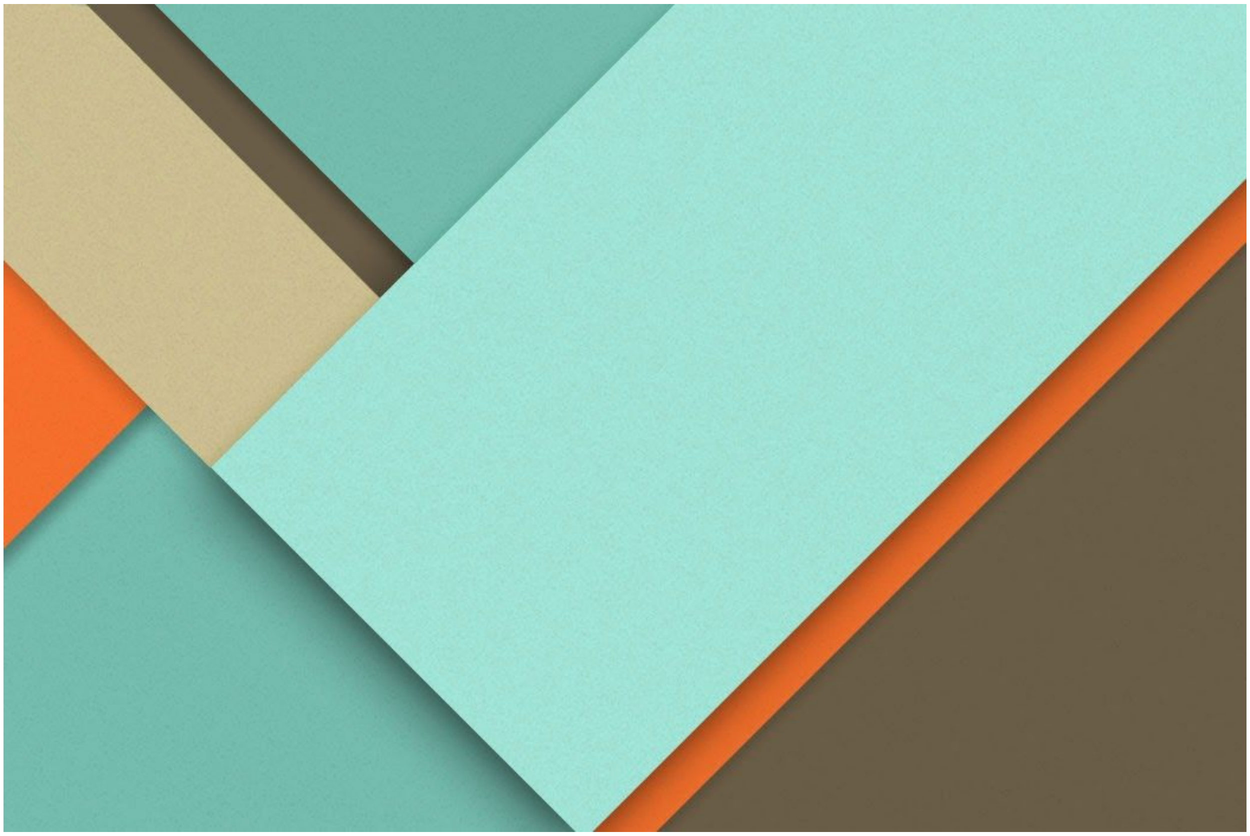


# APS 360 - Project Proposal

06.14.2020


1002965026 Muyi Chen

1002839690 Jianyu Wen

1003149416 Yuanzhuo Wang

1002147651 Bowen Wu

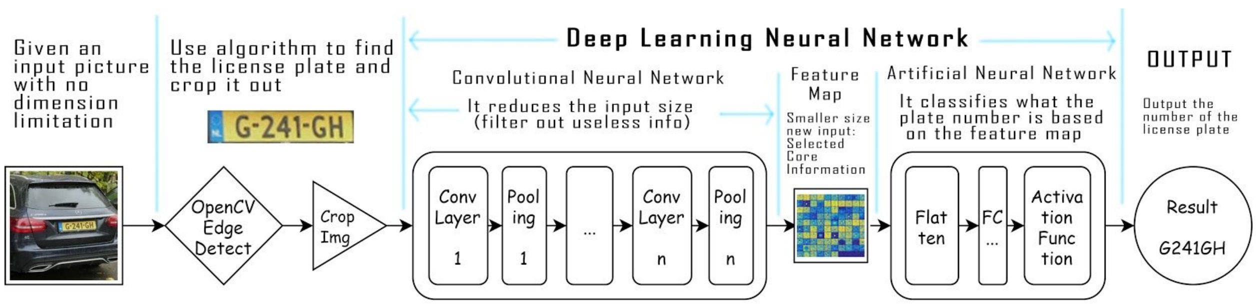

APS360 - word count: 1387

## Introduction

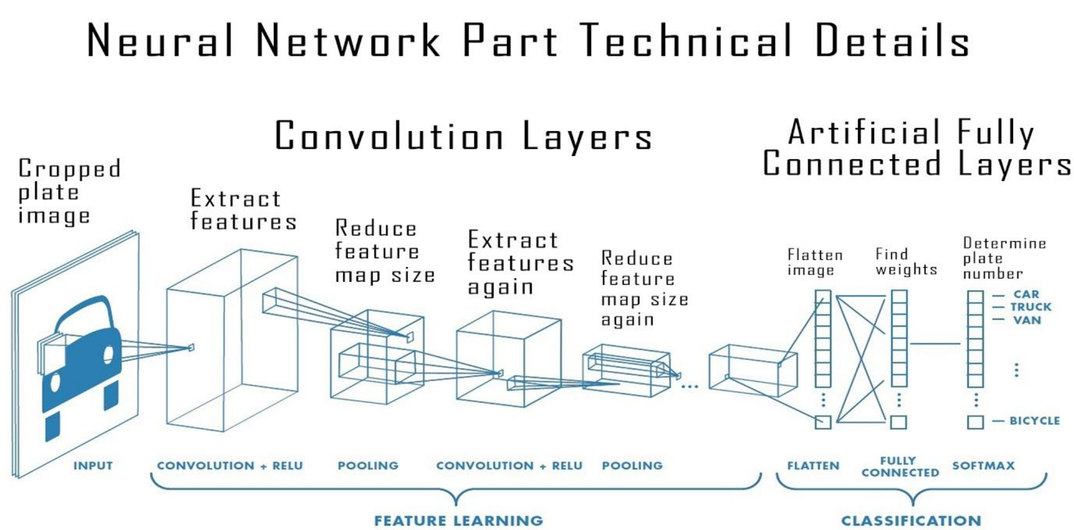
Vehicles have proven to be one of the most useful tools in daily lives. License plates are used to identify unique vehicles in the legal system. As camera technologies develop, it has shown great potential for Automatic License-Plate Recognition (ALPR) to replace manual plate-reading. As a result, public agencies such as the police are already making use of this technology for a variety of tasks. [1]

We propose to make an end-to-end ALPR system that, when given images, records the plate numbers that were shown in that image. We believe the trained model can be applied to both authority and civil utilizations. Also, since ALPR involves reading numbers, localising the ROI and individualising symbols, we believe it is a valid project to be done in machine learning.

## Figure




[2] [3]




[4]

## Background & Related Work

Reading plate-numbers automatically has been a popular research direction since vehicles had been around. Apparently, there were efforts made to make the task of reading plate-numbers automatically as early as the 1980s. [5] Since the new millennium, there has been an increasing number of research using machine learning. As a result, China has been utilizing automatic license plate recognition technologies on detecting vehicle numbers since the late 2000s to catch people driving restricted vehicles on a weekday of the week. A recent paper shows that with their end-to-end lightweight CNN, 95% accuracy can be achieved on live videos with Chinese license plates. [6]

## Data Processing

All the datasets we will be using are provided from a website that is offering Automatic Number-plate Recognition service.[7] The dataset includes a large amount of plate images with the plate's number as their labels.

