Vision System for Enhanced Traffic Sign Detection and Driver Compliance

Huandong Chang
Harvard University
huandongchang@fas.harvard.edu

Raphael Krief
Massachusetts Institute of Technology
raphk@mit.edu

Abstract

This study introduces a computer vision system that enhances road safety by monitoring driving behaviors through traffic sign detection. Employing the real-time object detection capabilities of the YOLO model, augmented with tracking methods for precise temporal analysis, this system provides direct feedback to drivers using a simple dashcam setup. By identifying traffic violations, the system aims to improve driver awareness and adherence to traffic laws.

1. Introduction

1.1. Background and Motivation

The increase in road traffic necessitates innovative, accessible technological solutions for enhancing driver safety. Although autonomous driving technologies are evolving, they often remain out of reach due to high costs and complex integration requirements. In the US alone, over 40,000 people died in motor vehicle traffic crashes in 2023, highlighting the critical need for effective safety interventions. Many drivers are not aware of their bad driving behaviors, which can lead to severe accidents. Current literature and systems on traffic violation detection predominantly rely on CCTV cameras, which are static and cannot provide direct feedback to drivers in real-time.

1.2. Project Objectives

This project aims to detect traffic rule violations using a dashcam setup and a telematic sensor, enhancing driver safety through a vision-based system. By employing the YOLO object detection system [1], enhanced with the SORT algorithm [2] that uses Kalman Filters [3], our system not only identifies but also tracks traffic elements dynamically across video frames. Unlike traditional systems that focus on penalizing drivers post-violation, our approach aims to assist drivers by offering real-time feedback. This helps new drivers form good driving habits and collaborates with insurance companies to potentially offer discounts to safe drivers, thereby incentivizing prudent driving behaviors. The integration of this technology in everyday driving could significantly reduce the number of traffic-related fatalities and injuries by alerting drivers to potential infractions before they occur.

2. Related Work

The domain of traffic violation detection has seen the development of various systems designed to identify and penalize breaches of road regulations. Notably, systems such as the one described by Xu et al. (2020) [4] leverage CCTV cameras strategically positioned at intersections to monitor and record traffic violations like running red lights and illegal turns. Similarly, Smith and Lee (2018) [5] implemented a network of interconnected CCTV cameras that utilize AI algorithms to detect speeding and unauthorized lane changes across urban areas. Another significant contribution by Johnson et al. (2019) [6] uses high-definition cameras along highways to capture and analyze traffic patterns, specifically targeting aggressive driving behaviors.

However, these systems primarily utilize fixed CCTV cameras and are geared more towards penalizing offenders after violations have occurred. In contrast, our system is based on a dashcam integrated within the vehicle, which provides a mobile and driver-centric approach to violation detection. The primary aim of our system is not just to penalize but to assist drivers by offering real-time feedback and alerts. This proactive approach helps in preventing violations before they occur, enhancing road safety directly from the driver's perspective. By focusing on immediate assistance rather than post-event penalties, our system aligns more closely with preventive measures and driver education.

3. Dataset Description

Our research utilizes the Mapillary Traffic Sign Dataset (MTSD), which is among the most comprehensive datasets available for traffic sign detection in academic research. This dataset includes street-level images from diverse geographic locations, featuring a broad spectrum of scenes and conditions. It comprises over 313 unique traffic sign classes

and more than 250,000 annotated signs [7].

The MTSD is designed to reflect the variability encountered in real-world settings, making it an excellent resource for developing robust detection algorithms. The dataset includes 52,453 images with fully annotated traffic sign bounding boxes, providing a rich base for training our models. The high-quality annotations encompass a variety of lighting and seasonal conditions, which enhances the utility of the dataset for training sophisticated traffic sign detection systems.

4. Proposed Method

Our proposed pipeline unfolds across several key stages. Initially, data preprocessing is conducted to prepare the input images, enhancing their suitability for subsequent analysis. Following this, we employ YOLOv9 for object detection to identify relevant traffic elements in the preprocessed images. As videos inherently contain interdependent frames, our next step integrates object tracking using SORT to maintain continuity across sequences. To refine our outputs, a denoising step is implemented to reduce noise and improve the clarity of our tracking results. The final stage involves detecting traffic rule violations, with a specific focus on identifying stop sign infractions.

