Traffic Sign Localization and Detection

Final Project Report

University of Toronto

Faculty of Applied Science and Engineering

Artificial Intelligence Fundamentals

April 9th, 2020

Github Project repository: https://github.com/Louis-He/APS360_Project

Colab Link:

https://colab.research.google.com/drive/1k1MeodC3BQhEBct4RtRXZ_L836czf1SB

Team 02:

Siwei He

Xinru Li

Jiani Li

Qihan Liu

Word count: 2452

Introduction

There are around 2700 people die every year because of car crashes in Canada, reported by the *Hamiltonlawyer* [1]. According to CNN news, the 8th most common death reason in the world contributes to traffic-related accidents. Among all reasons towards the traffic accident, not following the traffic signs is one of the leading causes for car accidents [1]. Therefore, our motivation for this project is to develop a safer and more reliable driving assistant system, and further, to assist with the future implementation of autonomous driving systems. The goal of our project is to detect and recognize the upcoming traffic signs from the driver's driving recorder quickly and accurately. Machine learning is a reasonable approach for this project since we are able to train a model to learn the pattern of each traffic sign and therefore distinguish every sign with low delay and high accuracy.

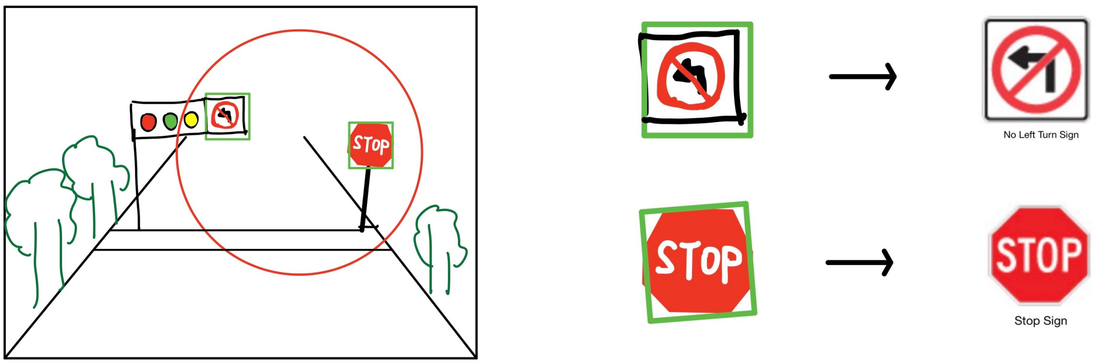


Figure 1. Goal -- localize and identify the traffic sign on a driving recorder

Ethical consideration

As an essential part of developing driver assistant and autonomous cars systems, satisfying the speed of the image process and the detection accuracy at the same time becomes the biggest challenge for our project. Either large delay of detection or a large detection loss would possibly mislead the driver and therefore results in receiving a fine, adding demerit points, or even worse, causing traffic incidents. Another possible issue that may arise for our project is that some drivers may rely too much on our traffic sign system and if there are any errors for our design, the consequence would also be severe.

Additionally, another challenge for our model is that it should work in any environment conditions, including night time and extreme weather such as storming or snowstorm. Furthermore, the training dataset for the situations mentioned above is rare online, so we might have to collect those ourselves.

Background and related work

The detection and recognition of traffic signs is a popular machine learning topic as it helps to improve driving safety and assist auto driving. We studied works related to this area to lay the foundation and improve our own project.

A research group from China focused on the detection and localization of traffic signs. They developed an algorithm combined “white balance, color image enhancement, and affine transform correction” [2]. They applied this method and successfully detected traffic signs under real world conditions, including extreme weather and being obscured. We can study and develop our own method for detection and localization based on this previous work. Another team in Slovak Republic designed different ANN models to recognize different traffic signs, and “the type of all used ANNs is a multilayer perceptron with one hidden layer” [3]. However the type of traffic signs that can be classified is limited and the accuracy is uncertain. As for our own project, we plan to focus on the classification of all types of traffic signs in Canada.

Data processing

Content	**Data processing skill**	**Data size**	**Remark**
Driving recorder	Python video processing related library	25,728 images with size (1280 * 720 pixels)	Collected from real world driving recorder
Traffic sign Augmentation	skimage lib, rotation, adding noise, flipping, cropping, scaling	8400 images for stage 1, and 5304 images for stage 2. Detailed calculation in Table 2.	Used to train CNN model that should identify traffic signs
Traffic sign Cropped from recorder	Python matplotlib library	around 200 images	Used to validate and test the classification model

Table 1. Summary of the data processing progress

The project contains two models, which are localization model and classification model respectively. We applied different data processing approaches to each model.

In the localization model, we have a dataset source from our driving recorder. For the data from the driving recorder, one photo was captured every half second by a python script using video related libraries for each video and they were resized to 1280 x 720.

