Road Detection

Part 3 – Vehicle & Sign Detection

EEL 6990 Project 3

Martin Kadzis

Description

Road detection is a computer vision topic popular among electric vehicles and their ability to comprehend the environments around them. In the real world, automated vehicles are able to continuously process their surrounding through technology like lidars and photo sensors, but for this project, a video footage of the driving scenes will be used to mimic the continuous aspect.

Road detection involves outlining the current lane the vehicle is in with guiding highlights, classification of other cars, pedestrians, road signs, and stop lights with rectangular bounding boxes labeled respectively.

A good solution to the topic should consider these criteria:

1. The current lane should be identified and highlighted correctly
2. All nearby cars, traffic signals/signs, and people should be detected and labelled correctly
3. Objects should not be falsely detected
4. Edges of bounding boxes should be placed as closely to the object as possible

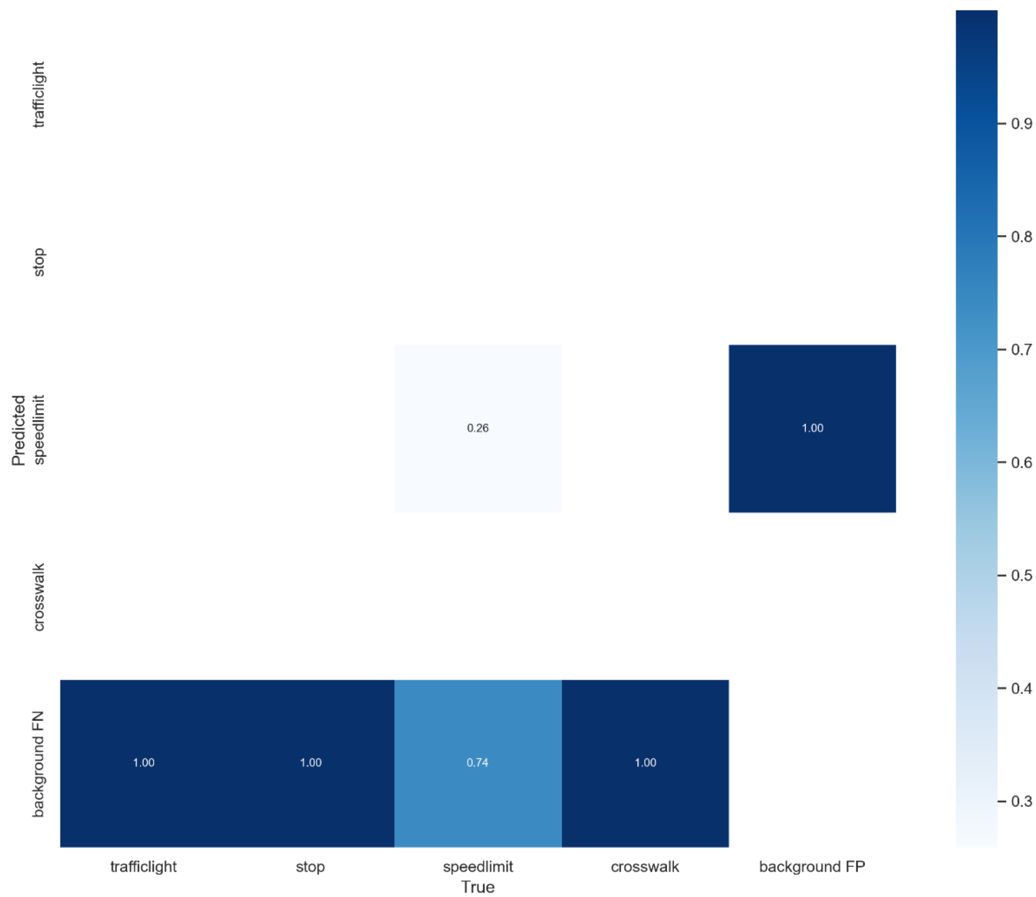
[Explain the purpose of this project]
For this project part, the goal is to examine the differences in performance and accuracy between a default model versus trained model. To do this I will explore creating training data and labels for YOLOv5, the established object detection model from project part 2, and train classifiers for traffic signs that are outside of YOLOv5's base model. YOLOv5 utilizes a detection algorithm that divides the video frame into a grid system with each grid cell responsible for its own local detection, giving it usability for real time analysis.

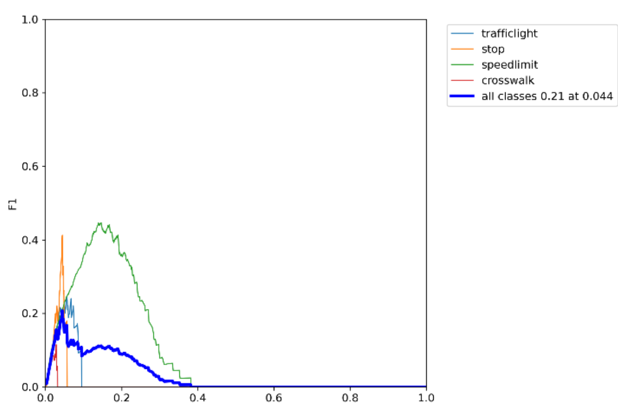
For classification and labeling, I am taking a preexisting large dataset of traffic entities such as: traffic lights, stop signs, speed limits and crosswalks and converting the annotation or labeling to that of YOLOv5 and training for the weights.

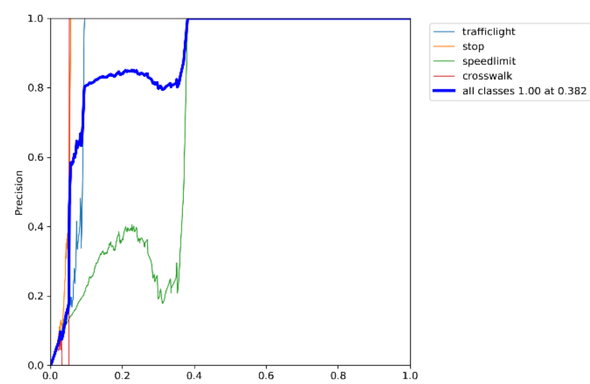
For measurement, I have systematized several driving footages on Nine Mile Road involving varying levels of traffic, whether its bright or dusk, and at a stopped intersect. This will help give comparisons between times of day traffic and lighting.