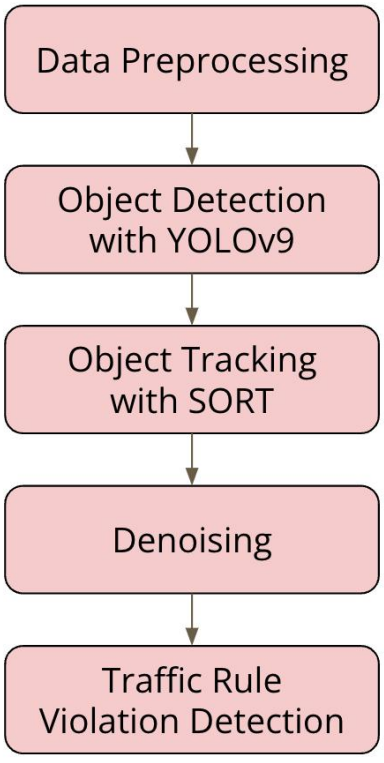

Figure 1. Algorithm Pipeline

4.1. Data Preprocessing

In the preprocessing phase of our project, the primary objective was to convert and simplify the dataset to better tailor it to our specific requirements for traffic sign detection with an emphasis on risk assessment. The original annotations provided in JSON format were converted into the YOLO format to facilitate the use of the YOLO object detection architecture.

Label Reduction and Categorization

Initially, the dataset comprised 313 traffic sign labels categorized into four types: regulatory, warning, information, and complementary. Given the overarching goal of assessing accident risk, we focused on 'regulatory' and 'warning' labels, which have direct implications for driver behavior and safety. We consolidated similar labels to reduce complexity—for instance, various speed limit signs were grouped under a single label. This approach not only simplified the training process but also aimed to enhance the detection model's accuracy by concentrating on the most impactful sign types.

Filtering by Presence and Size

We also eliminated labels that were underrepresented in the dataset and removed signs with bounding box areas smaller than 1000 pixels, as these were inaccurately recognized and could potentially skew the model's performance. After these adjustments, the dataset was narrowed down to 16,598 images spanning 33 distinct labels. This refined set of labels ensures that our model focuses on the most relevant traffic signs for assessing and mitigating accident risks, facilitating more effective training. This table presents a summary of the top 10 traffic sign labels used in our study, showing the count of each label retained for model training. The complete table can be found in Appendix A.

Label ID	Label Description	Count
0	R-SpeedLimit	4767
1	R-Yield	1843
2	R-KeepRight	1553
3	W-PedestrianCrossing	1492
4	R-Stop	1158
5	R-NoEntry	1141
6	R-NoParking	1010
7	R-NoHeavyVehicles	999
8	W-RailroadCrossing	830
9	R-NoOvertaking	818
10	W-MergeRight	726

R stands for Regulatory and W stands for Warning.

4.2. Object Detection with YOLOv9

YOLO combines what was once a multi-step process, using a single neural network to perform both classification and prediction of bounding boxes for detected objects [1]. Therefore, YOLO is very fast and is great for video processing. We finetune YOLO from pre-trained weights gelan-c with 300 epochs, 16 batch size, and 640*640 image size. The training is done on an NVIDIA GeForce RTX 3090 server, and it took about 40 hours to finish 300 epochs.

We reached 0.719 Precision and 0.599 Recall on the validation set when we set the confidence threshold to 0.5, as shown in 2. You can find more metrics and their details in Appendix B.

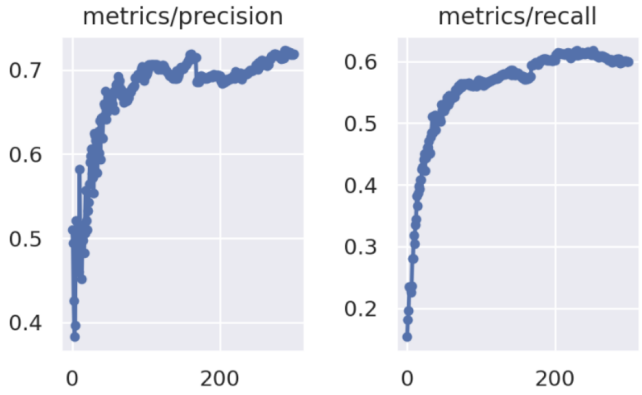

Figure 2. Recall and Precision during Training

Based on the Confusion Matrix in Appendix C (too large to fit here), we can tell our algorithm has a lot of false negatives where they predict traffic signs as backgrounds, but this makes sense as we set the confidence threshold to be high (0.5). Also, we can tell the algorithm sometimes mixes up between left and right (such as between Curve Left and Curve Right). Other than this, our algorithm can decently detect traffic signs, and the following steps in our algorithm pipeline can address some false positives and false negatives by considering dependencies between frames. Here is a demo video of driving in Cambridge with YOLO only, and you can find a sample frame in figure 3.