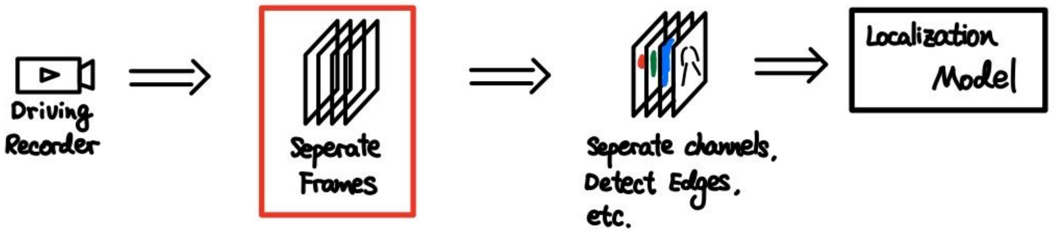


Figure 2. Driving recorder processing steps

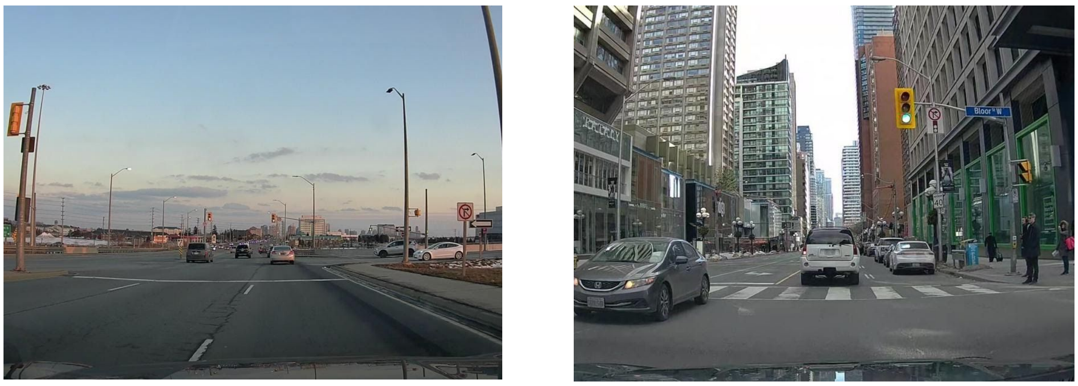


Figure 3(a) (b). driving recorder data samples

In the classification model, we downloaded the high resolution traffic signs from Ontario MTO[4]. Considering that traffic signs in real life are not always clear and standardized (e.g. distortion, faded colors), we generated training data by rotating, changing colours, three different levels of scaling, cropping and adding random noise from clear signs using skimage (details in Table 2), which will increase the accuracy of the model since more data are feeded.

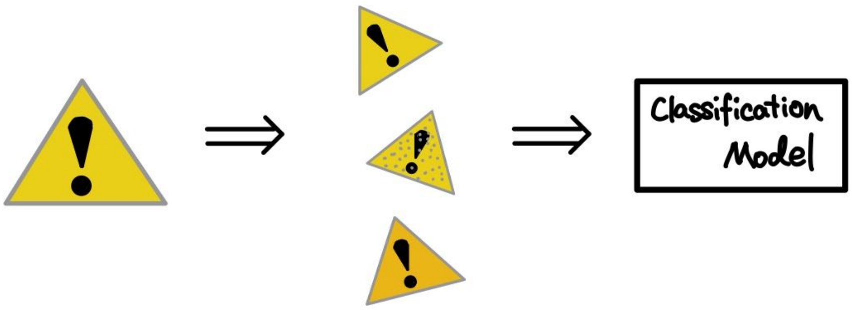


Figure 4. data argumentation from standard images

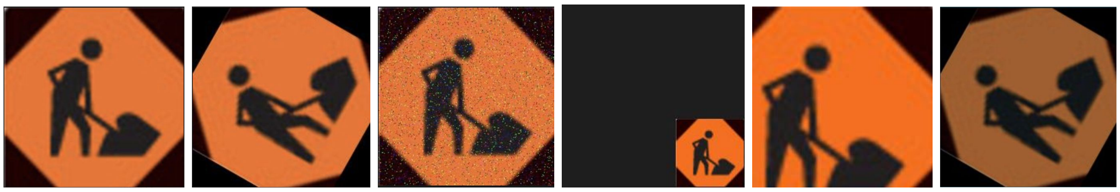


Figure 5(a)-5(e). Original sign, rotated sign, sign with random noise, scaling, cropping, and colour adjustment for "construction ahead" temporary sign

	Enlarge	Compres s	Horizontally Flip	Vertically Flip	Rotation	Noise	Change Color	Total Amount of Samples
Traning set stage 1	4	19	1	1	4	10	9	(1* + 4 + 19) x (1 + 1 + 4 + 10 + 9) x 14 = 8400
Validation set stage 1	4	19	1	1	2	3	5	(1 + 4 + 19) x (1 + 1 + 2 + 3 + 5) x 14 = 4032
Training set stage 2	4	19	0	1	4	3	9	(1 + 4 + 19) x (1 + 4 + 3 + 9) x 13 = 5304
Validation set stage 2	Real life samples and data augmentation based on that							around 2500

*: The 1 represents the original picture.

Table 2: Dataset Distribution

To help validate and test our classification model, we cropped traffic signs from the frames we captured from our driving recorder and resize them to 224 * 224 * 3 to match the input size of the classification model. The dataset contains around 8400 training pictures with regulation, temporary, warning and other signs for stage 1, and 5304 training pictures for stage 2. It is available to download from github since we opensouced it. (available via: https://github.com/Louis-He/trafficsign_dataset)

Figure 6(a)-6(c). image samples cropped from the driving recorder.

