Vehicle Number Plate Recognition Using Optical Character Recognition

Gui-Han Go (gg626), Chia-Hsuan Hsieh (ch1165); Kyunggeun Jang (kj460)
Department of Computer Science, Georgetown University

GEORGETOWN UNIVERSITY

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 the 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.

INTRODUCTION

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.

This automation is hugely successful in processing a large number of works in a very short period of time, but never be more accurate than the works that are carefully done by humans. 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. Further researches and studies are essential to overcome that flaws.

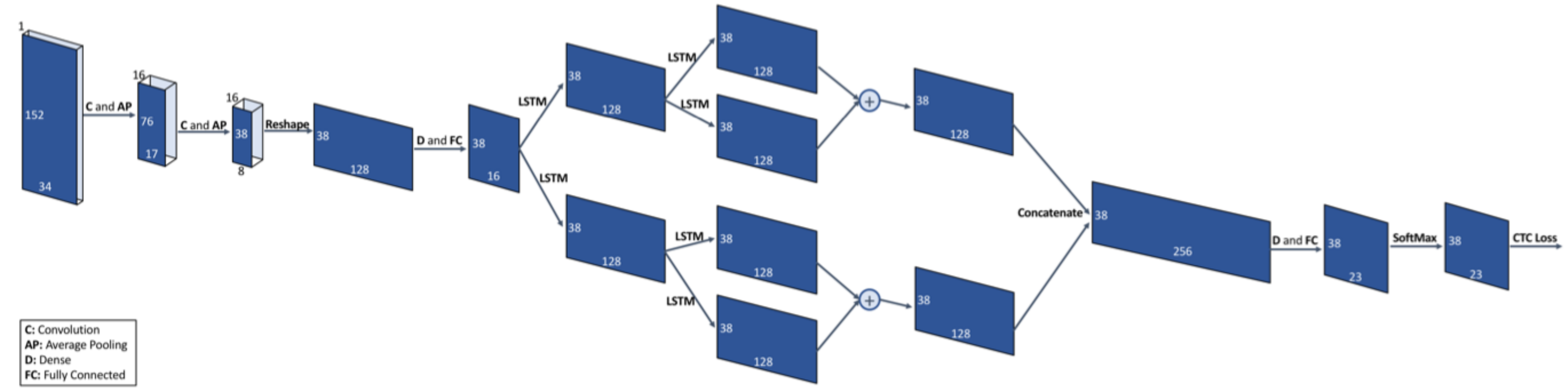
In this project, our group mainly focuses on **applying OCR in recognizing the characters in the image of vehicle number plates** and targeting to get the accuracy of over 90%. Contrary to the previous approaches, our team is implementing the concept of both Convolutional Neural Network (CNN) with two convolution layers and an average pooling operation, and Long Short-Term Memory (LSTM) over Recurrent Neural Network (RNN).

DATASETS

An artificially generated dataset that are very similar to the real world vehicle number plates.