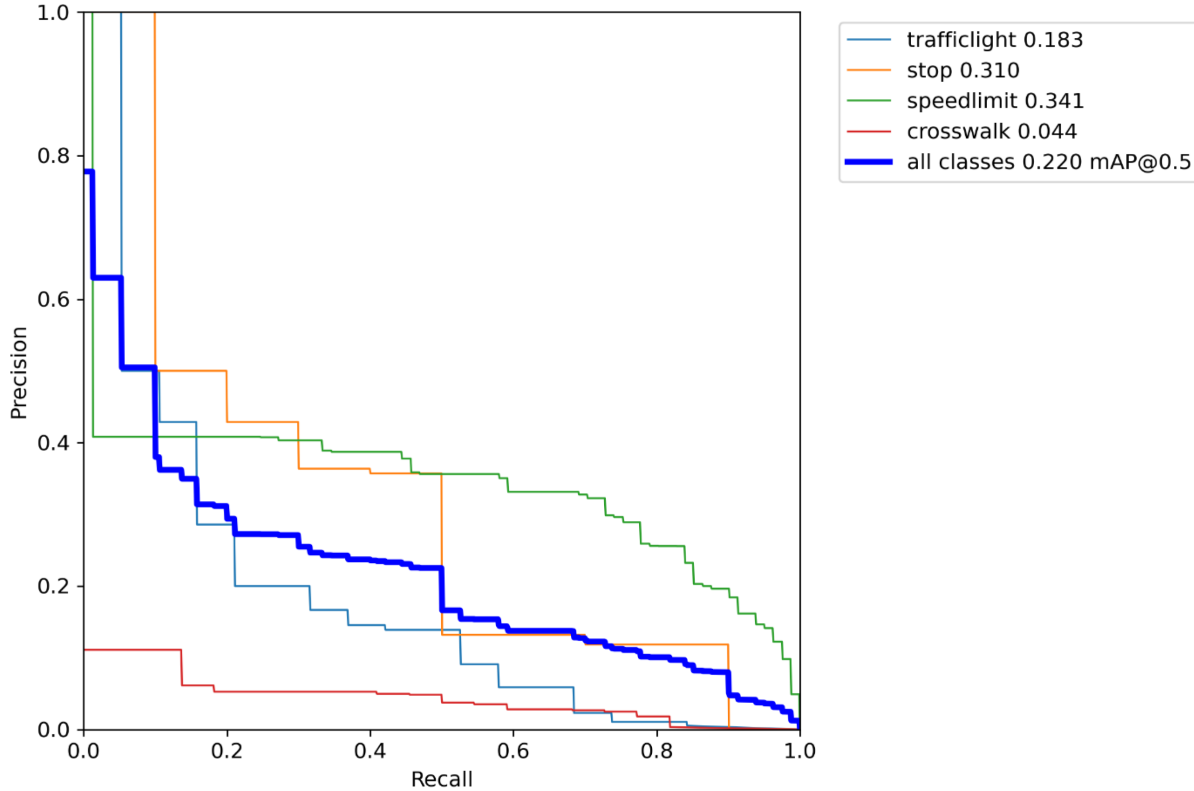
Results

yolov5x_street epoch 1 training results:

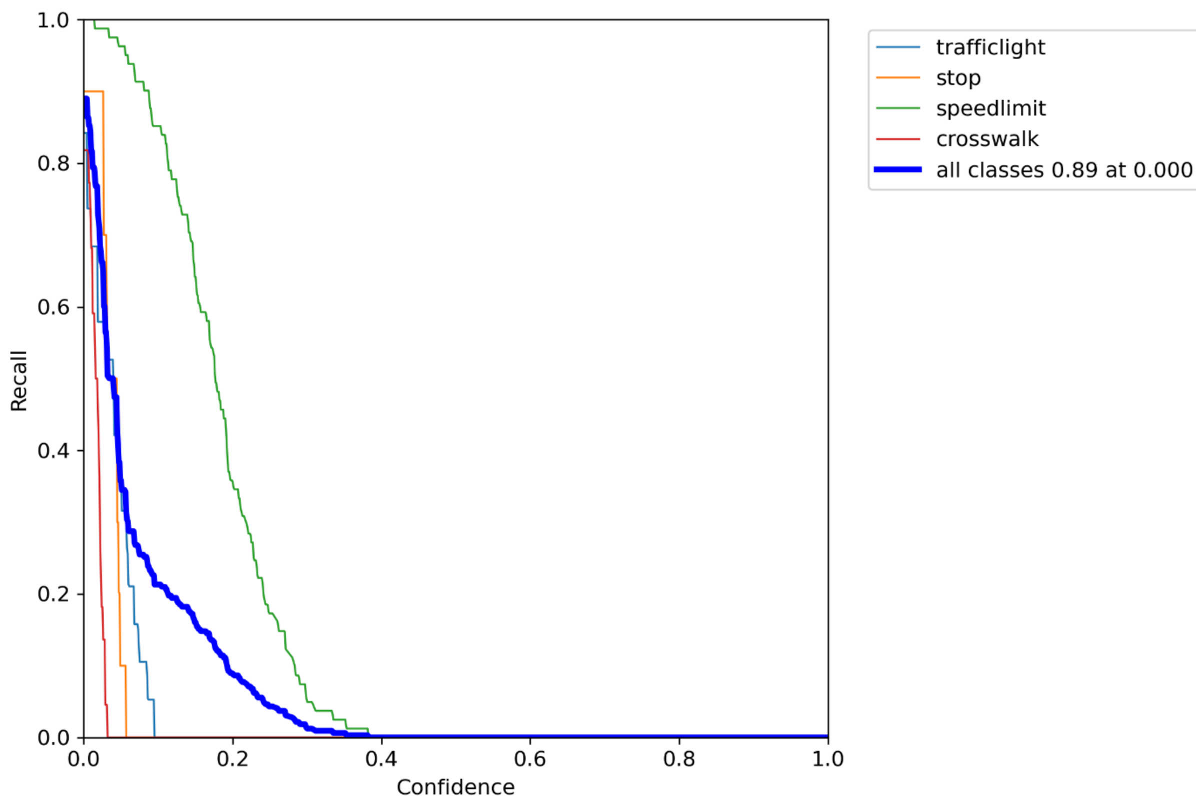






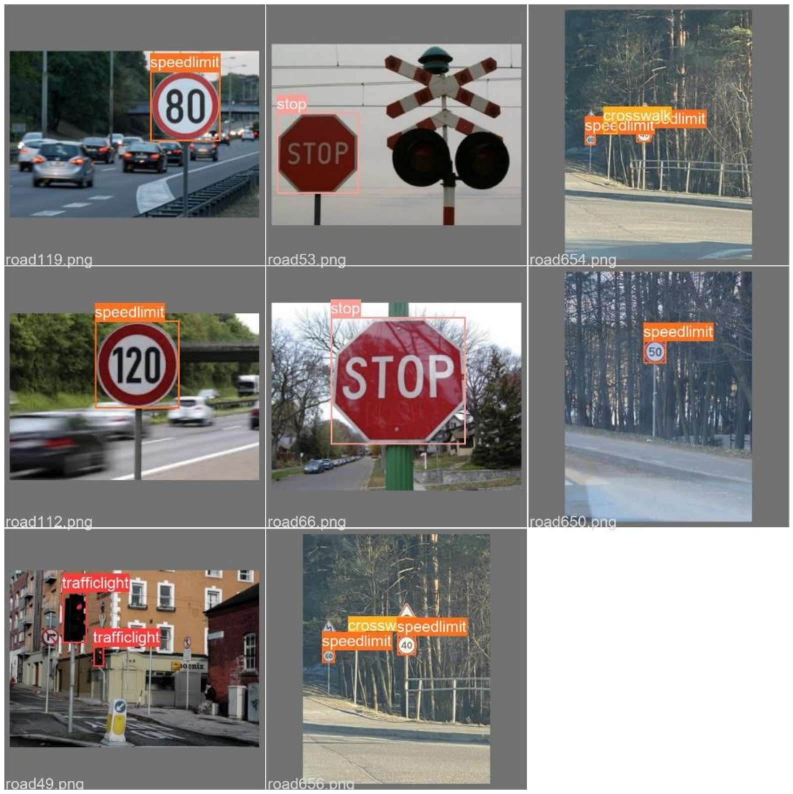


Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
trafficlight 0.183
stop 0.310
speedlimit 0.341
crosswalk 0.044
all classes 0.220 mAP@0.5