Baseline Model

We proposed a combination of sliding window and CNN as our baseline model. The sliding window algorithm is used for detecting the traffic sign. The “window” is moved around the image and makes small crops. Those crops will then be forwarded into CNN to classify. In the CNN model, the data will be processed through two convolutional layers, followed by two fully-connected layers and then passed to output layers. This CNN will be trained to categorize traffic signs into different types, such as warning and regulation.

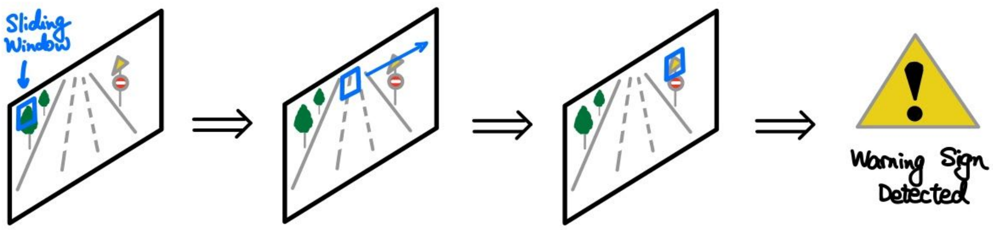


Figure 7. working principle of baseline model (using sliding window and CNN)

However, the sliding window algorithm requires a longer time to process as it moves around the images and makes crops compared to our final model.

Architecture

As mentioned before, our project consists of two stages: localization and classification. For the localization model, we transformed the image to different color spaces and used a heuristic method to localize the possible area of interest. There is no neural network involved in the localization stage. The reason for choosing a heuristic model is to increase the speed of the entire system to support real time identification.

In the classification stage, we divided the classification into two substages. The first substage is to categorize the traffic sign into four general classes of ‘regulatory’, ‘warning’, ‘temporary’ and ‘others’. Afterwards, the second substage will recognize which specific traffic sign it is, if it is not classified as ‘others’. To simplify the problem, we only consider the common types of signs as the following:

General Class	**Type**
Regulatory	Do not enter
	No left turn
	No right turn
	No straight
	No U turn
	Stop
Temporary	Construction
	Temporary detour
	Traffic control
Warning	Pavement narrow
	Pavement slippy
	Right lane ends
	Traffic lights ahead

Table 3: Selection of traffic sign types in each general class

We applied CNN to both stages, with detailed information shown below.

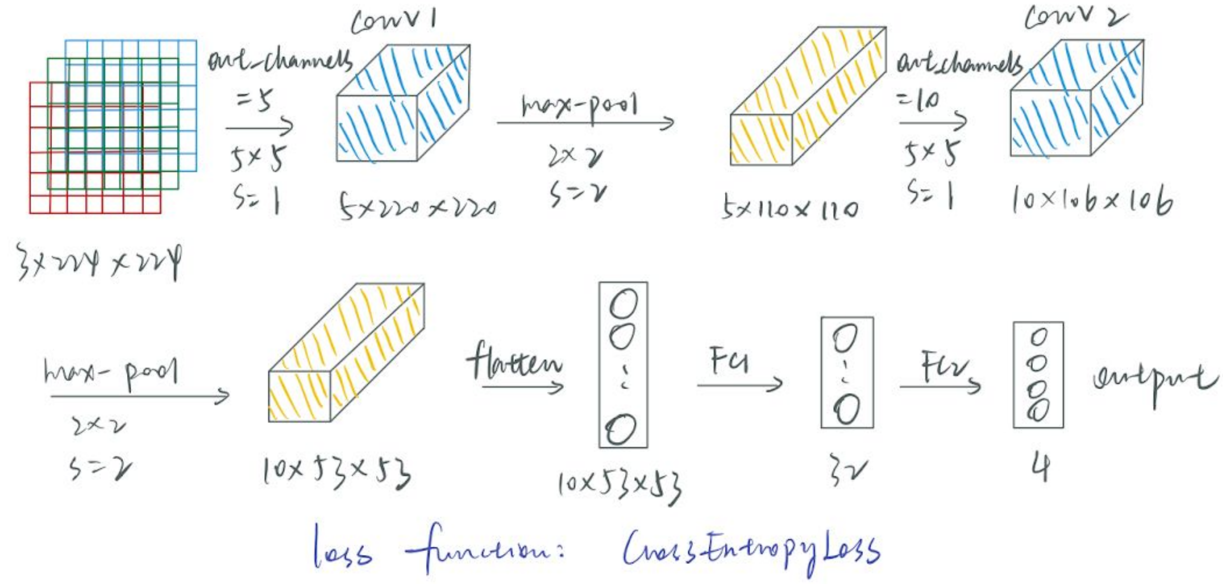


Figure 8: CNN structure for classification model stage 1

Since there are different numbers of classes in each general class, we are utilizing CNNs with different structures. The only difference between each model is the output dimension. Other layers have the same hyperparameter to guarantee the consistency of the models.

