuracy (there may be a workaround to this)
- Due to Selenium WebDriver utilizing a particular browser for the automation script, we may be using either Chrome or FireFox

## Related Work

<https://medium.com/programming-fever/license-plate-recognition-using-opencv-python-7611f85cdd6c>

<https://www.geeksforgeeks.org/license-plate-recognition-with-opencv-and-tesseract-ocr/>

<https://www.pyimagesearch.com/2020/09/21/opencv-automatic-license-number-plate-recognition-anpr-with-python/>

<https://selenium-python.readthedocs.io/getting-started.html>

[https://youtu.be/0-4p\\_QgrdbE](https://youtu.be/0-4p_QgrdbE)