The first group of dataset contains a group of low resolution and quality images that only contains license plates.[8] The dataset groups images into 8 different categories. Each license plate in each group has more than one picture in different conditions. Also, there is a spreadsheet csv containing image paths and plate numbers. All group names ending with “02” have test/validation data corresponding to the same name group ending with “01”. However, we need to split non-training data into test and validation based on our train:validation:test ratio.

The second group of dataset contains car pictures taken from the street.[9] This is used to train our model to spot the license plate from a street view picture. This dataset has all picture’s plate labels in text files, which we need to extract together with the paths, and create a spreadsheet csv with, in the same format as the previous dataset.[8] We would be loading data by taking in these csv files.

## Architecture

The first step of handling the input image is to locate where the license plate is. Usually the plate is not taken from a clear angle, and therefore hard to be recognized. Libraries such as OpenCV can detect edges of the plate and characters, as well as cropping the original image into a smaller plate image.

The processed image will then be parsed into a CNN which extracts the features of the image and reduces the size of the input. The CNN is expected to have at least 3 convolutional layers and 3 pooling layers, as the previous step already reduced the dimension of the input. Then the feature maps will be parsed into an ANN with a few numbers of fully-connected layers to make a final judgement.

Depending on the accuracy, we might consider trying YOLO v3 as the CNN part of the model, which has 106 layer fully convolutional underlying architecture. [10]

## Baseline Model

A baseline model is used to sense the quality of our network, which in this case the best model will be Random Forest. Since our problem is a classification problem that requires identifying the plate's characters from images, and each plate's common character range is from 26 alphabets and 0-9 numbers. This gives it 36 classes to choose from. Random forest is made up of decision trees at random sampling, which are "a series of questions about data in each node that lead to predicted class".[11] Therefore, implementing a large random forest to classify our 36 classes would be the choice due to its simple problem solving on classifications.[11]

## Ethical Considerations

Limitation of model

- Digit and English alphabet recognition only
- not expected to recognize region

Limitation of Training data

- Only plates with digit and English alphabet were collected
- Collected several years ago

Ethical issues

- potential to track daily activities, compromising privacy safety
- Misidentification causing trouble to unrelated people

## Project Plan

Communication:

- The team will be communicating mainly online.
- An update is required for each team member once any changes are made to the previous version of code.
- A weekly meeting will be held on each Friday night to discuss progress made during weekdays.
- The communication will be based on WeChat, facebook as well as other online chat services.
- In case of emergency, phone numbers will be required from each member.
- The main tasks will be split into 2 parts: Image processing and Model training.
- Each part will be completed by 2 team members, and if one member cannot be reached, the other would gain his responsibility to overtake the work remaining.

Version Control:

- The code version control will be based on Github.
- Each commit must include an update log file.
- After each commit, the author is required to notify others in the group chat.
- Local copies of the latest versions are required on each members' terminal.

Task split and responsibilities:

1. Image Processing:

| Jianyu Wen | Bowen Wu |
|---|---|
| OpenCV image processing, including edge detection, plate locating, license plate image cropping | Image import handling, including data collection, data labeling, image pre-processing |
| Team management, including managing milestones, communicating with another team and status report | Image export handling, including tensor export |

2. Model training:

| Together | Muyi Chen | Yuanzhuo Wang |
|---|---|---|
| Baseline model development | Paper research on related models | Local training and hyperparameter research |
| CNN and ANN model research and selection | Determine best fit Neural Network models | Team management, including managing milestones, communicating with another team and status report |

Assumed Deadlines:

July 3 - Baseline model available, dataset loaded and ready to use

July 10 - OpenCV available; Baseline model can generate results with our model's prototype structure ready

July 17 - Prediction with plate-only images

July 24 - Tuning hyperparameter to reach acceptable accuracy

July 31 - Crop out plate from car pictures using OpenCV

August 7 - Connect OpenCV's cropped plate to our model

August 14 - Further tuning for better test accuracy; Prepare for presentation

## Risk Register

Examining the process, we discover some potential risks that could hinder the success.

