C2TSR: Concurrent Canada-based Traffic Signposts Recognition System

Ms. Suby Singh

200434194; ssz389@uregina.ca

Department of Computer Science

University of Regina, SK, Canada

June 2020

1 Introduction

C2TSR as abbreviated for Concurrent Canada-based Traffic Signposts Recognition is a proposed deep learning model to classify and detect real-time traffic signs and signals on the streets of Canada. Road infrastructure has traffic signs and signals as an integral part of it which warns and instructs information to the vehicle drivers and pedestrians using symbols and words. These road facilities help road users to adjust their driving or walking behavior by providing them critical information and sometimes compelling recommendations, and ensure that they follow road regulation that is currently being enforced. Without these useful and meaningful signs i.e. if no feedback available on the speed limit, or is not directed about work in progress, turns, or school zone ahead, we might get exposed to more accidents, collisions, and traffic jams. Car manufacturing companies, such as BMW, are proactively interested in the Advanced Driver Assistance System (ADAS) as a result of advancement in automotive intelligent techniques. Systems like ADAS have a central controlling activity which not only includes lane detection but also includes traffic signs and signals recognition [8]. So, these autonomous vehicles must also abide by the legislation of road traffic signs and henceforth these car systems must understand and recognize these rules. The proposed model of recognition technique can be incorporated with the Advanced Driver Assistance System in the autonomous car driving in real-time scenarios. Pedestrians must also adhere to traffic rules and regulations to avoid traffic accidents. I feel that this model may also be advantageous for visually impaired people if the proposed model is incorporated with advanced smart eyeglasses having camera lenses to capture the real-time videos while using road facilities. Below are some motivational points considered while picking up this topic:

1. A prelude of fully automated driving or providing semi-automotive driving systems to assist maintaining speed limit and following other regulations
2. Assistant for drivers to be more attentive (if used with speaker mechanism)
3. Reducing the proximity of accidents caused due to driver's distraction
4. Assisting visually impaired individual when incorporated with advanced smart eyeglasses and speaker mechanism
5. Implementing Car dashboard for the indication of the speed limit and other indicators for the comfort of the driver
6. Help in regulating the traffic by informing condition of the road ahead
7. Assist in improving the comfort and safety of the driver on the road
8. All the above points with the aim to achieve zero accident
9. Less number of research/analysis available for Canadian traffic signs due to lack of public dataset

2 Problem Statement

Around 94% of van crashes are caused due to human error as stated by National Highway Traffic Safety Administration(NHTSA) as well as [12][13]. Among these, recognition and decision errors are the most common and frequent reason for accidents. Hence, implementing a technique to avoid accidents is strongly recommended. Nowadays, Vehicles with Advanced Driver Assistance System (ADAS) in few areas such as automatic braking, lane detection and departure warning system, automatic parking, driver drowsiness detection have been developed [8]. Car manufacturing company Tesla has worked on stop sign and traffic lights but it does not consider all types of signs, and this feature has it's own limits as Tesla made cars get slow down after detecting any type of traffic lights including green light [16]. For ADAS or legally blind people using smart eyeglasses, we required to have a model to be actively working on detecting traffic signs and signals to avoid traffic accidents. Traditionally, many methods including deep learning[14] and image processing techniques[15] were employed to detect and recognize traffic signs but they were not highly effective. These existing models cannot be used in Canada as these models use different traffic signs. There isn't much work done for Canadian traffic signs due to the unavailability of required dataset. Analysis has been performed on Germany and Swedish, or othe European countries but their traffic notation is different from Canada as shown in Fig. 1. Hence, Considering these points I propose an

Figure 2.1: Speed limit sign in Europe (Left) and Canada (Right)

application empowering a Canada-based traffic dataset. The proposed model to be built by taking advantage of deep learning method, possibly YOLO-You Only Look Once algorithm[2] to detect and classify the traffic signs in real-time scenario. Goal of this project is to contribute efforts in the field of traffic sign-posts in order to make roads safer place for pedestrians and vans. The proposed model can be incorporated with speakers to keep the drivers informed and avoid any distractions which probably be caused by sleepiness or drowsiness.