4.3. Object Tracking with SORT

Simple Online and Realtime Tracking (SORT) is an object tracking algorithm that is widely used in video processing due to its efficiency and effectiveness in tracking objects across frames. SORT operates in two primary steps with Kalman Filter and Hungarian Algorithm [2].

The first step in the SORT algorithm involves applying the Kalman filter to predict the future state of each detected object. The Kalman filter is a well-known algorithm in signal processing and control systems for inferring the internal

Figure 3. Inference on Cambridge Driving Video using YOLO. It can detect most of the traffic signs that exist in the training set, but sometimes it makes some mistakes as shown in the pink bounding box. We do not have this specific "No Left Turn" Sign in the training set but YOLO predicts it as "No U Turn".

state of linear dynamic systems from a series of noisy measurements. In the context of SORT, the Kalman filter estimates the position and velocity of each object based on its current and previous states. This prediction step is crucial as it allows the algorithm to account for motion between frames and anticipate where each object is likely to appear in the next frame.

The second step of SORT utilizes the Hungarian Algorithm, a combinatorial optimization algorithm also known as the Munkres or Kuhn-Munkres algorithm, for data association. In SORT, the Hungarian Algorithm is used to associate predicted object states (from the Kalman filter) with new detections in the subsequent frame. This step ensures that each object maintains a consistent identifier across frames, even when the object's appearance may change due to movement, occlusion, or changes in viewing angle [8].

These two steps enable SORT to robustly track multiple objects in real time. By predicting object positions and optimally assigning new detections to existing tracks, SORT minimizes the total prediction error across all tracked objects. This capability makes SORT particularly effective in environments where objects move relatively predictably, such as in our case.

4.4. Denoising

After applying the SORT algorithm, we still get some noises in the video. For example, some IDs are only associated with one or several frames, which are very likely to be false predictions by YOLO. Therefore, we only keep traffic signs that appear more than 20 frames (our video is 60 frames/second, so 0.33 seconds).

Also, since the SORT algorithm does not take class prediction as input but only bounding box coordinates, there are some cases where one ID can map to more than one class prediction. However, only one class prediction is (more likely to be) correct. Therefore, if an ID has more

than one class prediction, we take the mode. Here is a demo video of driving in Cambridge with YOLO & SORT, and you can find a sample frame in figure 4.

Figure 4. Inference on Cambridge Driving Video using YOLO & SORT. Compared to Figure 3, the correct prediction is kept with an ID number, and the wrong prediction on the "No Turn Left" sign disappears. This is because YOLO only makes mistakes when the driver is far away from the "No Turn Left" sign, so the wrong prediction only shows up in the first several frames but get removed during denoising.

4.5. Traffic Rule Violation Detection

For this part, we mainly focused on a stop sign detection algorithm. Leveraging the YOLO object detection framework, the algorithm identifies potential stop signs within video frames. Each detected object is assigned a unique tracking ID and is categorized based on its label. Once a stop sign is detected, the algorithm tracks its location across consecutive frames using the Intersection over Union (IoU) method to assess the stability of the stop sign's position.

The stability of detected stop signs is evaluated by calculating the IoU for the bounding box across successive frames. An IoU threshold of 99% is set to ensure that minor detection variances between frames do not affect the continuity of tracking. This high threshold is crucial to minimize the impact of small movements or camera shakes that could otherwise lead to inaccurate detections.

To ascertain if a vehicle has fully stopped at the sign, the bounding box of the stop sign must remain nearly identical (99% IoU) for at least 15 consecutive frames. This criterion represents a 0.25 second stopping period, considering the video is recorded at 60 frames per second, and helps distinguish between a complete stop and a mere slowdown.

As the vehicle stops and subsequently starts moving, the algorithm annotates the video by changing the color of bounding boxes: red indicates that the vehicle is moving, and green appears when it has stopped for the required duration.

This straightforward method provides an effective mean for detecting stop sign violations and offers a foundation for future enhancements, such as integrating real-time alerts for drivers approaching a stop sign or conducting further traffic pattern analysis.

Results

To evaluate the effectiveness of our traffic rule violation detection algorithm, we analyzed two specific scenarios involving stop signs.