Any teammates could drop the course due to unexpected reasons: although before formation, we discussed this topic to ensure all are interested in this course and unlikely to give up. Despite that, we also considered safety measures: increasing workload for reminders while looking for potential new members interested in our project to fill in the gap.

Training is time-consuming due to unexpected complexity among the data. To stay in plan, we consider adopting transfer learning to extract features using pre-trained Networks. Also we would consider purchasing Colab pro (or other equivalent product) to gain access to extra hardware resources.

Machine learning projects are data driven. Quality of the data is crucial. We borrow the open source dataset from an Plate Recognition project, perhaps combine with similar project dataset online to mitigate the bias, and take more pictures ourselves.

## Link to Github or Colab Notebook

https://github.com/SauryCC/APS360-Project

## References


[1] Office of the Privacy Commissioner of Canada, “Privacy Tech-Know Blog: Who’s Watching Where You're Driving? - Office of the Privacy Commissioner of Canada,” 13-Jun-2017. [Online]. Available: https://www.priv.gc.ca/en/blog/20170613/. [Accessed: 13-Jun-2020]

[2] “Olav’s Dutch license plates - Number plates of the Netherlands.” [Online]. Available: http://www.olavsplates.com/netherlands.html. [Accessed: 14-Jun-2020]

[3] M. Amrani, F. Jiang, X. Tan, M. Zou, and X. He, “Fig. 3 Deep features visualization: (a) output feature maps and (b).,” *ResearchGate*, 17-Oct-2017. [Online]. Available: https://www.researchgate.net/figure/Deep-features-visualization-a-output-feature-maps-and-b-convolution-kernels-of-the_fig3_320461081. [Accessed: 14-Jun-2020]

[4] Prabhu, “Understanding of Convolutional Neural Network (CNN) — Deep Learning,” Mar. 2018 [Online]. Available: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148. [Accessed: 14-Jun-2020]

[5] “History of ANPR - ANPR Internatonal,” *ANPR Internatonal*. [Online]. Available: http://www.anpr-international.com/history-of-anpr/. [Accessed: 13-Jun-2020]

[6] A. G. Sergey Zherzdev, “LPRNet: License Plate Recognition via Deep Neural Networks,” Jun. 2018 [Online]. Available: https://arxiv.org/abs/1806.10447. [Accessed: 13-Jun-2020]

[7] “Number Plate Datasets | Plate Recognizer ANPR,” *Plate Recognizer ANPR*. [Online]. Available: https://platerecognizer.com/number-plate-datasets/. [Accessed: 14-Jun-2020]

[8] “Holistic Recognition of Low Quality License Plates by CNN using Track Annotated Data [IWT4S-AVSS 2017] | Traffic Research.” [Online]. Available: https://medusa.fit.vutbr.cz/traffic/research-topics/general-traffic-analysis/holistic-recognition-of-low-quality-license-plates-by-cnn-using-track-annotated-data-iwt4s-avss-2017/. [Accessed: 14-Jun-2020]

[9] openalpr, “openalpr/benchmarks,” *GitHub*. [Online]. Available: https://github.com/openalpr/benchmarks. [Accessed: 14-Jun-2020]

[10] A. Kathuria, “What’s new in YOLO v3? - Towards Data Science,” Apr. 2018 [Online]. Available: https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b. [Accessed: 14-Jun-2020]

[11] W. Koehrsen, “An Implementation and Explanation of the Random Forest in Python,” Aug. 2018 [Online]. Available: https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76. [Accessed: 14-Jun-2020]