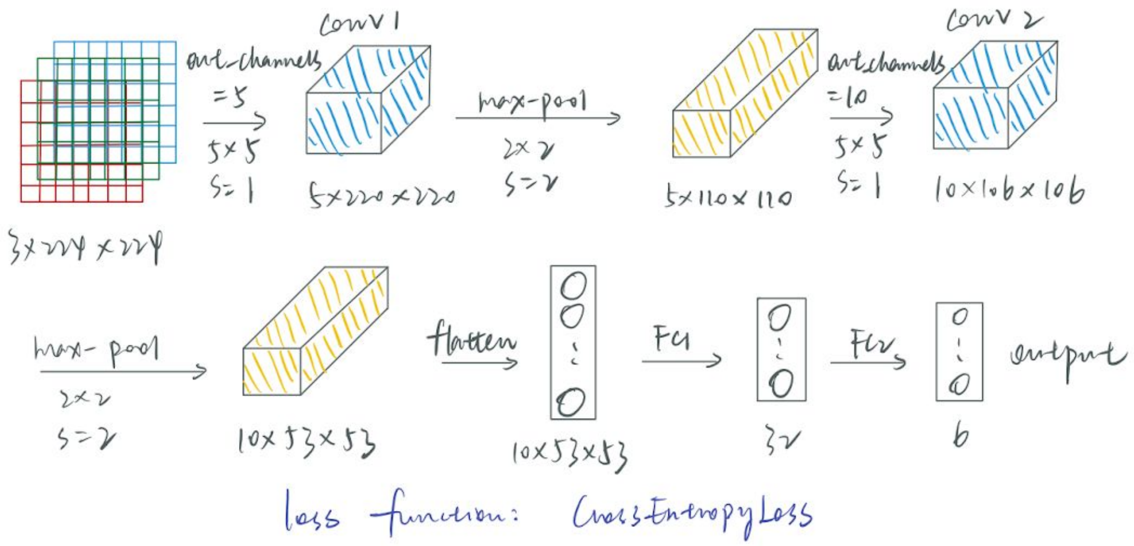


Figure 9: CNN structure for regulatory, classification model stage 2

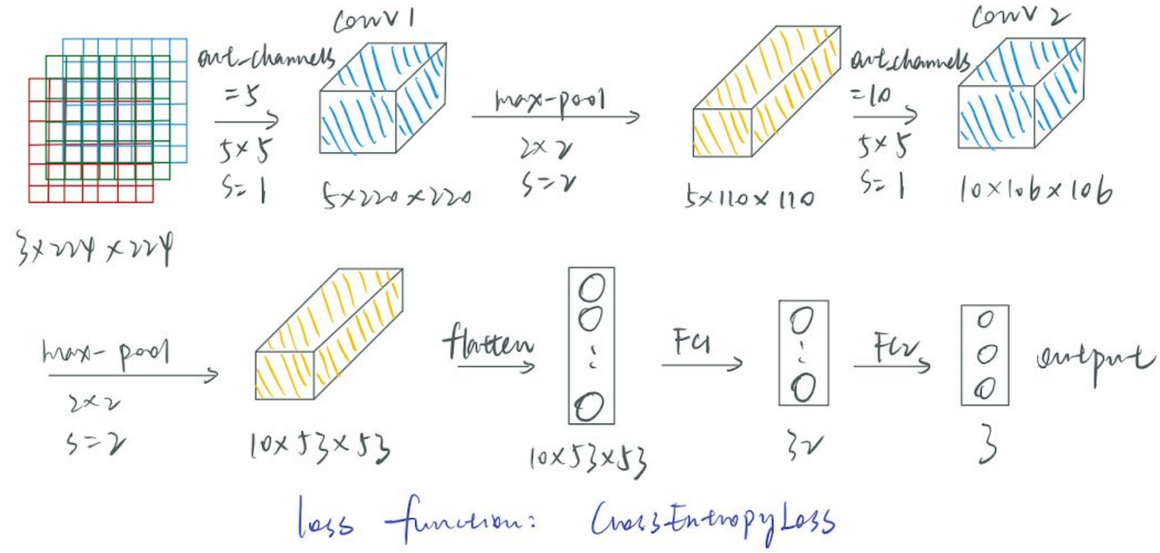


Figure 10: CNN structure for temporary, classification model stage 2

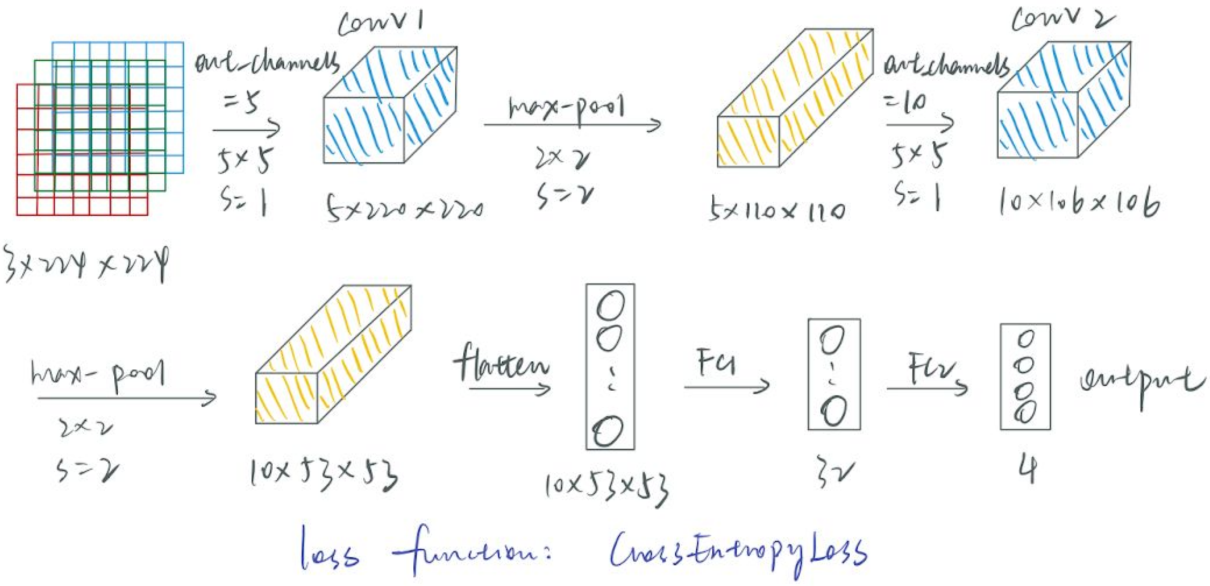


Figure 11: CNN structure for warning, classification model stage 2