The first scenario demonstrates a driver who correctly stops at a stop sign. As shown in Figure 5, the dashcam captures the stop sign with a blue bounding box and the label *"Stopped for 0.27s"*. This indicates that our algorithm successfully detected the stop sign and verified that the vehicle remained stationary for the required duration.

Figure 5. Detection of a stop sign with the vehicle stopped for 0.27 seconds.

The second scenario involves a driver who fails to stop at the stop sign. In Figure 6, the dashcam captures the stop sign with a red bounding box and the label *"Moving"*. This indicates that our algorithm correctly identified the traffic violation by detecting that the vehicle did not come to a complete stop.

Figure 6. Detection of a stop sign with the vehicle moving, indicating a traffic violation.

Here is a demo video of the two stop scenarios. These results validate the functionality of our traffic rule violation detection system. The algorithm accurately distinguishes between compliant and non-compliant behaviors at stop signs.

5. Limitations & Future Work

The primary limitation of the current implementation of the stop sign detection algorithm is the complexity of scene detection, particularly as it also detects stop signs intended for drivers on perpendicular paths. This results in false positives, as the algorithm does not distinguish between stop signs facing the driver and those intended for others. To address this, initial work has begun on an algorithm that analyzes the temporal evolution of a detected stop sign's bounding box. By examining changes in the sign's shape, size, and position within the image, the algorithm attempts to determine if the sign is directed towards the driver. Typically, a stop sign approaching directly towards the vehicle will appear to grow larger and maintain a square shape. However, due to the complexity of road layouts, sign placements, and camera angles, this method has not yet proven reliable. A potential improvement could involve annotating videos with one-hot encoding and performing clustering on the time series data of bounding boxes to achieve more robust results. For instance, Liao's work on Clustering of Time Series provides foundational techniques that could be adapted for improving the accuracy of bounding box time series clustering in this context [9].

Furthermore, the algorithm's processing pipeline needs optimization for real-time application. The goal is to enable real-time inference to promptly alert drivers potentially committing traffic violations. This proactive feature would significantly enhance road safety by integrating real-time alerts for drivers as they approach a stop sign, facilitating immediate corrective action. Such developments require efficient real-time processing frameworks, as discussed by Zhao et al. in their work on optimizing real-time computer vision systems [10]. Additionally, expanding the system to include more comprehensive traffic pattern analysis could lead to broader applications in traffic management and urban planning.

Once enhanced, this method could be proposed to insurance companies to better calculate their risks, potentially allowing drivers to reduce their insurance premiums. Raphael actually plans to focus on developing this startup idea throughout the summer, having identified a market need after discussions with several insurance companies, which currently rely only on telematic sensors.

Lastly, for our project we can only quantitatively evaluate the YOLO results, but not the following steps. We do not have access to labeled video data, so we can only qualitatively evaluate it.

6. Individual Contributions

In this project, we each had specific roles that contributed to the development and enhancement of the traffic sign detection system. Raphael focused on the preprocessing of data given the YOLO training feedback from Huandong. This task involved converting and simplifying the dataset to make it suitable for traffic sign detection applications. Additionally, Raphael was responsible for developing and optimizing the SORT (Simple Online and Realtime Tracking) algorithm and Stop Detection Algorithm, which is used for maintaining the continuity of object tracking across video frames.

Huandong was primarily responsible for integrating and fine-tuning the YOLO object detection model, which is crucial for identifying traffic elements within our videos. Huandong also worked closely with Raphael on implementing and optimizing the SORT algorithm. Additionally, Huandong led the denoising efforts, enhancing the clarity of our tracking results by reducing noise and refining the outputs from the object detection phase.

References

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. 2016. 1, 3

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. *arXiv preprint arXiv:1602.00763v2*, 2017. 1, 3

[3] Tuan Le, Meagan Combs, and Qing Yang. Vehicle tracking based on kalman filter algorithm. 2023. 1

[4] Yifan Xu, Ming Chen, and Liu Wang. Advanced monitoring of traffic violations at urban intersections. *Journal of Urban Traffic Management*, 34(2):122–134, 2020. 1

[5] John A. Smith and David S. Lee. A network of ai-enabled cctv for traffic management and safety. *International Journal of Smart Traffic Monitoring*, 29(1):45–59, 2018. 1

[6] Mark E. Johnson, Sarah L. Rodriguez, and Amit K. Kapoor. Utilizing high-definition cameras for analyzing aggressive driving patterns on highways. *Highway Safety Research*, 45(4):205–219, 2019. 1

