Simon Chau
Nichelle Chung
Peter Hung
Todd Larson
Alex Tan

CS 170 Group 10 - License Plate Reader

Introduction

To approach the problem for license plate detection, we utilize two papers that have similar qualities to our goal.

The report by Jason Grant shows that in order for the license plate detection to function properly, the cameras must have a quick shutter speed to allow precise capture of quick moving objects. This provides the ability to capture the license plates whilst moving quickly or from a specific angle. Grant uses Optical Character Recognition in a way to convert text or motion images into machine-encoded text. Grant further explains how there are four steps for license plate recognition. This includes Creating a Binary Image, Segment License Plate Bounding Box, Extracting Characters from the image, and Using Optical Character Recognition.

In the second report by Amr Badr, Mohamed M. Abdelwahab, Ahmed M. Thabet, and Ahmed M. Abdelsadek; they describe the automatic number plate recognition (ANPR). This acts as a mass surveillance method that uses optical character recognition on images to read the license plates on vehicles. Using ANPR, they are able to capture and store images and text from license plates. The goal for the ANPR to function at best capacity, it will locate the region that it should concentrate on and account for the license plate. The steps they authors use are Preprocessing, License plate localization, Character segmentation, and Character recognition.

Problem Space

Our problem space is image character recognition specifically for license plate character recognition for the detection of stolen vehicles. We use OpenALPR, which is an automatic license plate recognition API. In order to test the API we took multiple pictures of our own vehicle’s license plates from different angles and distances. We then make our own program to find the most likely license plate number out of all the possibilities given by OpenALPR. This is the solution we went with out of the many possible solutions to this problem.

Project Purpose

Our project is a license plate number reader. We are using image processing in order to read the license plate number. After we read the number, we will identify if the car corresponding to the license plate number is registered as stolen. We are using the OpenALPR to detect the top 10 detected license plate numbers from a picture. We then use our own program to determine the actual license plate number from the 10 possible. In hopes of using image processing, we expect the images to be clear and legible which will be able to match with any stolen vehicle reports. This is the solution we are taking in hopes of correctly detecting stolen vehicles.

California is ranked as the state that has the most cars stolen in the US. This program will aid law enforcement in finding more stolen cars. This will increase the rate at which cars are found and the amount that are found. This program will be a deterrent for car thefts as cars would be more easily identified as stolen. People should care as theft is the most common crime in the United States. As this is one of the most common recurring crimes, it would be helpful if more counter measures were taken. This alternative is an added on benefit in addition to what we already have for vehicle thefts.

Methodology

Our license plate detection consists of the following steps:

(1) Acquisition of License Plates
(2) Run OpenALPR, Using Image Processing
(3) Determination of License Plate Number
(4) Comparison of Detected License Plates to Dataset of Stolen License Plates

(1) Acquisition of License Plates

The license plates we obtained/used are of our own vehicles. This is for the ease of testing and would allow us to be more flexible with testing multiple vehicles. The main goal is to make sure we can detect the license plate. Using various pictures that vary in distances, cars, and angles, we expect to get confidence percentages on the correct license plates.

(2) OpenALPR:

First, OpenALPR uses the local binary pattern (LBP) algorithm to detect the license plate region. Normally LBP is used in facial recognition. However in this program, it is used to find the license plate in the picture. This is important to filter out all unnecessary portions of the picture.

It then goes through a binarization process to create multiple binary images of the license plate. This process uses the Wolf-Jolien method and Sauvola method to binarize pictures. This process helps more recognizable characters for easier recognition.

In order to recognize characters the API finds blobs that are within the width and height that license plate characters should be. This is done multiple times to search for both smaller and larger characters, which helps narrow down exactly where the license plate actually is in the image. The process then finds the final edges of the license plate to process.

In the case that the image is skewed or angled, OpenALPR detects the edges of the license plate and then de-skews the image to make it a standard size to prepare for character segmentation.

The character segmentation phase uses a vertical histogram to identify gaps between characters, which allows each character to be individually read and processed. Next is the Optical Character Recognition phase which computes the possible characters for each of the detected characters along with its confidences and is the final image processing step.

During post-processing a threshold is set and if a predicted character is below the threshold then it is not considered to be the correct character. This final phase determines the top 10 most possible character combinations of the license plate.

(3) Determination of License Plate Number:

The final result of the OpenALPR API is a top ten list of possible license plates. With the top result having the highest probability of correctness. Our program finds the most occurrences of one license plate number. We search the most occurrences first is because this eliminates outliers. If we just went with the best average confidence, the final license plate some of the time would not be the one in the image.

If the number of occurrences are the same, then we check the average of the correctness between the two license plate numbers. This gives us an overall better result of the license plate number.

We use multiple pictures of the same license plate to get each of the picture's top possibilities. We use multiple pictures because sometimes if there are obstructions or if the picture is at high angle, then Open ALPR won't output the list for that picture. We also rule out any impossibilities such as having too few characters/numbers or too many.