Result and Evaluation

Localization Model:

- **Qualitative results:**

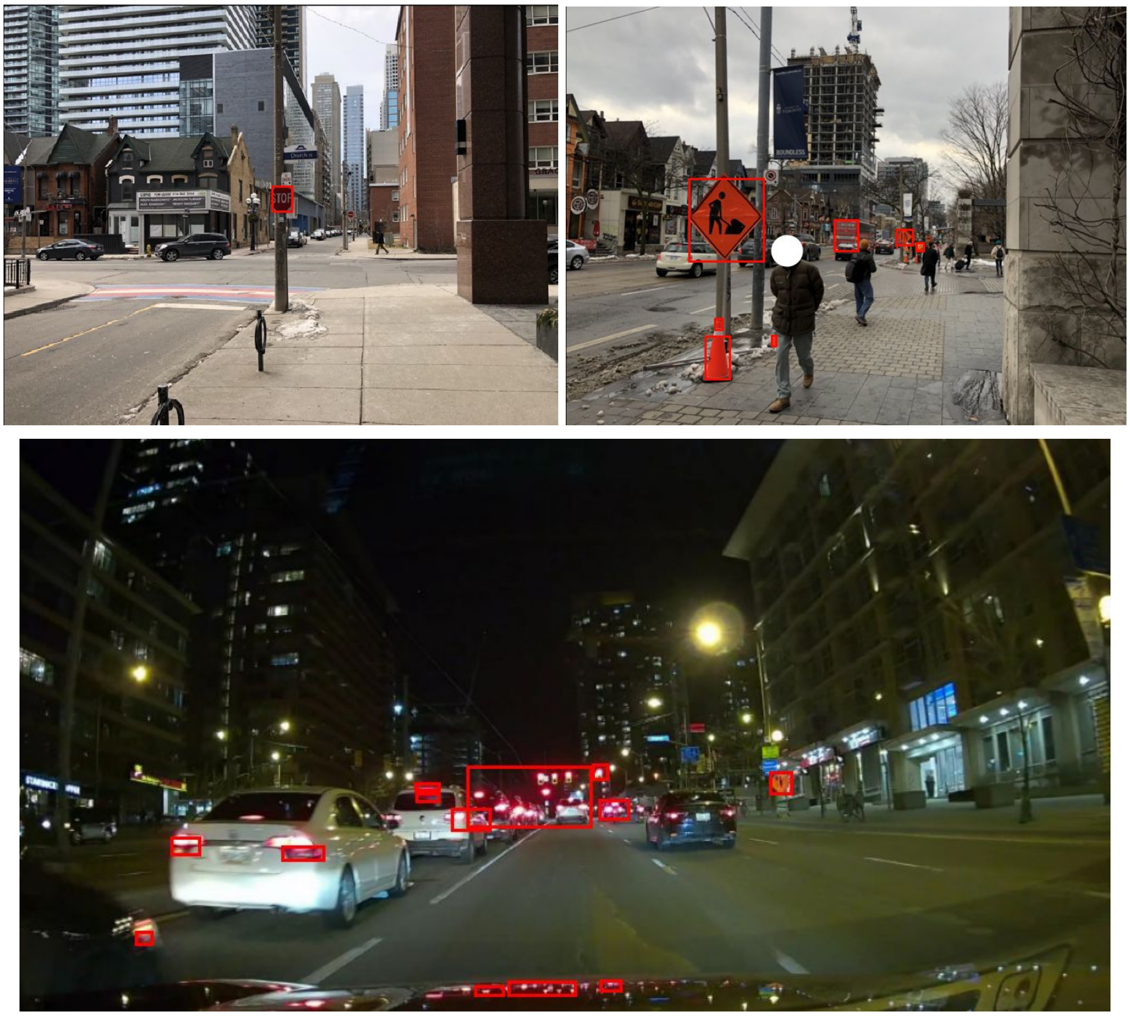


Figure 12(a), 12(b). Examples of localization result

The localization model has better behavior in day time than night and performs better when there is less background noise.

