

# Vehicle Number Plate Recognition Using Optical Character Recognition

Gui-Han Go

*Department of Computer Science  
Georgetown University  
Washington, District of Columbia  
gg626@georgetown.edu*

Chia-Hsuan Hsieh

*Department of Computer Science  
Georgetown University  
Washington, District of Columbia  
ch1165@georgetown.edu*

Kevin Kyunggeun Jang

*Department of Computer Science  
Georgetown University  
Washington, District of Columbia  
kj460@georgetown.edu*

**Abstract**—Optical Character Recognition (OCR) is one of the key technologies in Automatic Number-Plate Recognition (ANPR). In this study, our team is building a machine learning model that implements OCR system to complete the character recognition tasks on the vehicles number plate images. More specifically, our group is applying the concept of both a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) over Recurrent Neural Network (RNN) to build a model that can accurately recognize characters in the image of a number plate.

## I. INTRODUCTION

As years pass by, things that were done manually with human's hands gets automated and digitized by computers. As a result, the importance and significance of technology increases and often times, it enormously benefits people's daily lives. Optical Character Recognition (OCR), [1] the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo or from subtitle text superimposed on an image, is one well-known example. Through OCR, manual works like data entry for business documents (e.g. check, passport, invoice, bank statement and receipt), automatic number plate recognition, and many others are now automated by computers. This automation is hugely successful in processing a large amount of works in a very short period of time, but never be more accurate than the works that are carefully done by humans. In other words, the quantity and speed of works is now guaranteed, but the quality of works is not yet perfect. As an instance, Automatic Number-Plate Recognition (ANPR) still have the following difficulties: [2] blurry images caused by a motion blur, poor lighting, an object obscuring part of the plate, and many others, which still requires humans attention to complete the recognition process. Therefore, further researches and studies are essential to overcome that flaws. In this project, our group mainly focuses on applying OCR in recognizing the characters in the image of vehicle number plates and targeting to get the accuracy over 90%. Contrary to the previous approaches, our team is implementing the concept of both Convolutional Neural Network (CNN) with two convolution layers and an average pooling operation, and

Long Short-Term Memory (LSTM) over Recurrent Neural Network (RNN).

## II. RELATED WORK

- Character Recognition in Automatic Vehicle License Plate Recognition by Anuj Kumar [3] Unlike our project, this study used videos of a moving car as dataset. Their approach on data processing was first convert the video into frames, select key frames, and extract the plate image from the image of a vehicle. This image will be passed to their character recognition system to output the plate numbers. In their training process, Artificial Neural Network (ANN) was used to extract the pixels in the number plate image. Then, a standard back-propagation learning algorithm is used for training and testing the model. In conclusion, they could not solve the ambiguity problem between two similar looking characters such as o and 0, I and 1, B and 8, C and G.
- Number Plate Recognition Using OCR Technique by Er. Kavneet Kaur and Vijay Kumar Banga [4] This study used the template matching approach to recognize the characters in a plate. Instead of building a model using a neural network, this group simply selected the characters from the image by cropping the characters and used their noise removal technique to increase the readability of the characters. They concluded that their approach was not effective especially when there is noise in image and on recognizing characters printed with different font types.

## III. DATASETS

For this project, our group used the artificially generated dataset that are very similar to the real world vehicle number plates. This dataset were downloaded from Supervisely website. The dataset contains total of 22,768 files, where half of the files are in JSON format and another half in PNG format. Out of 22,768 files, our group decided to use 21,644 files to train the models, and the rest (1,124 files) to test the models.

### A. File Types

- JSON : This file contains the information on the label (plate number), size (width and height) and tag (train or

test) of the given number plate. The above information will be used as the attributes for the given number plate.

- PNG : This file is the artificially generated image of the given number plate.

### B. Data Structure

There are three classes created to store the dataset.

- TrainTestDataSet : This class contains the DataSet object for both training and testing datasets. The build\_train\_test\_dataset function in this class can be called to build the datasets.
- DataSet : This class contains the list of DataElement object for a single dataset (training or testing). The build\_dataset function can be called to read and parse the given data files and create and store the DataElement object for each data file.
- DataElement : This class contains the variables for each attribute obtained from a single data file. There are total of five variables in each DataElement object, which are:
  - 1) label : label (plate number) of a given number plate
  - 2) height : measured height of a given number plate
  - 3) width : measured width of a given number plate
  - 4) label\_length : number of characters in a given number plate
  - 5) img : parsed image file

### C. Data Cleanliness

The cleanliness of the data were measured to avoid using any data with flaws. The following values were used to measure the cleanliness.

- Data Redundancy : There may exist redundant data in both training and testing dataset. This can result giving more weights to this redundant data while training the model. Therefore, our group should investigate, detect and remove redundant data from the combined dataset. This operation will improve data quality and make data analysis process more accurate.
- Data Missing : There may exist data that are missing key attribute values. Our group will remove those data to improve the overall quality of the dataset.
- Noise Data : Our group put an effort to reduce data with noise attribute values from collected data as much as possible to really increase the accuracy and preciseness on our predictions.

Due to the above reasons, our group will create functions to clean the combined dataset to maximize the overall quality of our analyses. However, surprisingly, the dataset were already clean enough. Further analysis on this will be explained below.

### D. Data Cleaning Process Logic

The following are the steps our group followed to resolve and measure the cleanliness issue on the dataset:

- 1) In the process of reading and parsing each data file, verify the cleanliness of this data using the valid\_json function in DataSet class using the below steps.

- 2) Check if the given data has any missing attribute values (label and size of the given number plate).
- 3) Check if the given data has any noisy attribute values. In our case, the given data is considered as noisy if any of the following conditions are not satisfied: the number of characters in a label must be eight, and the size of the plate must be 152 by 34 (width by height).
- 4) Check if the given data already exists in the dataset.
- 5) If the given data violates any of the above steps from step 2) to 4), our group dropped this data from the dataset.

### E. Figures and Tables

a) *Positioning Figures and Tables*: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. 1”, even at the beginning of a sentence.

TABLE I  
TABLE TYPE STYLES

Table Head	Table Column Head		
	Table column subhead	Subhead	Subhead
copy	More table copy <sup>a</sup>		

<sup>a</sup>Sample of a Table footnote.

Fig. 1. Example of a figure caption.

### REFERENCES

- [1] “Optical character recognition”, Wikipedia, The Free ENcyclopedia. Wikipedia, The Free Encyclopedia, 28 Nov. 2018. Web. 30 Nov. 2018.
- [2] “Automatic number-plate recognition”, Wikipedia, The Free ENcyclopedia. Wikipedia, The Free Encyclopedia, 21 Nov. 2018. Web. 03 Dec. 2018.
- [3] Anuj Kumar, “Character Recognition in Automatic Vehicle License Plate Recognition”, in International Journal of Advanced Research in Computer Science and Software Engineering. 2016, vol. 6.
- [4] Er. Kavneet Kaur, Vijay Kumar Banga, “Number Plate Recognition Using OCR Technique”, in IJRET: International Journal of Research in Engineering and Technology. 2013, vol. 2.
- [5] “Latest Deep Learning OCR with Keras and Supervisely in 15 minutes”, Hacker Noon. Supervisely.ly, 2 Nov 2017. Web. 28 Nov 2018.