[7] Mapillary AB. Mapillary vistas dataset. https://www.mapillary.com/dataset/vistas, 2017. 2

[8] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995. 3

[9] T. Warren Liao. Clustering of time series data—a survey. *Pattern recognition*, 38(11):1857–1874, 2005. 5

[10] Feng Zhao, Yu Wang, and Xiao Chen. Optimizing real-time computer vision systems. *Journal of Real-Time Image Processing*, 21(4):645–660, 2019. 5

Appendix A

Label ID	Label Description	Count
0	R-SpeedLimit	4767
1	R-Yield	1843
2	R-KeepRight	1553
3	W-PedestrianCrossing	1492
4	R-Stop	1158
5	R-NoEntry	1141
6	R-NoParking	1010
7	R-NoHeavyVehicles	999
8	W-RailroadCrossing	830
9	R-NoOvertaking	818
10	W-MergeRight	726
11	W-RoadNarrows	639
12	W-CurveLeft	621
13	W-CurveRight	579
14	W-SchoolZone	566
15	R-OneWayLeft	561
16	W-Roundabout	542
17	W-Roadworks	521
18	W-SlipperyRoad	477
19	W-TrafficSignals	475
20	R-OneWayRight	470
21	R-NoUTurn	465
22	W-OtherDanger	387
23	W-Crossroads	357
24	W-AddedLaneRight	292
25	R-BicyclesOnly	286
26	R-NoTurnOnRed	245
27	W-WildAnimals	217
28	W-BicycleCrossing	157
29	W-SteepAscent	155
30	W-FallingRocks	131
31	W-TwoWayTraffic	130
32	R-LeftLaneMustTurnLeft	103

R stands for Regulatory and W stands for Warning.

Appendix B

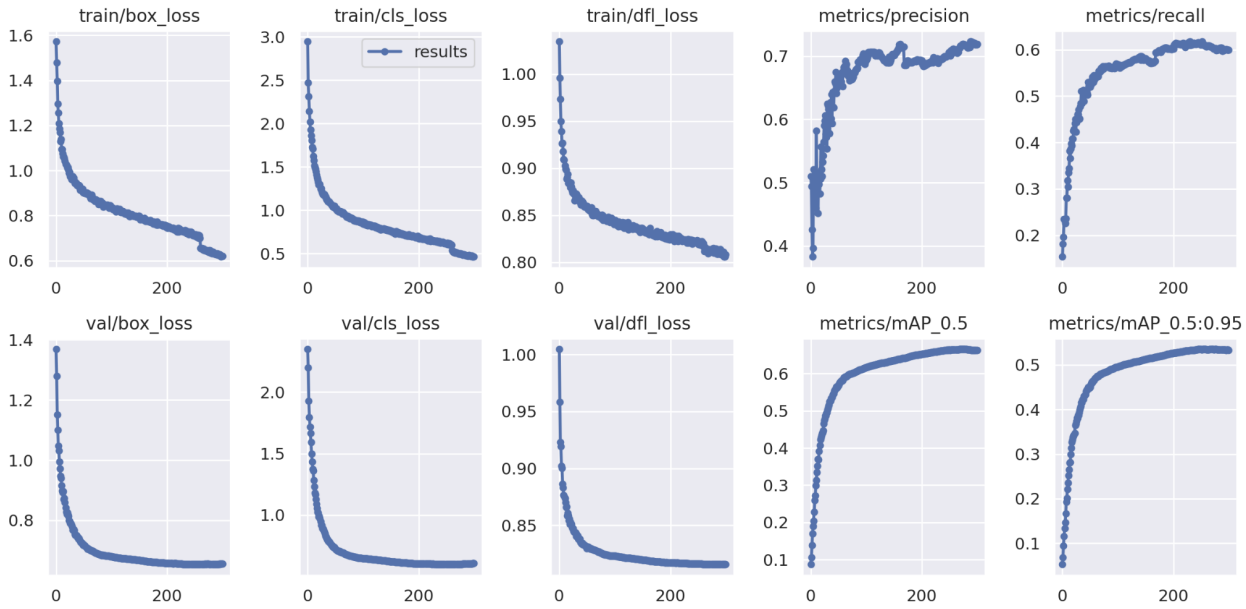

Figure 7. Metrics of Validation Set During Training. Everything goes as expected; in other words, losses are generally decreasing with more epochs, and precisions and recalls increase with more epochs. We can see a sharp drop around epoch 250, that is when we close the mosaic augmentation. Mosaic Augmentation can increase robustness for earlier epochs.