Classification Model:

Quantitative results for classification model stage I:

- **Learning curves**

Hyperparameter set: batch size = 16, learning rate = 0.005, number of epochs = 10

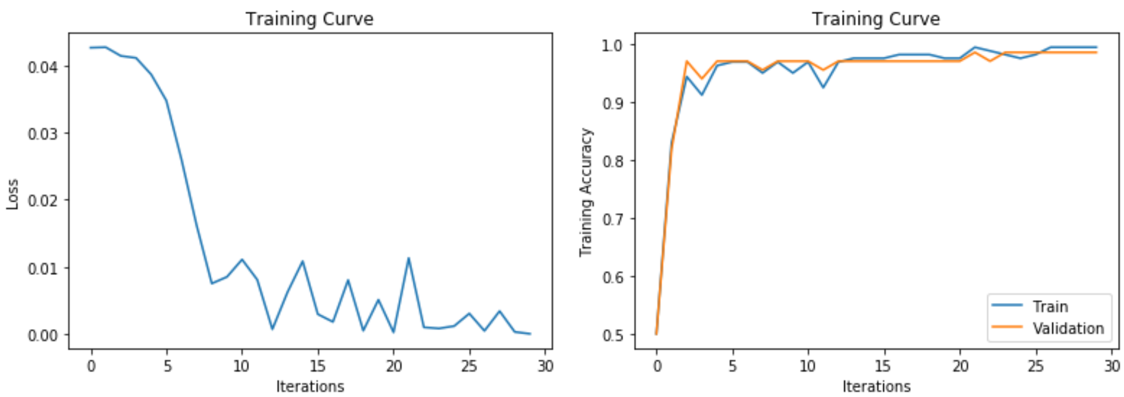


Figure 13. Training loss plot and accuracy plot for hyperparameter set II

Quantitative results for classification model stage II:

- **Regulatory learning curves**

Hyperparameter set: batch size = 128, learning rate = 0.002, number of epochs = 25

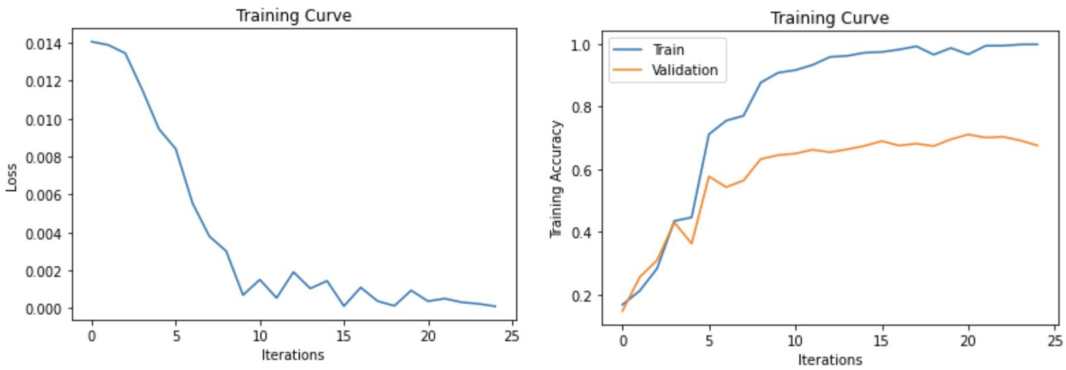


Figure 14. Training loss plot and accuracy plot for regulatory signs, stage 2

- **Temporary learning curves**

Hyperparameter set: batch size = 128, learning rate = 0.002, number of epochs = 25

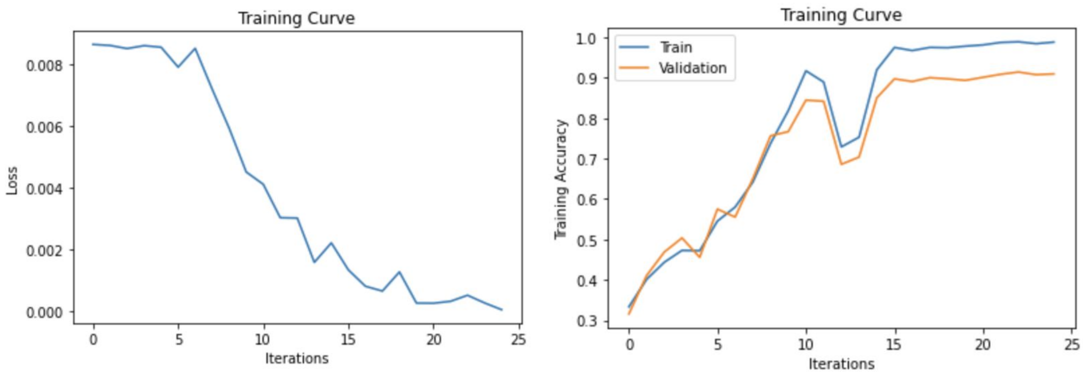


Figure 15. Training loss plot and accuracy plot for temporary signs, stage 2

- **Warning learning curves**

Hyperparameter set: batch size = 128, learning rate = 0.002, number of epochs = 25