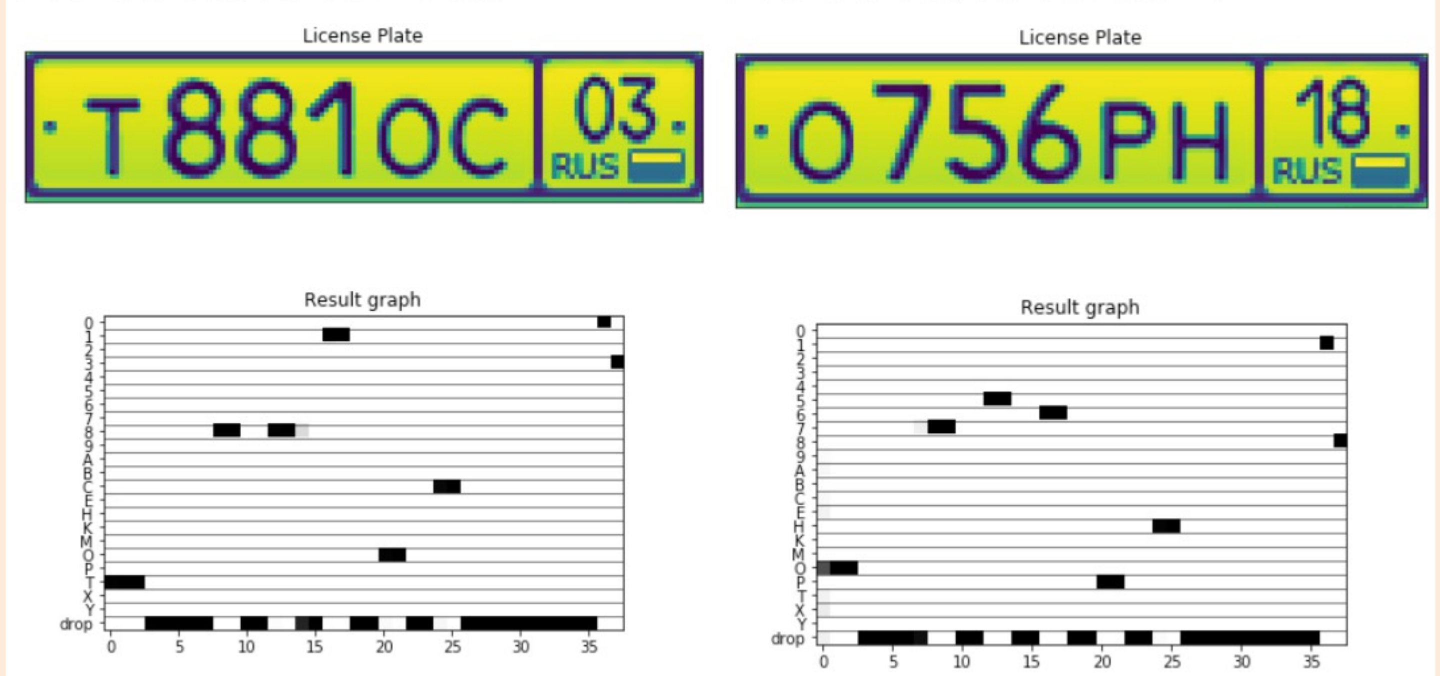
- **Source**: downloaded from 'Supervisely' website.
- **Size**: total of 22,764 data (JSON format + PNG format).
- **Split**: use 11,382 data to train the models, and the other 11,382 data to test the models.
- **Data structure**:
  - *TrainTestDataSet*: DataSet object for both training and testing datasets.
  - *DataSet:* DataElement object for a single dataset
  - *DataElement:* Contains variables for each attribute obtained from a single data file.
- **Cleaning logic**:
  - Check if the given data has any missing attribute values (label and size of the given number plate).
  - The number of characters in a label must be eight, and the size of the plate must be 152 by 34 (width by height).
  - Check if the given data already exists in the dataset.

STRUCTURE OF OUR MODEL



RESULT

- **There are total 22,764 dataset and we split it half and half.**
- **Training**: 11,382 dataset.
- **Testing**: 11,382 dataset
- **Accuracy**: 11,344 of 11,382 are correct. (99.67%)
- **Examples of result**



DISCUSSION

There is 4 kind of loss. Most of the misclassified license plate containing M in the beginning. Our model will misclassify it to H or K. We also misclassified some O to C.

1) M-H case (33 cases)

Actual	Predict
M811CO23	H811CO23
M485PX01	H485PX01

2) M-K case (2 cases)

Actual	Predict
M152KY86	K152KY86
M155EP17	K155EP17

3) O-C case (2 cases)

Actual	Predict
O611AK84	C611AK84
O611HC07	C611HC07

4) Etc. (1 case)

Actual	Predict
K645TA74	K445TA74

Below is the comparison table for the experiments. The different choices of the number and the size of pooling affect the accuracy of the 1-2-1-2-1 structure a lot. However, it doesn't affect 1-2-4-2-1 that much. For filters equal to 64, 128, or 256, the accuracy for 1-2-4-2-1 is equal to 99%

Layer Structure		1-2-1-2-1	1-2-1-2-1	1-2-4-2-1	1-2-4-2-1	1-2-4-2-1	1-2-4-2-1
CNN	filters	16	16	16	16	16	16
	Pooling times	2	1	2	2	1	2
	Pooling size	(2, 2)	(4, 4)	(2, 2)	(2, 2)	(4, 4)	(2, 2)
RNN (LSTM)	filters	128	128	64	128	128	256
Accuracy		86%	97%	99%	99%	97%	99%

CONCLUSIONS

The very important part of the model is **making pairs of the LSTM in RNN process.** One of them is forward and the other one is backward. We then add them up or do the concatenation. We would get a very different and bad accuracy if we don't have the backward LSTM layer.

Our **future work** is to make the design of the model simpler and more efficient so that we can run the dataset faster. In addition, we want to apply more different kind of RNN model and the activation function in this model to see whether we can make the accuracy higher.

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.

ANLY-590
Final Project

Gui-Han Go (gg626)
Chia-Hsuan Hsieh (ch1165)
Kyunggeun Jang (kj460)