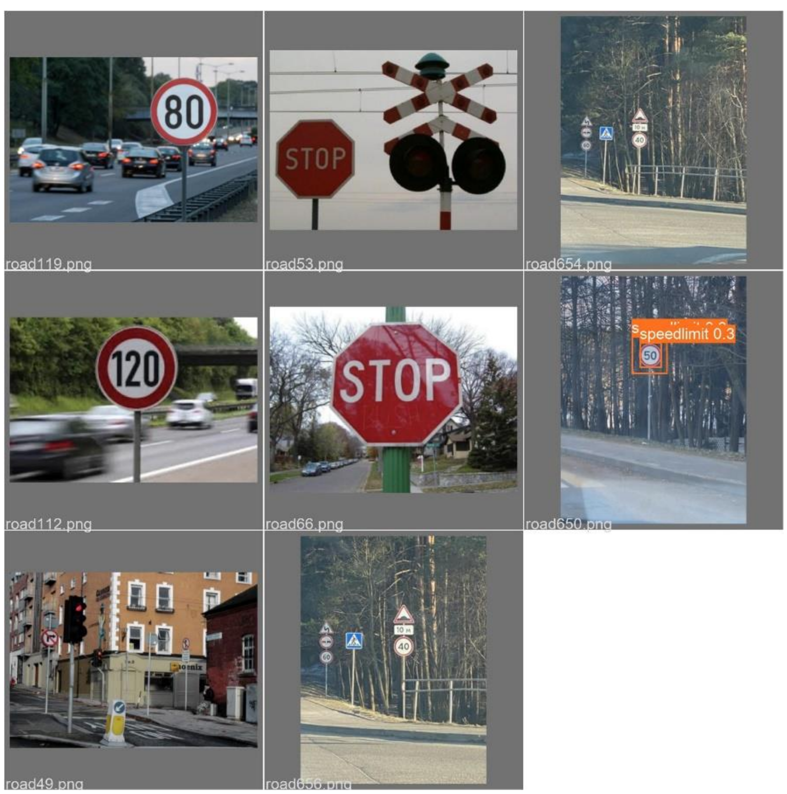


Recall
Confidence
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
trafficlight
stop
speedlimit
crosswalk
all classes 0.89 at 0.000

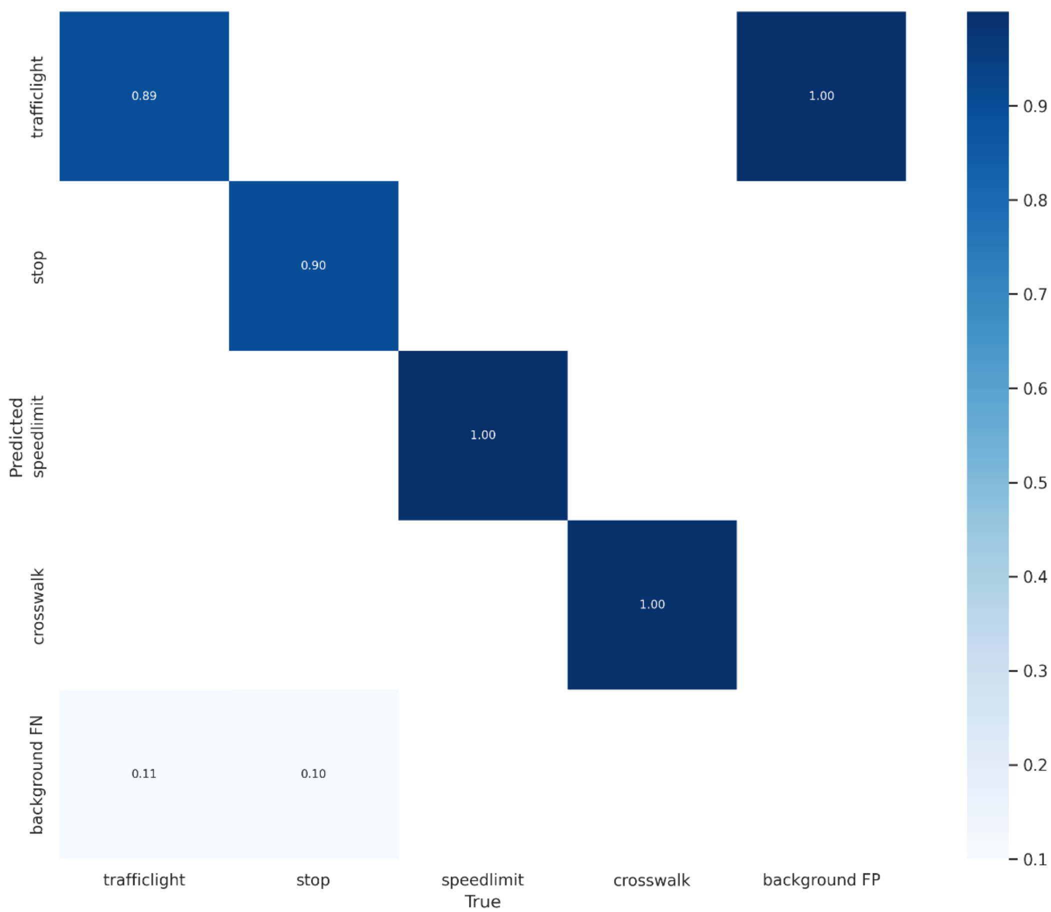
label:

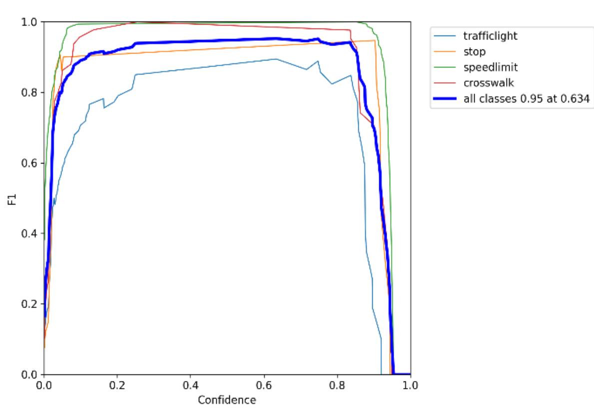


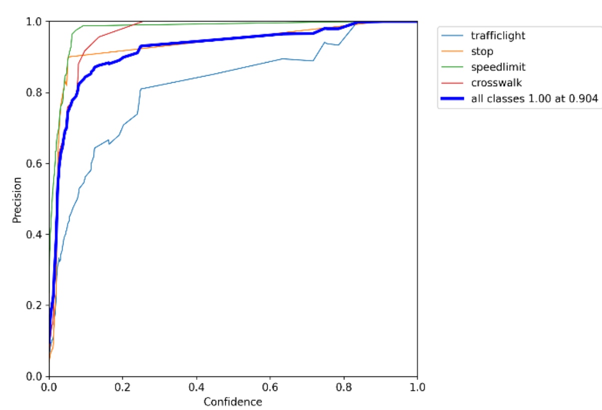
prediction:

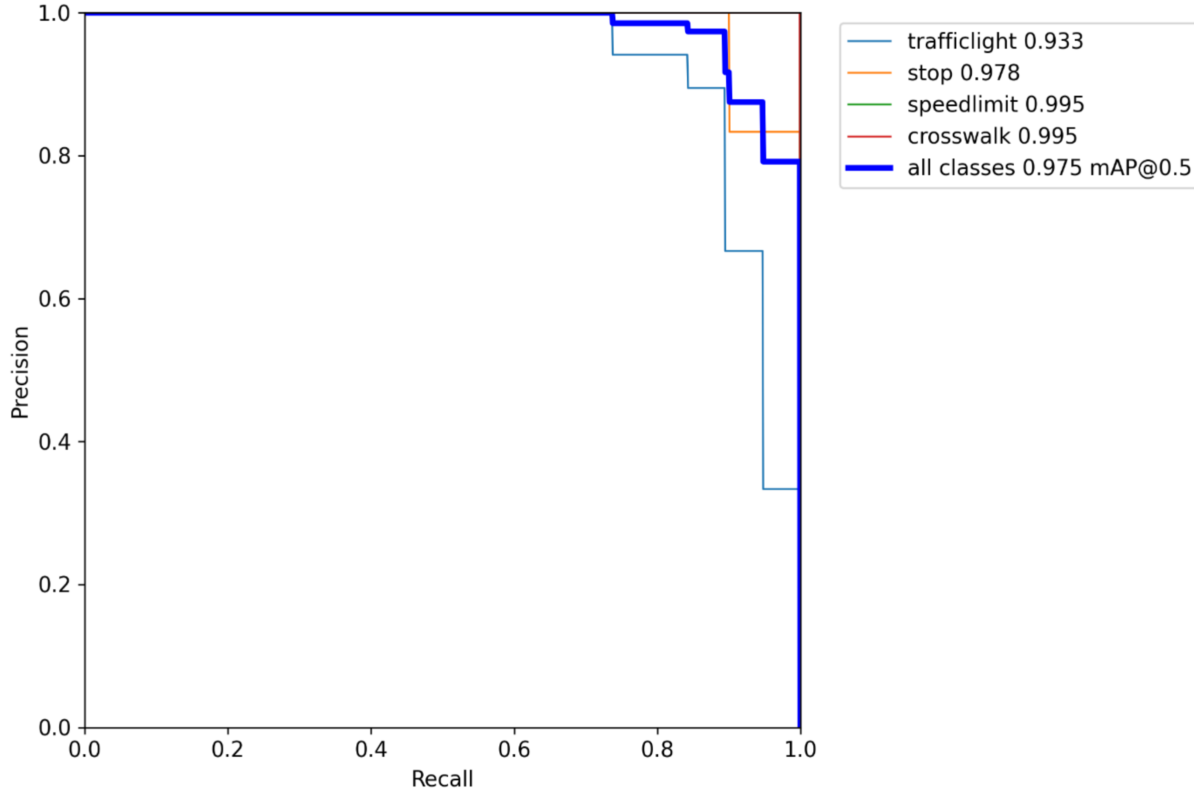


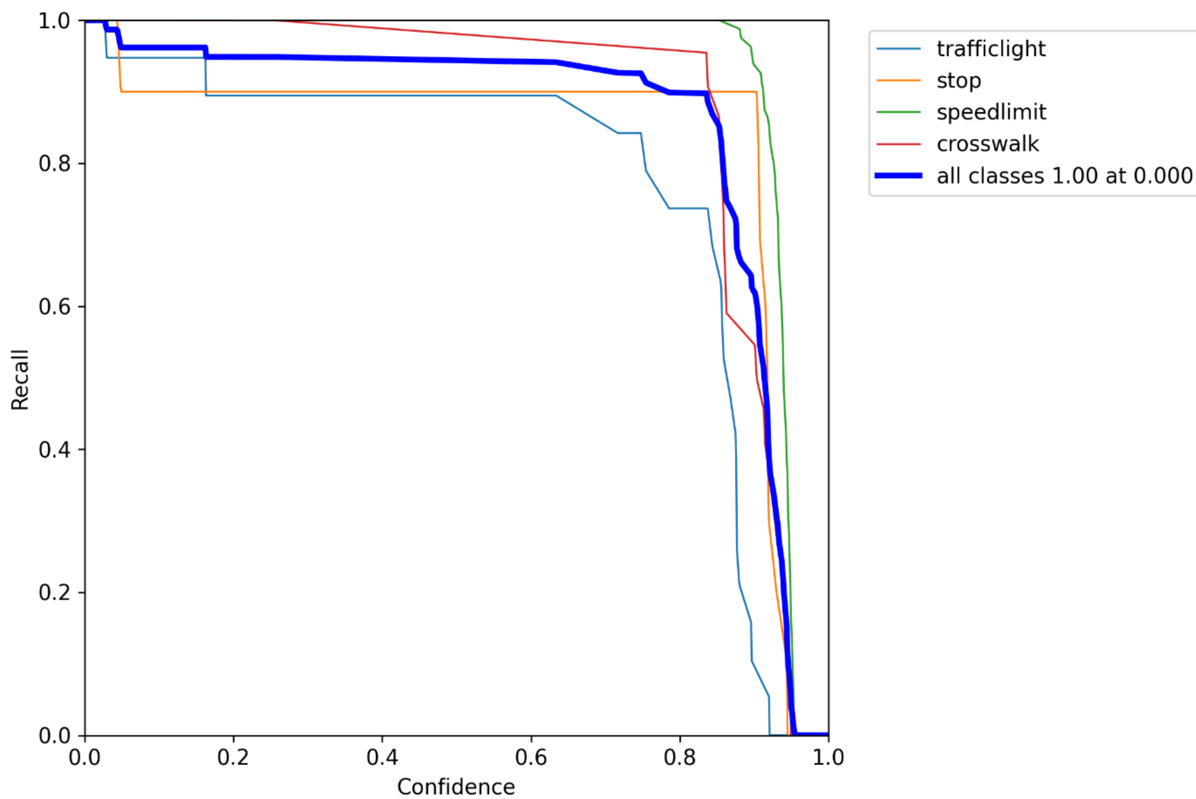
yolov5x_street epoch 20 training results:



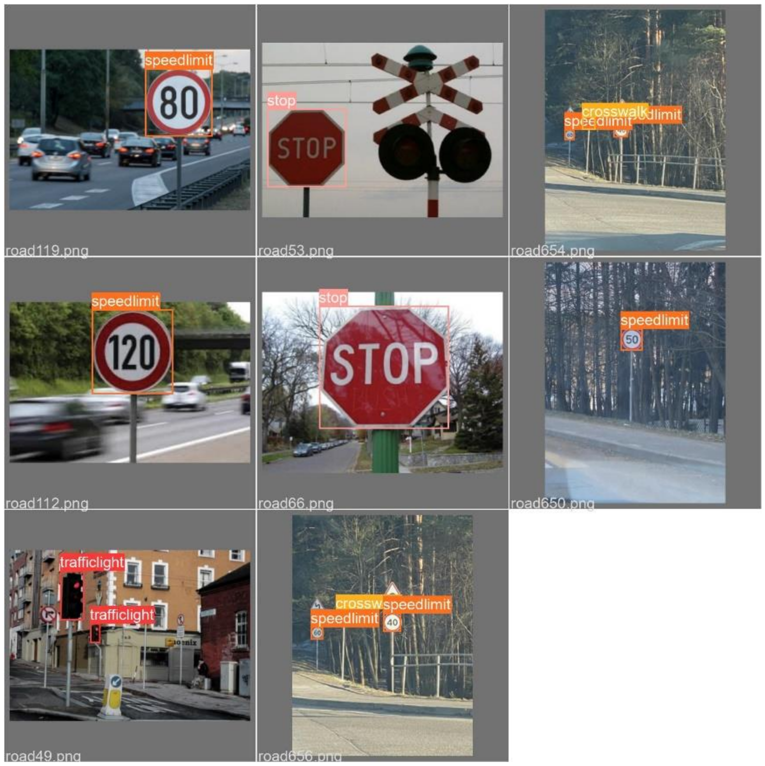




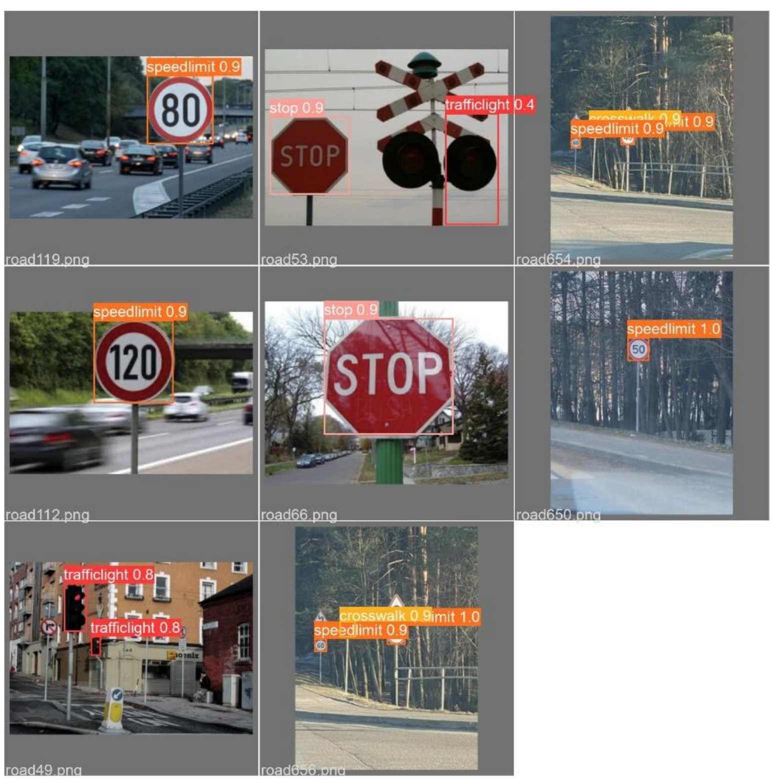

trafficlight 0.933
stop 0.978
speedlimit 0.995
crosswalk 0.995
all classes 0.975 mAP@0.5
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0


trafficlight
stop
speedlimit
crosswalk
all classes 1.00 at 0.000
Recall
Confidence
0.0
0.2
0.4
0.6
0.8
1.0

label:



prediction:



Vehicle: 60% , Traffic Signs: 45%

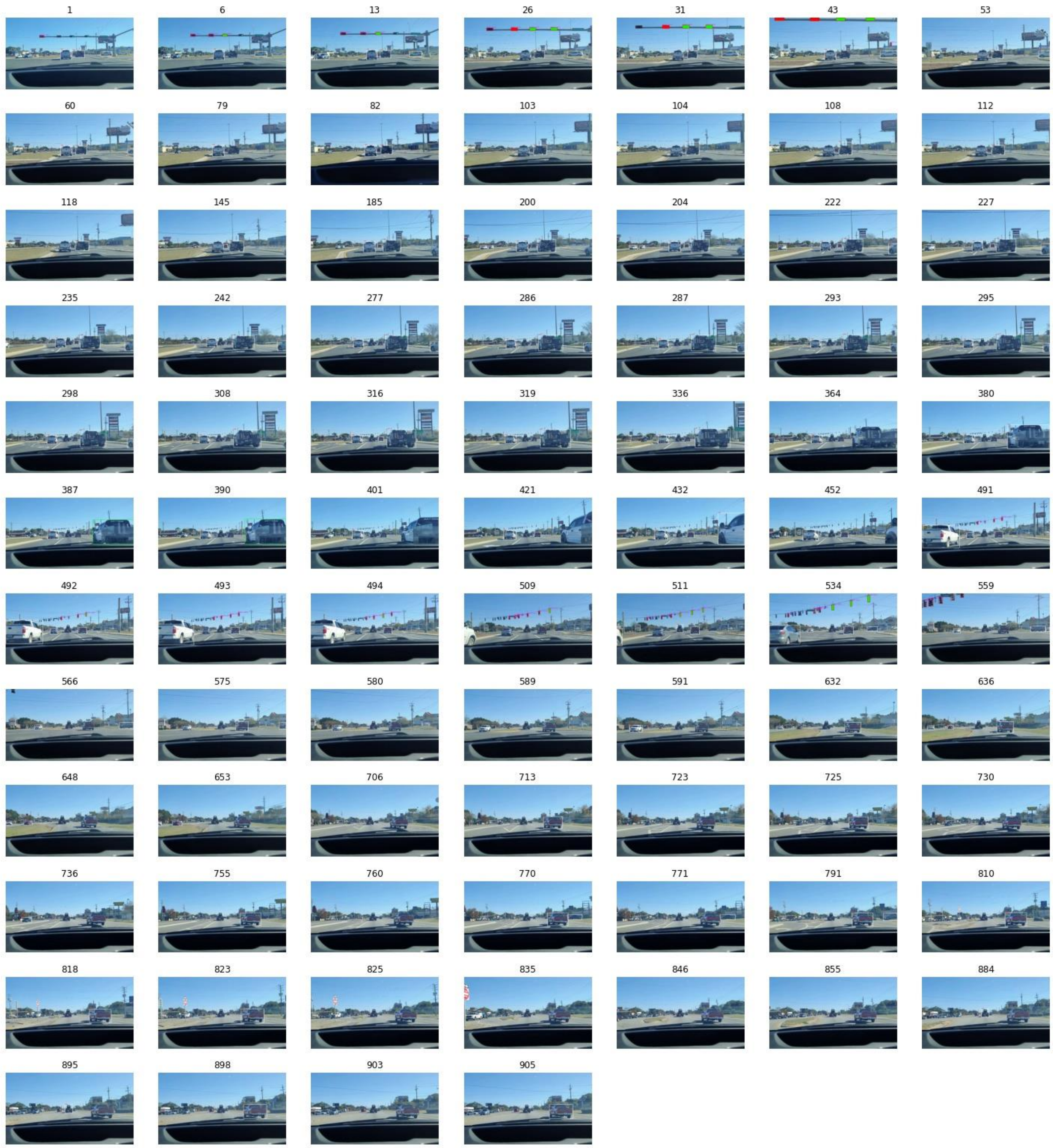
Total Detection per Frame (binary)