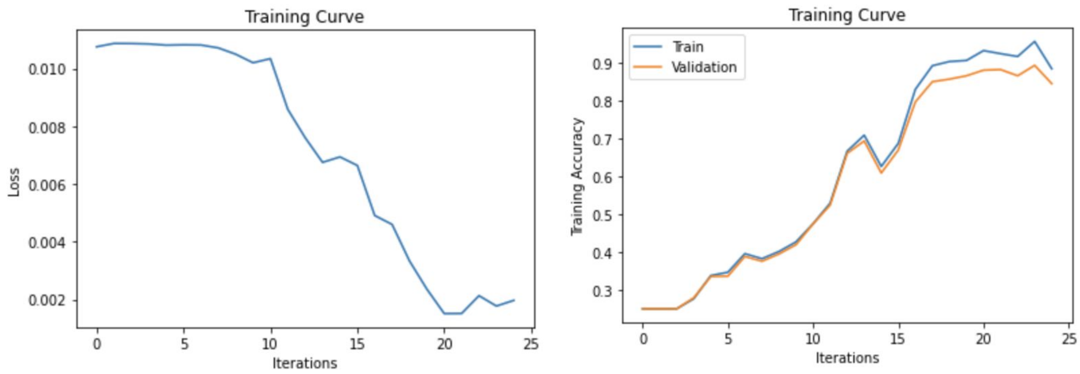


Figure 16. Training loss plot and accuracy plot for warning signs, stage 2

Qualitative results for Testing on New Samples

In the first substage of the classification model, we extracted fourteen signs from the Ontario Official Auto Drive Handbooks[4] and applied data augmentation, as described in Table 2, to expand our training and validation dataset.

In the second substage, we used fourteen signs and their augmentations as training dataset, and real life samples (the result of localization model) and their augmentations as the validation dataset. This arrangement is aimed to both expand the dataset and mimic the real life dataset.

After training models and tuning the hyperparameters, we tested both of the classification submodels. For the test dataset in both stages, we selected sample pictures from the localization result. Those samples are different from the validation dataset to ensure the quality of our model and test result, so the model has never seen them before. It should be noticed that the size of the test dataset is small because we applied the localization model on the screen shots of driving recorders and its storage will be cleared every few days so the video data is limited.

Stage I: Type classification

We tested on 17 new sample pictures, and the overall test accuracy is 94.1%. Here are three qualitative examples.

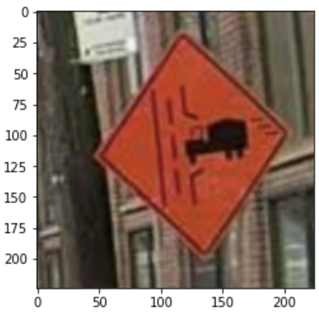
Sample test pictures	True label	Output distribution tensor	Classification result with probability
	Regulatory	tensor([-3.7759, 3.1587, -0.6869, -3.1139])	The probability of being a regulatory sign is 0.98, which is correct.
	Temporary	tensor([-0.6687, -1.5104, 1.7822, -1.2634])	The probability of being a temporary sign is 0.85, which is correct.
	Regulatory	tensor([-1.7787, 0.8621, 0.3281, -1.8231])	The probability of being a regulatory sign is 0.58, which is correct.

Table 4: Qualitative examples of general signs classification, stage 1

Stage II(a): regulatory

We tested on 18 new sample pictures, and the overall test accuracy is 83.3% Here are two qualitative examples.

Sample test pictures	True label	Output distribution tensor	Classification result with probability

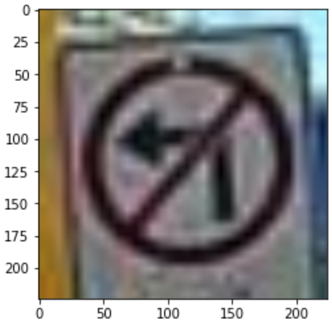
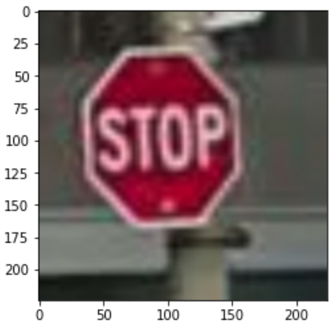
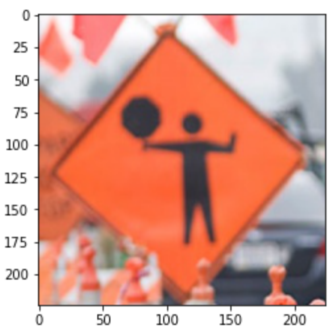
	No left turn	tensor([5.9753, 10.4152, -4.2517, -10.5053, -1.7218, 1.0131])	Correct prediction. The probability of being a no left turn sign is 0.99.
	Stop	tensor([1.0221, 3.2642, 0.2790, 4.1954, -3.8123, 5.1538])	Correct prediction. The probability of being a stop sign is 0.64.

Table 5: Qualitative examples of regulatory signs, stage 2

Stage II(b): temporary

We tested on 13 new sample pictures, and the overall test accuracy is 69.2%. Here are two qualitative examples.

Sample test pictures	True label	Output distribution tensor	Classification result with probability
	Traffic control	tensor([7.6757, -9.1175, 8.9846])	Correct prediction. The probability of being a traffic control sign is 0.79.