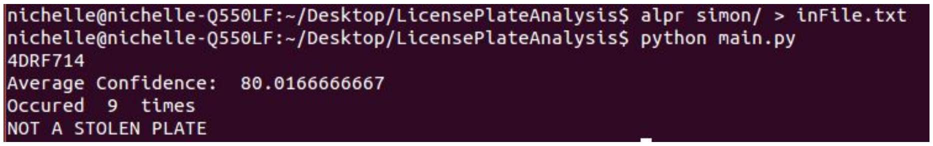
(4) License Plate Comparison after Determination

After weighing the possible license plates we find the most likely license plate number of all the images. Then take that number and see if it is in the stolen car database. For these testing purposes, we use only California License Plates. The comparisons will make sure the license plate that is being found is correctly matched to the database, being our own cars.

Setup

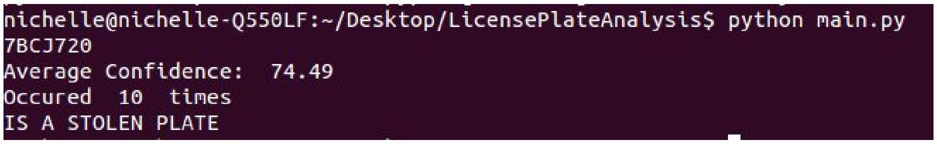
1. In this setup, we are assuming you are using an Ubuntu Linux OS with Ubuntu version 16.04+. If not, refer to section b.
    a. Enter: `sudo apt-get update && sudo apt-get install -y openalpr openalpr-daemon openalpr-utils libopenalpr-dev`
    b. Windows - https://github.com/openalpr/openalpr/wiki/Compilation-instructions-%28Windows%29
    c. OS X - https://github.com/openalpr/openalpr/wiki/Compilation-instructions-%28OS-X%29
    d. Ubuntu Linux version lower than 16.04 - https://github.com/openalpr/openalpr/wiki/Compilation-instructions-%28Ubuntu-Linux%29
2. Enter: `cd ~/Desktop`
3. Enter: `git clone https://github.com/phung001/LicensePlateAnalysis.git`
4. Enter: `cd LicensePlateAnalysis/`

5. Enter: alpr simon/ > inFile.txt (It may take a few seconds)
6. Enter: python main.py
    a. You should have an output looking like the following:
        i. This is the result of running the program on a license plate that is not reported as stolen.



```
nichelle@nichelle-Q550LF:~/Desktop/LicensePlateAnalysis$ alpr simon/ > inFile.txt
nichelle@nichelle-Q550LF:~/Desktop/LicensePlateAnalysis$ python main.py
4DRF714
Average Confidence:  80.0166666667
Occured  9  times
NOT A STOLEN PLATE
```

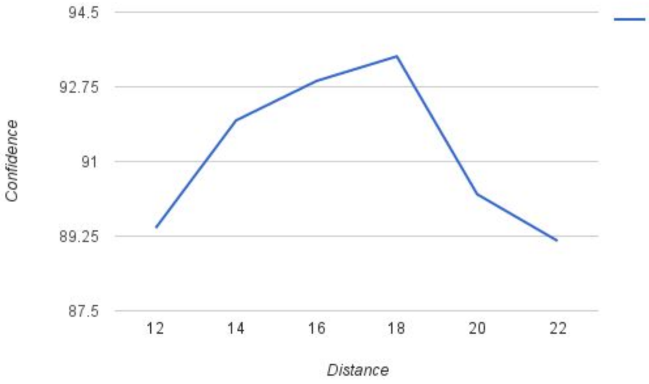
7. Enter: alpr nichelle/ > inFile.txt (It may take a few seconds)
8. Enter: python main.py
    a. You should have an output looking like the following:
        i. This is the result of running the program on a license plate that is reported as stolen.



```
nichelle@nichelle-Q550LF:~/Desktop/LicensePlateAnalysis$ python main.py
7BCJ720
Average Confidence:  74.49
Occured  10  times
IS A STOLEN PLATE
```

Results

This program functions properly as long as the license plate is in full view in the photos used. However, any possible hindrance with the vision of the license plate can affect the results from the image being processed. If the license plate is obstructed by a license plate frame or if the photo is taken at an angle that doesn't clearly show the license plate, the license plate number will most likely not be detected and the image is ignored. As long as the image of the license plate is in full view and clearly visible, then there should output a confidence percentage that correlates to the license plate. With our program, we will be able to obtain the top ten possible license plates that will each have a different corresponding confidence percentage. With these results we are expecting that the highest confidence percentage should be the correct license plate. Once the license plate is detected, we can determine if the vehicle was stolen or not.



We took 6 pictures from different distances and compared the confidences of the actual license plate. The graph above shows the resulting confidence in relation to the distance of the license plate. We found the best distance to be around 18 inches from the license plate. When we increase the sample size, we should expect a more defined bell-shaped curve.

Conclusion

Through the use of OpenALPR, we are able to detect the license plate with specific images of the license plate. With the prediction algorithm with OpenALPR, we gain some results that are the possible license plate in the image provided. The issue however, is that we must have a non-obstructed image of the license plate in order to obtain results that can be used. The other issue is that the picture or image must be a front shot that contains a complete full view of the license plate. Any angle or slight obstruction such as a license plate border, can result in nothing.

References:

These are the two papers we found that have relation to our project:
https://www3.nd.edu/~jgrant3/cw/alpr.pdf
http://inf.ucv.ro/~ami/index.php/ami/article/viewFile/388/351
OpenALPR:
https://github.com/openalpr/openalpr