Training Model	Trial 1	Trial 2	Trial 3	Avg
Day Medium E1	3/81	5/80	3/73	4.690%
Night Heavy E1	30/79	23/76	22/75	32.52%
Night Stop E1	--			
Day Medium E20	78/81	75/80	70/73	95.31%
Night Heavy E20	49/79	53/76	53/75	67.48%
Night Stop E20	--			

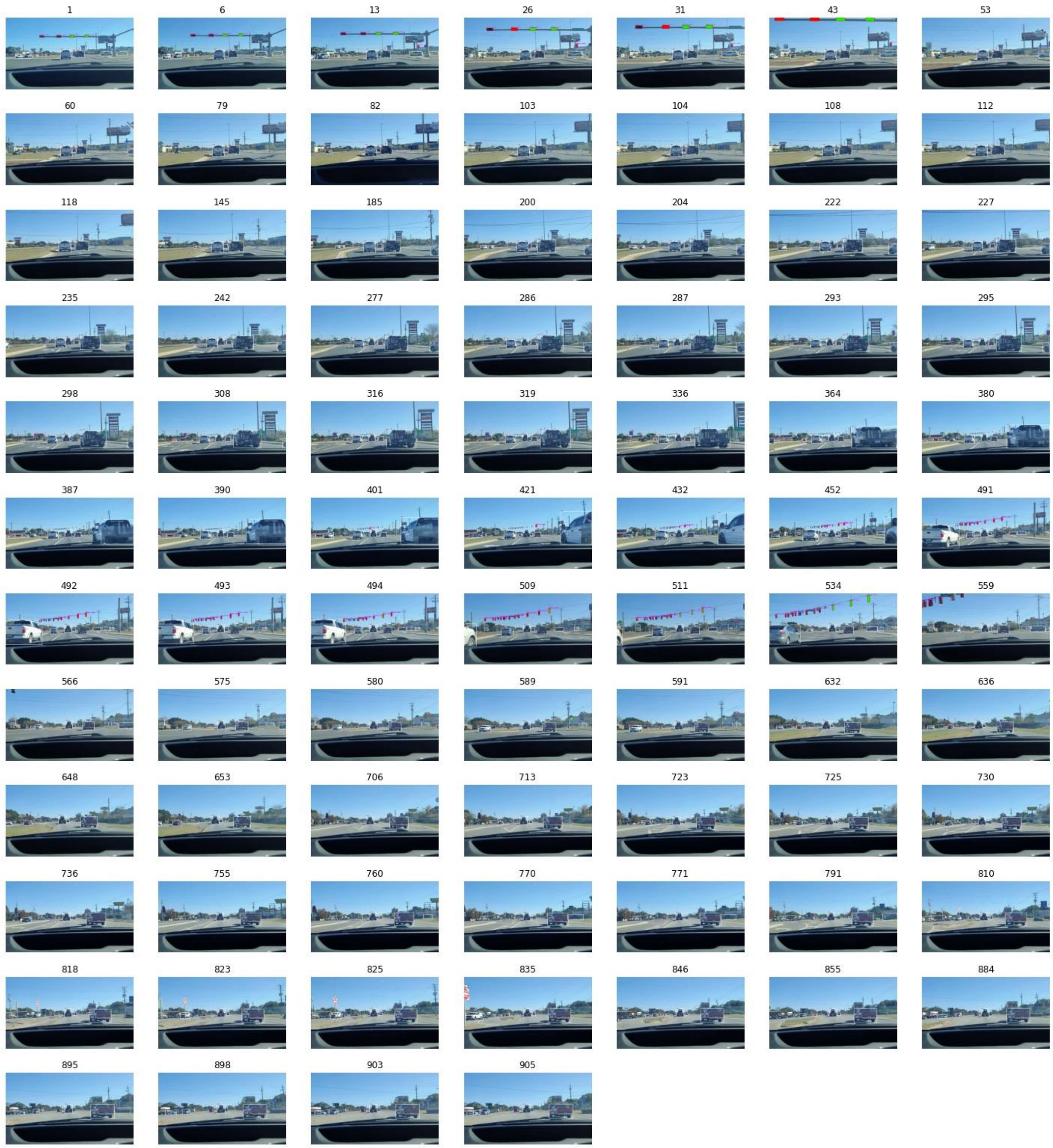
Total Frames with Detection

Training Model	Trial 1	Trial 2	Trial 3	Avg
Day Medium E1	71/81	70/80	67/73	88.98%
Night Heavy E1	79/79	76/76	75/75	100.0%
Night Stop E1	--			
Day Medium E20	81/81	80/80	73/73	100.0%
Night Heavy E20	79/79	76/76	75/75	100.0%
Night Stop E20	--			