	Construction	tensor([4.4580, -2.9481, 4.0897])	Correct prediction. The probability of being a construction sign is 0.59.

Table 6: Qualitative examples of temporary signs, stage 2

Stage II(c): warning

We tested on 13 new sample pictures, and the overall test accuracy is 76.9%. Here are two qualitative examples.

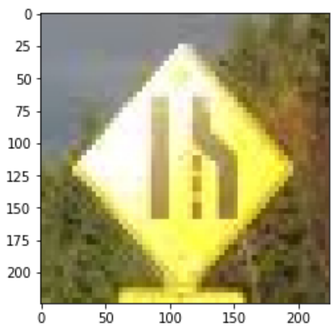
Sample test pictures	True label	Output distribution tensor	Classification result with probability
	Pavement narrow	tensor([7.0960, -0.7579, 5.9214, -5.4086])	Correct prediction. The probability of being a pavement narrow sign is 0.76.
	Right lane ends	tensor([4.3233, 6.5906, 7.8426, -10.7456])	Correct prediction. The probability of being a right lane ends sign is 0.76.

Table 7: Qualitative examples of warning signs, stage 2

Project Difficulty

Our project has multiple stages which consists of five different models (one heuristic model and four CNN models). One of the difficulties we faced when developing the heuristic model is the misclassification of the area of interest. Since we used colours and shapes to find the location of the traffic signs, when the input images contain a lot of misleading information such as red trucks, or round-shape restaurant plates, the model would recognize those as traffic signs too. We resolved this problem by adding a category of “other” to rule out non traffic signs at the first stage of the classification model.

Another important issue we faced during the project is the lack of available Canadian traffic signs datasets. The datasets we found online are images not from a driving recorder, or images that are hard to see even by human eyes, or contain signs not from North America. This increases the difficulty as we aimed to identify signs in real life from a driving recorder. Since the pictures we manually cropped and labeled from the driving recorder videos are not quantitatively sufficient to split into train, validation, holdout sets, we chose to use all of the data from the localization model for testing. Therefore, we used the virtual traffic signs with data argumentations to train our model.

Figure 17(a) - (c): sample data from online datasets

Discussion

Overall, we think our model is performing well. For the first substage of the classification model, our model can classify signs into general classes even if we did not train on those types, with an overall test accuracy of 94.1%. This indicates that our CNN model is able to learn sufficient patterns from small amounts of traffic signs to distinguish more general signs.

As for the second substage of the classification model, due to the lack of real world dataset, our model is trained only by virtual traffic signs (i.e. standard signs). However, we used real world signs to test the model. Compared to the virtual traffic signs, real world signs have very low resolution and confusing backgrounds as Figure 18 shows. Therefore, the average overall test accuracy is around 76.5%, which is lower than we expected but still a good result regarding the difficulties we met.

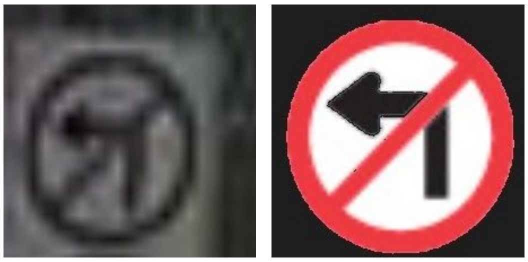


Figure 18: (left) sample test data, (right) sample train data

The result is interesting in a way that we didn’t foreseen earlier. Our model only takes virtual traffic signs as its input; however, as the results shown above, it could actually learn enough information to classify the traffic signs in the real world. What we have learned from the project is that the quality of the dataset is as important as the structure and parameters of the neural network.

In this project, the team spent more than two months and delegated a great amount of effort into it. Though the result is good for this project, it is not reliable enough for the real life implementation. Since in the real world, there should be zero tolerance allowed for any of the misclassification. One small error could cause a life and death difference, as we discussed in the ethical consideration section. As future engineers, we should always remind ourselves of the obligations and ethics associated with our profession instead of only focusing on the numeric number. In addition, after experiencing the difficulties of this project, the team respects all the scientists and engineers who contributed to the current autonomous driving systems, as what they are doing could make a huge difference to the world.

Reference

[1]"What Are The Leading Causes of Car Accidents in Ontario?", *Lalande Personal Injury Lawyers*, 2020. [Online]. Available: https://www.hamiltonlawyers.com/car-accidents/leading-causes/. [Accessed: 15- Mar- 2020].

[2] X. Han, C. Chen, K. Wang, Y. Yuan and Y. Song, "Study on Detection and Localization Algorithm of Traffic Signs from Natural Scenes", *Advances in Mechanical Engineering*, vol. 6, p. 318635, 2014. Available: 10.1155/2014/318635.

[3] E. Kršák and Š. Toth, "Traffic Sign Recognition and Localization for Databases of Traffic Signs", *Acta Electrotechnica et Informatica*, vol. 11, no. 4, 2011. Available: 10.2478/v10198-011-0039-2.

[4] Ontario.ca. 2018. *Traffic Signs And Lights*. [online] Available at: <https://www.ontario.ca/document/official-mto-drivers-handbook/signs>. [Accessed 20 March 2020].