3 Solution

Based on the report given by [12][13], ADAS technology with its great feature can potentially help in avoiding 36% of the car crashes and 35% of collision injuries. The proposed solution, which can be a great contribution to ADAS technology is to

1. Implement a technique to inform drivers about incoming traffic signs and warn them about the upcoming situations
2. Create a large dataset for Canadian traffic signs and road signals by extracting frames from the videos taken by using Python and OpenCV. images or the raw data must cover traffic lights and signs in different shades of light and climate.
3. Build a model using deep learning method (YOLO- You Only Look Once) to classify and detect the traffic signs and signals as trained using frames/images taken out from the videos and make this model learn new features as it evolves over time or as it fetches more traffic signs as well as simulate this model in the real-time scenario

As shown in the Figure 3.1, we can observe that I am planning to re-use the similar signs from the existing dataset such as GTSRB - German Traffic Sign Recognition Benchmark [7] due to it's open source availability and reusability.

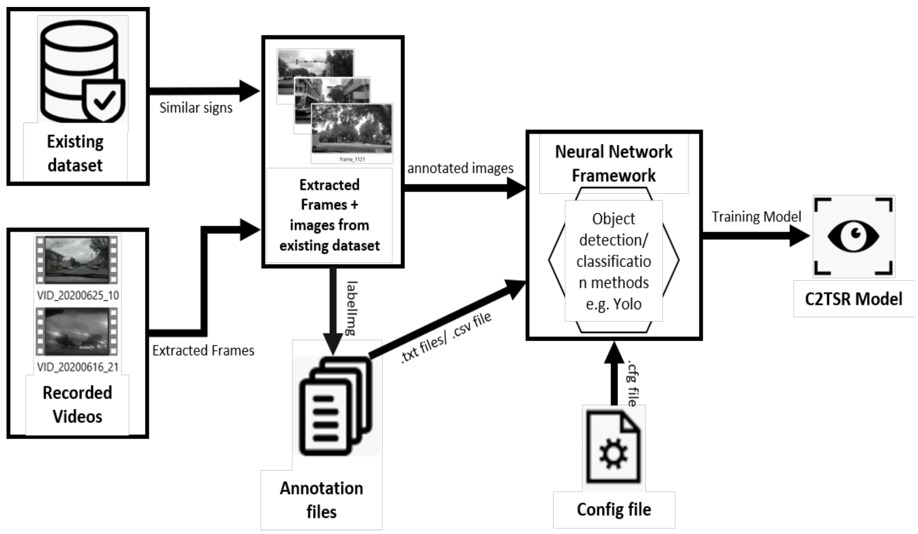

Figure 3.1: C2TSR Model architecture

At the same, I will be collecting or recording videos on the streets of Regina while covering different signposts used in Canada. The selected images from the existing dataset and extracted frames from the recorded videos will then be used for creating annotation files. Please refer Creating dataset section for more details on annotation files. The more images or frames and their corresponding annotation files we use, the more accurate our model would become. So, creating a huge and useful dataset is a primary focus in this project. Later, annotated files and frames/images will be used in training YOLO algorithm in Darknet framework. YOLO is a state-of-the-art model for real-time object detection system and faster than other algorithms such as SSD and Faster R-CNN. Darknet is an open source neural network framework written C to be used on CUDA-enabled GPUs for training objection detection model [2]. There are other frameworks too, but darknet with Yolo provides better performance as it is written in C [2]. Once our model is ready, we can use the weight file (trained output file) generated while training and configuration file to detect traffic signposts in images and videos.

4 Data Preparation

- **Source:** Extracted frames from recorded Videos
- For this, I will be recording videos during different timings to capture the different levels of visibility
- **Format:** JPEG images with different brightness and dpi level

- Annotation files corresponds to each extracted frames from the recorded video
- **Access:** store the created dataset on Google drive and access it using Google Colab
- **Statistics:** 50+ different classes of traffic signs and signals (Each class with at least 1000 images)
- **Risk factors while preparing dataset :**
  - Different lightning and climate conditions
  - Poor visibility
  - Camera alignment and focal point
  - Motion effects while capturing video
  - Different traffic colors for signs and boards
  - Damaged traffic sign boards

5 Creating Dataset

Creation of the dataset includes two main sources of images:

1. Extracted frames from the recorded videos
2. Similar signposts from other publicly available dataset
3. Other could be web scrapping

A frame per second from the video will be extracted using openCV and python. later, the extracted frames will be stored on local machine initially. Once all the required frames have been extracted and similar signs have been collected from the existing dataset, we need to create annotation of objects in the image for each frame or image file using LabelImg tool. LabelImg is an annotation tool written in qt and python for Graphical User Interface. It generates annotation in XML/PASCAL VOC or yolo format [4]. For this project, I am going to create bounding boxes for the objects in yolo format. It is a simple format which contains class id, coordinates (x and y), width and height of the rectangular portions which are manually drawn to label different objects in an image. It creates a simple text file having annotation details at each line per object in the corresponding image [4].

- **Step 1:** Record Videos and collect images via web scrapping
- **Step 2:** Extract frames (1 frame per second) and remove unwanted frames
- **Step 3:** Re-use signs and posts images from existing dataset such as GTSRB - German Traffic Sign Recognition Benchmark which is available on Kaggle, an online community of ML practitioners and data scientists [7]

- **Step 4:** Create class/data file. Class Id starts from 0. Following is the class/data file created for C2TSR model:

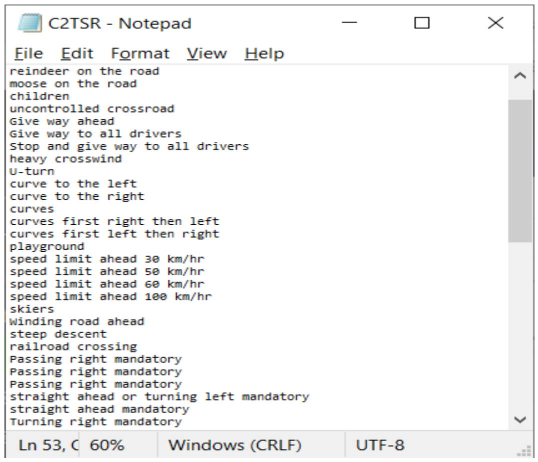

Figure 5.1: Class names

- **Step 5:** Create annotation files corresponding to each frame using labeling tool such as labelimg
  To run this tool, used following command:

```
(base) C:\Users\Sushitha Rajeev\Suby\labelImg-master>python labelImg.py ../CTS2R/Data/Raw/Frames ../CTS2R/miscFiles/C2TSR.names
```

Following is an overlook of LabelImg tool while annotating a frame:

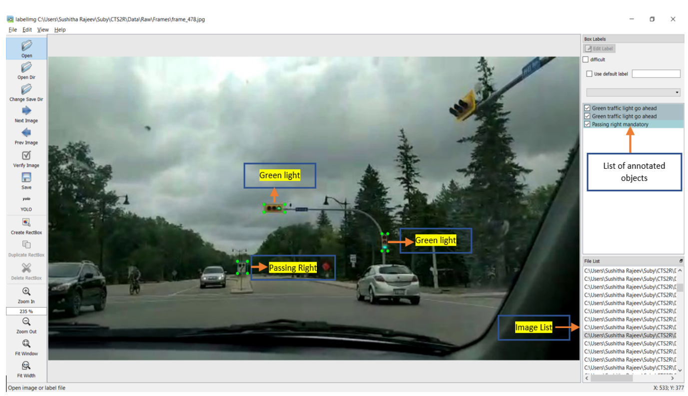

Figure 5.2: LabelImag Example

Following is the annotation file generated by LabelImg tool for above frame:

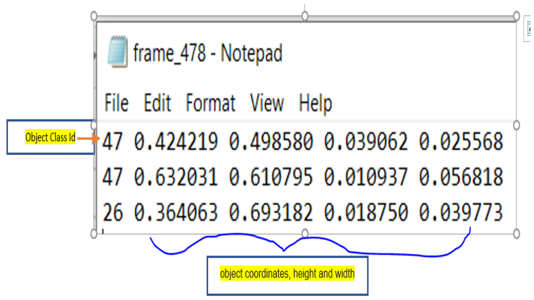

Figure 5.3: LabelImag annotation file

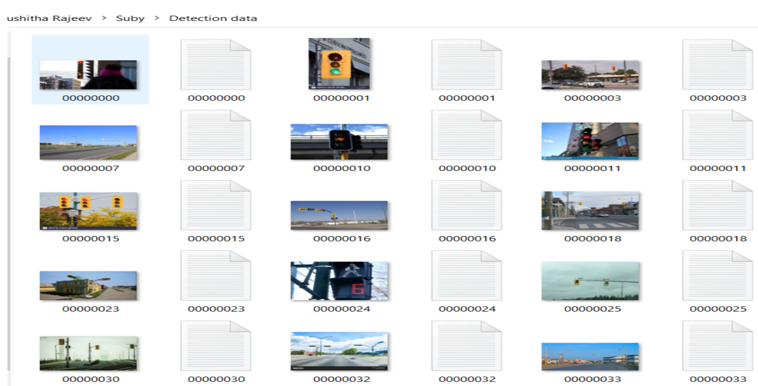

Figure 5.4: Folder having frames and it's corresponding annotation folder

- **Step 6:** Customize configuration file. Please refer Figure 5.5 on next page. Here, I have mentioned about only one parameter i.e. classes which would be the number of object classes. There are many other parameters which need to be reset as the number of classes increases. I referred [1][5][6] while creating classes.
- **Step 7:** Store dataset on Google Drive and access using Google Colab notebook. Create a zip file of the image for speedy transfer to the Google drive.
- As of now I have collected frames altogether of 1 GB without annotation files. Later, this zip file will be loaded into data folder of darknet framework and used while training C2TSR model.

```
filters=30
activation=linear


[yolo]
mask = 3,4,5
anchors = 10,13,  16,30,  33,23,  30,61,  62,45,  59,119,  116,90,  156,198,  373,326
classes=5
num=9
jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=1



[route]
layers = -4
```

Figure 5.5: YOLO configuration file snippet

6 Model Testing and Expected Outcome

- The generated will first be tested using a set of images to detect the traffic signposts. Criteria for measuring the success will be accuracy and confidence level for the detected objects in an image. Once successful, the model will then be tested using recorded videos or videos taken from Camera.

- A Dashboard or User Interface window to load the videos from a folder or camera and display the information on it as it detects and classifies different traffic signs.

- Following are the testing models:
  1. on images - refer Figure 6.1
  2. on videos (either recorded or camera) - refer Figure 6.2

