Machine Learning Engineer Nanodegree

Capstone Proposal

Christel Ilaka Gipangu

October 4th, 2018

Automatic License Plate Recognition and Search

Domain Background

It's no longer a science fiction for a computer to be able to see and recognize object on an image and describe them. Nowadays, the field of computer vision is getting better and better and it is used in many applications like predicting lung cancer based on x-rays images or unlock our cellphones with Face-Id password. The combination of computer vision with other machine learning algorithms such as supervised and deep learning algorithms allows us to build powerful applications that make our lives better, efficient, and easier.

Problem Statement

Recently I was in my country (D.R. Congo) where the traffic was insane because the government agency that registers motor vehicles was investigating cars to catch to ones that didn't paid the tag fees. They do this manually by creating check points on streets; and thus, creating bottlenecks. My friend asked me if I could create an application that will be able to take the picture of a car, process it, identify the license plate number and then verify if the car had paid the fees or not; if not, the police could then stop car without jamming the traffic. I know similar system exist already but I think there's no one-size-fits-all solution for this problem. Therefore, I'm going to create an Automatic License Plate recognition and search system where the input will an image that will be processed to extract the Plate number and then search it into a database. Also, our plan is to pitch the governor of the city of Lubumbashi so that we could implement our system.

Datasets and Inputs

It is difficult to come across a license plate's dataset that is publicly available due to privacy issue and each country, state, or city has its own design. For this reason, we are going to create our own dataset of **250 car's images** collected from friends and family. Although these images will come in different shapes, once the license plate has been detected on an image during the preprocessing step, it will be extracted and resized to have a width of 420px. We will extract the training sets from the image bellow and this set will be used to train our base model; we downloaded the fonts that are closed to the characters on the license plate of the city of Lubumbashi (D.R.Congo) ([1]).

Figure: Characters to be extracted to create the training dataset

Solution Statement

The most challenging part of our system is to get the License Plate Number (LPN) in plain text that will be query into the database. The LPN of my country is different than for example the one of New York which is also different from the one of Boston, etc. Thus, the first challenge is to localize the region where the LPN is located. This will be done by leveraging a combination of image processing techniques. Once we have found the region of the image that contains the license plate, we will segment each character of the license plate and extract he feature vector associated with each character that will then be pass to a machine learning model to classify and recognize the character. After this, we will have our final LPN stored as a plain text variable that will be used to query the database.

[1] http://

Benchmark Model

I will use a Logistic Regression model as the benchmark for this problem and the result of this model will be compared to the final and optimized model.

Evaluation Metrics

The metrics that we will use here is **F-beta score** because we want to be precise when classifying the characters of a license plate otherwise the query to database will return false information. For this reason, we will use beta = 0.5 since with this value we have more emphasis on precision.

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

Project Design

We need first to acquire a picture of the car that has a license plate that we want to detect and recognize. In real world, this is done by using a camera and a trigger system; we will skip this process and assume that we already have our car image in a folder. Now we start by localizing the region on the image that contain the license plate by using image processing techniques. Once we have the region containing the license plate, we then segment each character from the license plate background; here we could use techniques such as adaptive thresholding or scissoring to cut the characters from the license plate. Finally, each character will be quantified via feature extraction; for this we will the block binary pixel sum descriptor to get the feature vectors that will be used into our SVM classifier. The output of the classifier will be the license plate in plain text that will be used to query the database.

References

1. YenChing Chang et al (2013), License Plate Character Recognition Using Block-Binary-Pixel-Sum Features, https://www.atlantis-press.com/proceedings/iccnce-13/6483
2. Adrian Rosebrock, Pratical Python and OpenCV + Case Studies, 3rd Edition 2016. https://www.pyimagesearch.com/practical-python-opencv/
3. Baoguang Shi, (2017), Detecting and Recognizing Text in Natural Images. From https://bit.ly/2xWSaOA
4. Vishnu Sundaresan, Jasper Lin, Recognizing Handwritten Digits and Characters, http://cs231n.stanford.edu/reports/2015/pdfs/vishnu_final.pdf