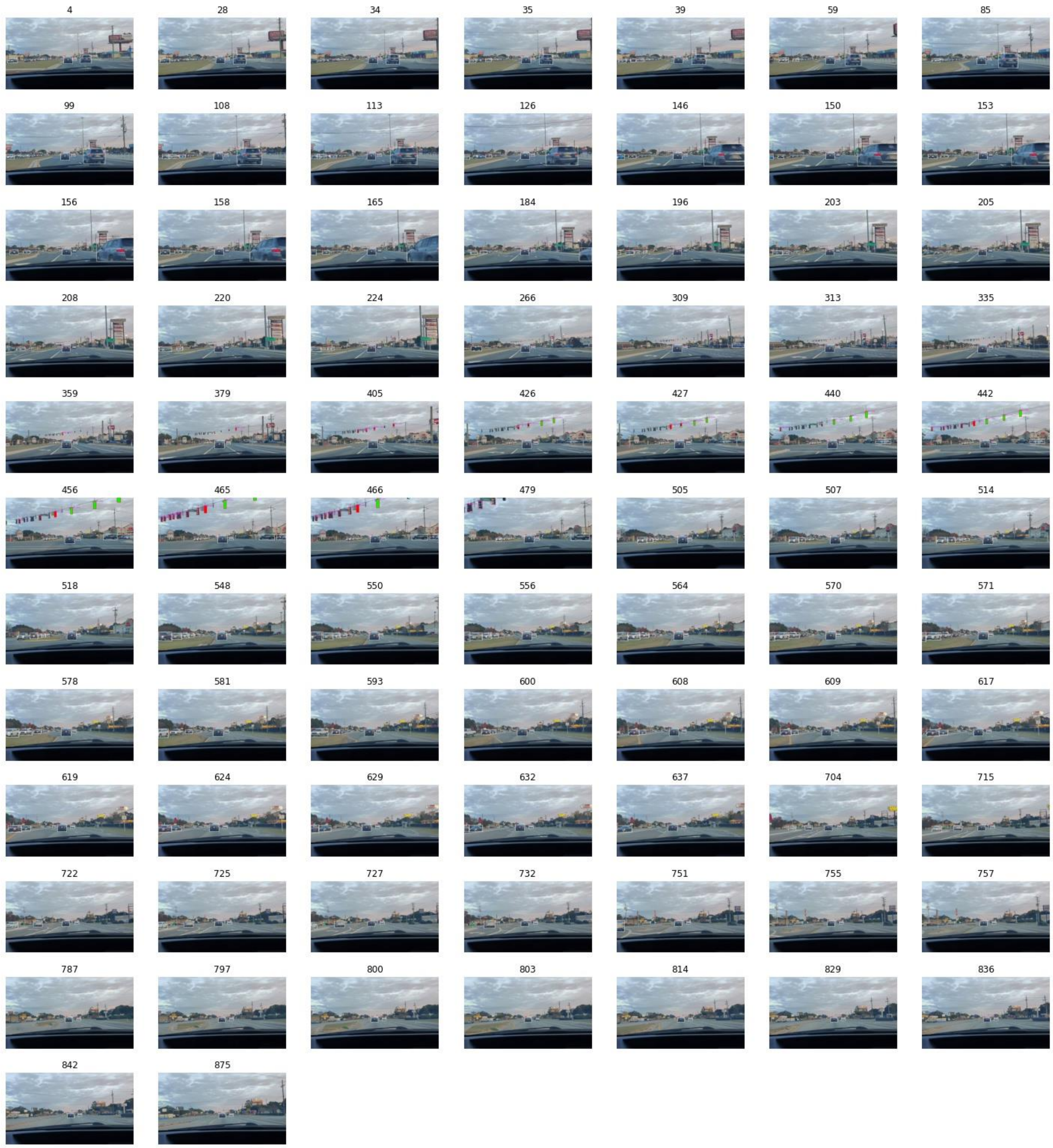
Trial 1: day-medium epoch 1



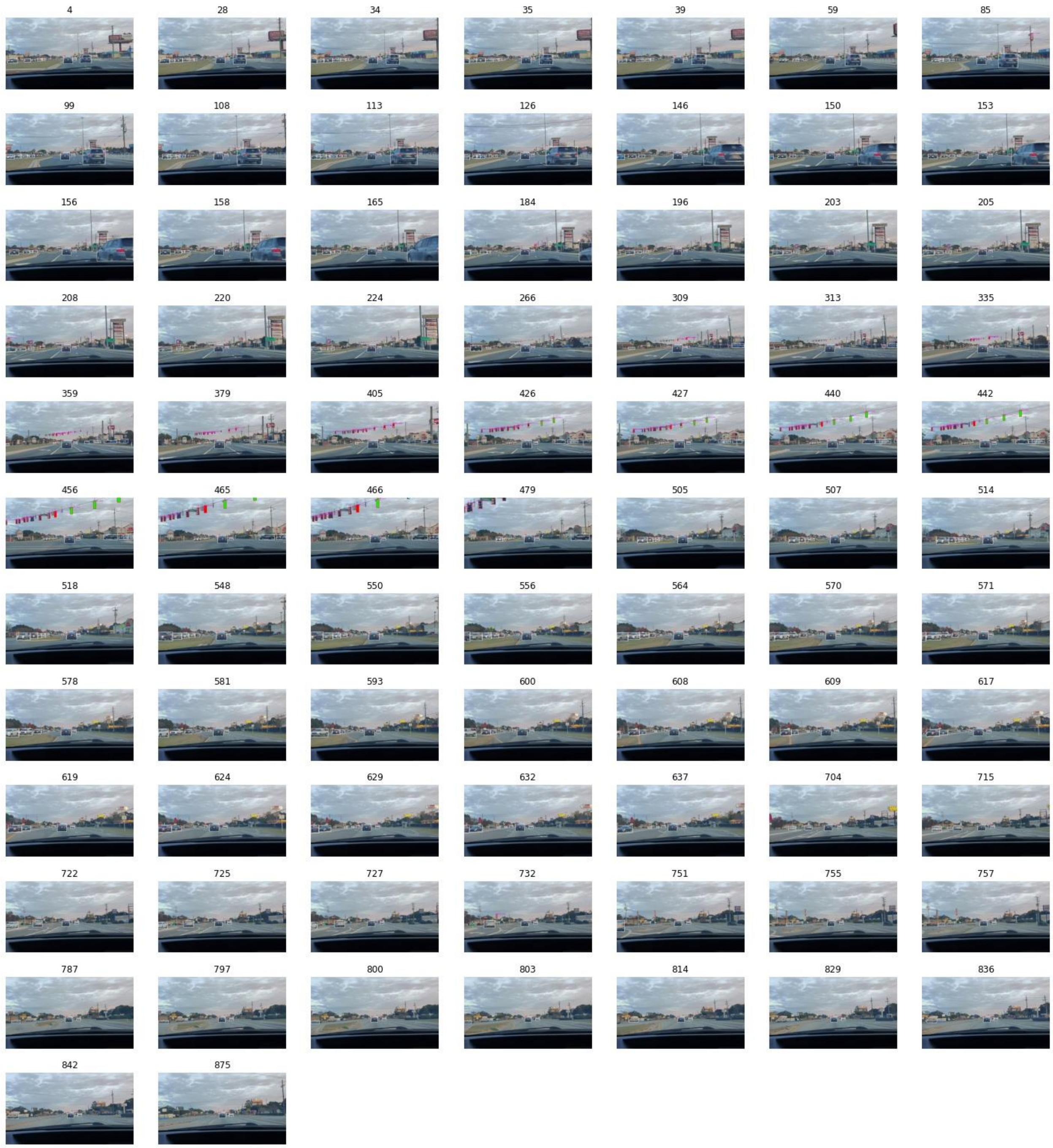
Trial 1: day-medium epoch 20



Trial 1: night-medium epoch 1



Trial 1: day-medium epoch 20



Trial 1: night-stop epoch 1

Trial 1: day-medium epoch 20

Analysis

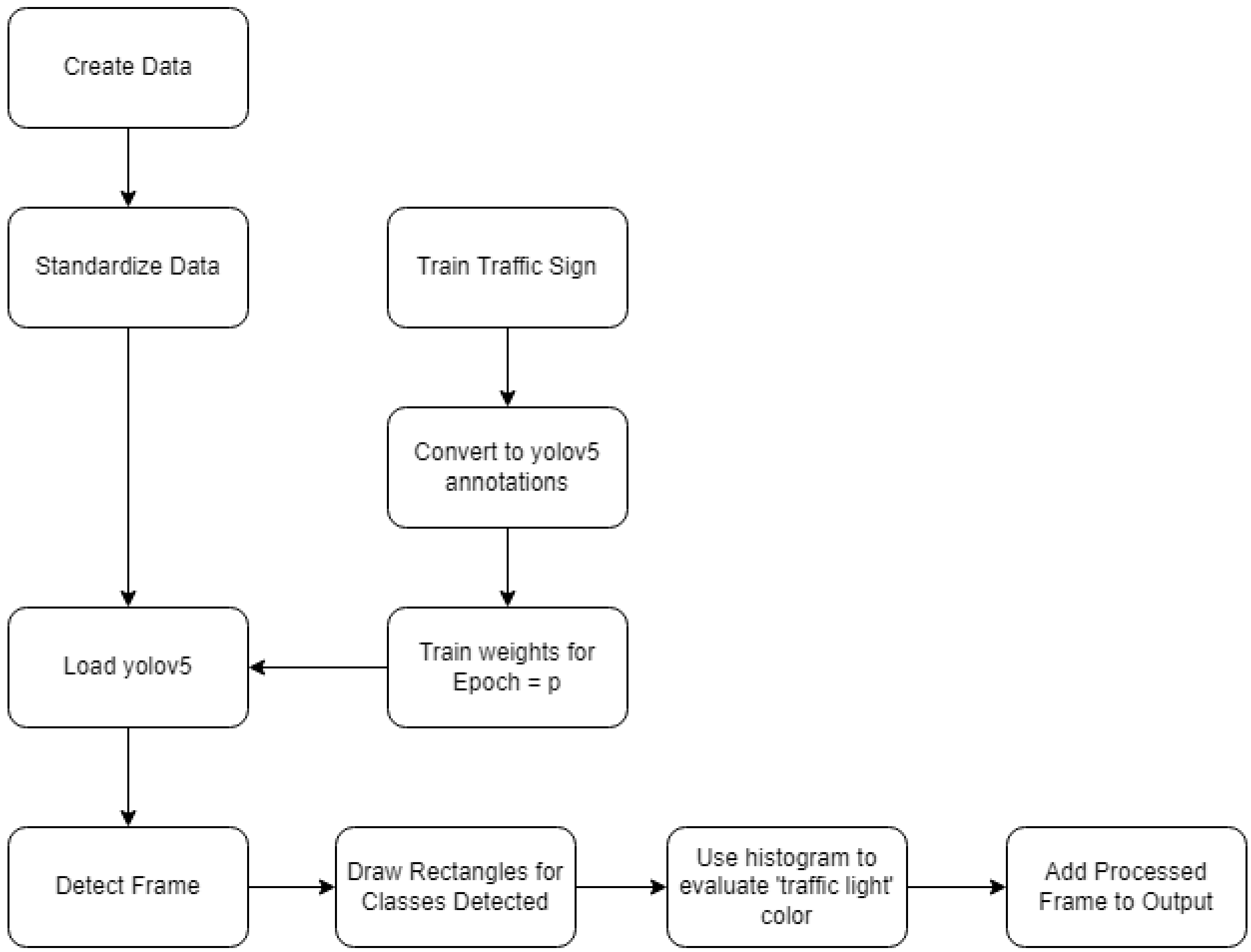
During this project's exploration, after training weights for the yolov5_street file, I realized that the dataset was a collection of Belgium traffic signs, so in the latter stages of evaluation, the various U.S speed limits and median signs are not recognized.

In addition, the results for night-stop, the twilight driving footage for stopped intersect, was ultimately disposed due to too many varying cases from the other two footages. The video had continual detection for stationary traffic lights aside from moving vehicles across the lane, this already created a separate criteria for detection accuracy. The video comparison also had controversial results with the weaker model producing better detection for traffic lights than the stronger model, while maintaining its worse vehicle detection. (head cannons for this being my traffic light histogram that was not prepared for the luminosity of traffic lights during nighttime that greatly affects the color and density.

In respect to the results above, its easy to say the bigger and more trained model provide overall better results for detecting more objects and more types of objects. But because of the lack of time to produce trained weights for other epochs (1 epoch iteration for my computer takes 45-50mins, 30 epoch is around 17.6hrs), for longer and more day/night processed videos (1 video, 30secs, takes around 40-45mins), it is hard to take concrete conclusions regarding the impacts of traffic density and time of day. (NOTE: the current script for processing each video utilizes essentially 3 individual models, preferably I would've liked to do combined the vehicle and traffic detection to one model)