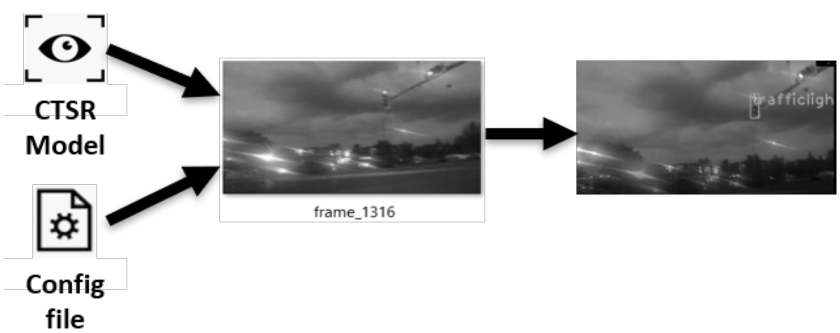

Figure 6.1: C2TSR model testing using images

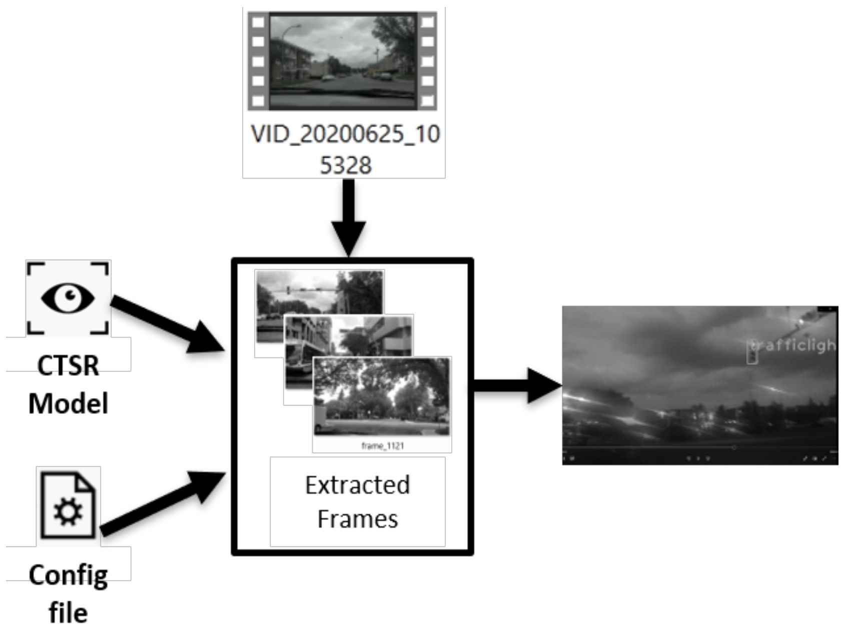

Figure 6.2: C2TSR model testing using videos

7 Tools

- **Annotation :** LabelImg
- **Modules :** OpenCV-python, numpy, panda, os, glob, matplotlib, random etc.
- **Framework :** tensorflow-gpu, darknet
- **Platform :** Jupyter lab; and Overleaf for writing documents
- **Storage :** Google Drive to store dataset and model

8 Prototype of the proposed model

Figure 8.1 depicts the detailed implementation of the model. I have created a sample test data and model following this approach and stored on Goggle drive at
`https://drive.google.com/drive/folders/1vEc8Fb3din6YdnqTbAnAH1s8V2HGw5-k?usp=sharing`

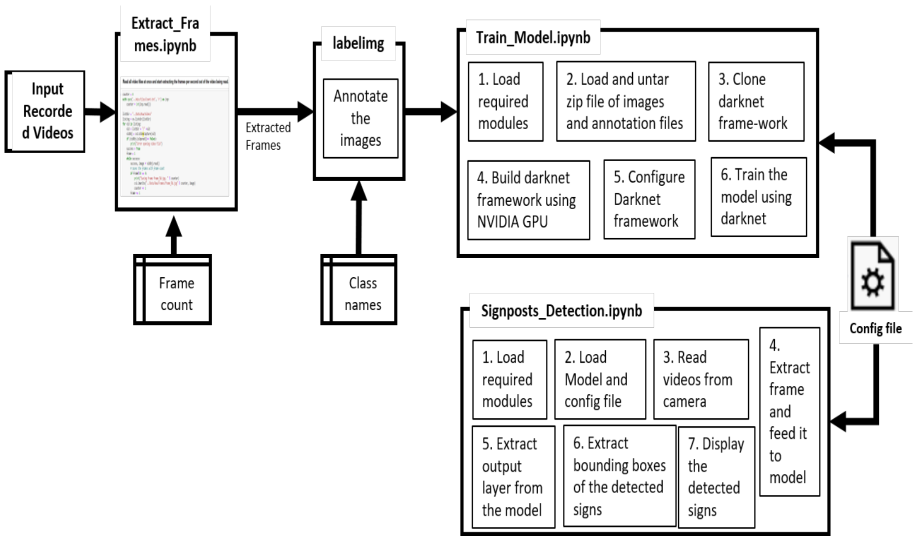

Figure 8.1: Prototype of the proposed model

List of scripts and miscellaneous/config files:

- Extract_frames.ipynb
- Train_Model.ipynb
- Signposts_Detection.ipynb
- Yolo_config.cfg, ¡annotation_file¿.txt, classes.txt, count.txt

Read all video files at once and start extracting the frames per second out of the video being read.

```
counter = 0
with open('../miscFiles/Count.txt', 'r') as inp:
    counter = int(inp.read())

listDir = "../Data/Raw/Videos"
listing = os.listdir(listDir)
for vid in listing:
    vid = listDir + "/" +vid
    vidObj = cv2.VideoCapture(vid)
    if (vidObj.isOpened()== False):
        print("Error opening video file")
    success = True
    frame = 1
    while success:
        success, image = vidObj.read()
        # Saves the frames with frame-count
        if frame%30 == 0:
            print("Saving frame frame_%d.jpg. " % counter)
            cv2.imwrite("../Data/Raw/Frames/frame_%d.jpg" % counter, image)
            counter += 1
        frame += 1
```