Appendix C

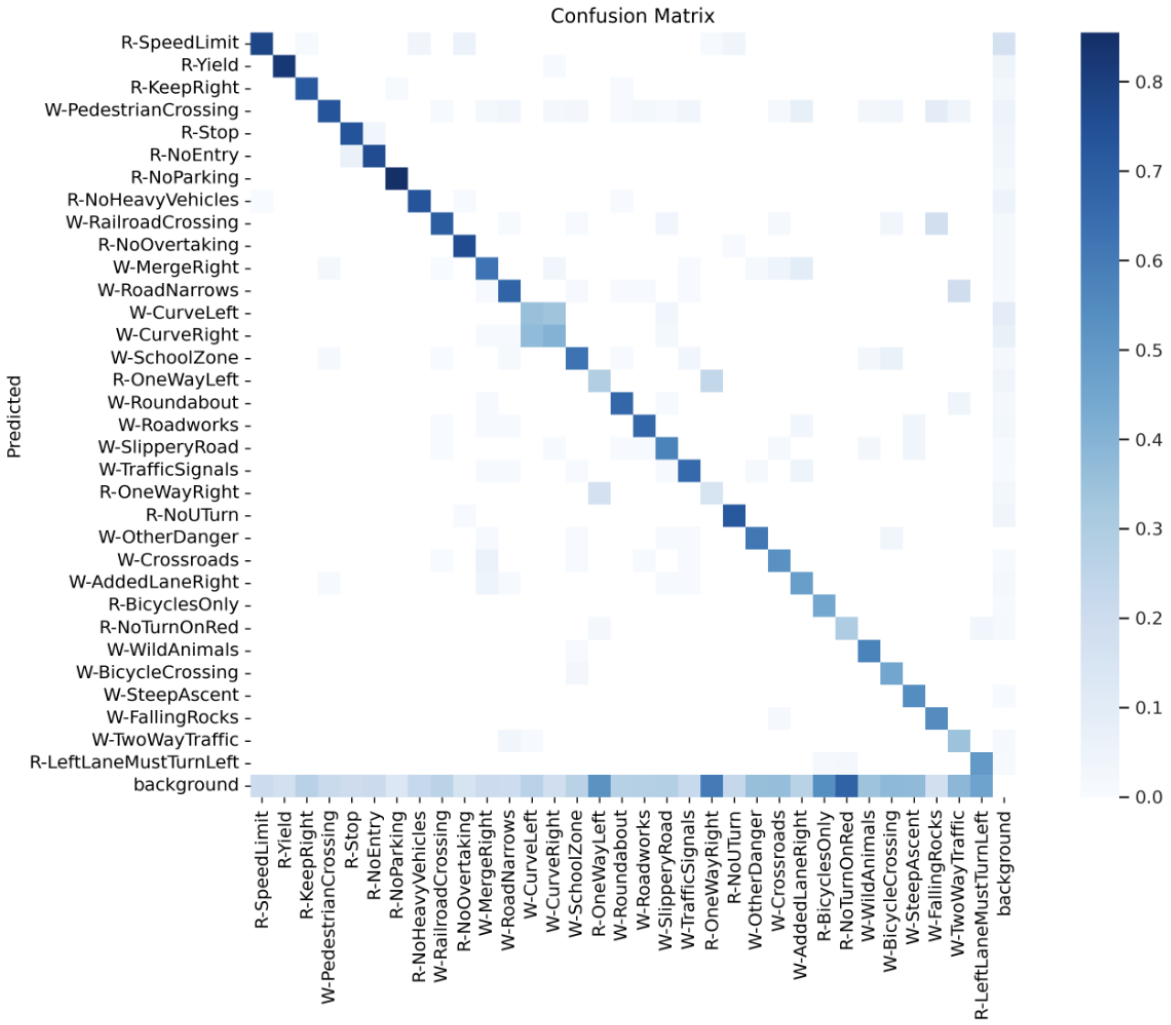

Figure 8. Confusion Matrix of Finetuned YOLO (Validation Set). Our algorithm has a lot of false negatives where they predict traffic signs as backgrounds, but this makes sense as we set the confidence threshold to be high (0.5). Also, we can tell the algorithm sometimes mixes up between left and right (such as between Curve Left and Curve Right). Other than this, our algorithm can decently detect traffic signs.