Flow Chart

Note: Previous features frame standardization and lane detection were removed due to new focus on vehicle and traffic sign detection. YOLOv5 breaks the image as whole into smaller regions for analysis, there for no reason to standardize specific frame sizes.



Create Data

The data is in the form of video captures (mp4) of a front cam driving footage of Nine Mile Road, Pensacola. Each sample video has been trimmed to 30 second footages separated depending on time of day and traffic density level.

Train Traffic Sign

- Convert to yolov5 annotations: the imported dataset's annotation/label for images are in xml, this process returns a txt file that formats the necessary information in yolov5 format.
- Train weights: Using the converted yolov5 labels and class, I can use the train.py in yolov5's cloned repository to get the best weights for dataset.
- YOLOv5s: Apply the trained model on the image frame, overlay identification boxes around specifically cars and traffic lights. If the object class is a traffic light, perform a histogram analysis that checks for the highest color density to determine if the light is red, yellow, or green.

Acceptable Results

For this project part, I'm not testing the real-time capabilities of the model since there are more elegant ways combined the capabilities of the models i had into one. With this, an acceptable result is a distinguishable difference between the pretrained and enhanced yolov5 model. This performance difference can be derived from increasing the confidence threshold for classes drawn.

Appendix

Git repository: https://github.com/oogway-boogway/Road-Object-Detection.git

In the repository, there is a *Project3.* that is the google collab file I used to batch run sample frames for epoch 1 and epoch 20 comparisons