Figure 8.2: Code snippet from Extract_Frames.ipynb script

9 Potential Application of the Model

- The proposed model can be integrated with ADAS technology and contribute as a great feature in modern cars. There are many Autonomous vehicle companies such as Tesla, GM Motors, Yandex and Wayo to name a few are involved in developing self driving cars. By CRM staff, more than half of the drivers in Ontario have no or little knowledge about ADAS as per a survey conducted by Ipsos (a market research and consulting firm) having 1,202 Canadians as participants, in the year 2019 [9]. At the same time another survey conducted by McKinsey & company(a consulting company) with 4,500 participants revealed that most of these were not aware of ADAS features and among those who knew about ADAS, 87% to 89% drivers have bought ADAS equipped cars [10].
- So, we can estimate that as car consumers awareness about ADAS features increases, there will be a potential increase in the number of consumers using these features.
- The proposed model may also be advantageous for visually impaired people if the it is incorporated with advanced smart eyeglasses having camera lenses to capture the real-time videos and keep them giving a sense of warning about upcoming situations while accessing road facilities

10 Business value of the model

- This model will underline a key advancement in the field of autonomous vehicles and their development in ADAS system in Canada and around the world.
- As per [13], it is estimated that market value for Autonomous vehicles will reach to $\$77 Billion$ by 2035.
- There have been some underlying growing demands in the number of workers for the development like these. In 2016, it was seen 213,300 workers in Canada's connected and autonomous vehicle(CAV) where as it is expected to have 3% growth in the numbers of workers i.e. near about 248,000 workers in 2021 [13].

11 Timeline

About 2 months to:

- record videos
- collect publicly available and applicable annotated images
- extract frames

- create annotations for the extracted frames
- train model
- test the created model with images
- test the created model with recorded videos
- test the created model with camera

References

[1] Ministry of Transportation, https://www.ontario.ca/document/official-mto-drivers-handbook/signs

[2] Redmon, Joseph and et al., "YOLOv3: An Incremental Improvement", arXiv, 2018

[3] S. Xu, Z. Zhao, X. Wu, "Object Detection with Deep Learning: A Review", IEEE Transactions On Neural Networks And Learning Systems, 2019

[4] Tzutalin. LabelImg, Git code, https://github.com/tzutalin/, 2015

[5] www.AdcIdl.com, International Driver's License Application online, Canadian Road Traffic Signs"

[6] sgi, "2019-20 Saskatchewan Driver's Handbook" https://www.sgi.sk.ca/handbook/-/knowledge_base/drivers/traffic-signs, 2019

[7] GTSRB, https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign

[8] Wikipedia, "Advanced driver-assistance systems", "https://en.wikipedia.org/wiki/Advanced_driver-assistance_systems"

[9] CRM staff, "https://www.collisionrepairmag.com/adas/", 2019

[10] McKinsey and company, "https://www.mckinsey.com/industries/semiconductors/our-insights/advanced-driver-assistance-systems-challenges-and-opportunities-ahead#"

[11] Cutean, A. Autonomous Vehicles and the future of work in Canada, Information and Communications Technology Council(ICTC), Ottawa, Canada, 2017

[12] Canadian Auto Dealer, "https://canadianautodealer.ca/2020/01/canada-is-behind-in-adas-awareness-training/", 2020

[13] Belron Canada Inc., The hidden link between windshields and road safety, https://www.speedyglass.ca/DATA/CMSDOCUMENT/46.pdf, 2019

[14] R. Kulkarni, S. Dhavalikar and S. Bangar, "Traffic Light Detection and Recognition for Self Driving Cars Using Deep Learning," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-4, doi: 10.1109/ICCUBEA.2018.8697819.

[15] C. yu, Y. Bai, A Traffic Light Detection Method, Advanced Technology in Teaching pp 745-751

[16] Kirsten Korosec, https://techcrunch.com/2020/04/27/tesla-vehicles-recognize-and-respond-to-traffic-lights-stop-signs-with-latest-software-update/, 2020