Real-Time Detection of Vandalized, Obstructed and Fake Stop Signs using YOLOv8

Devson Butani, Kim Lam, Sydney Ross, Nicholas Paul, CJ Chung

College of Arts & Sciences, Lawrence Technological University

INTRODUCTION

Autonomous vehicles (AV), also known as self-driving cars, have the potential to revolutionize transportation by offering increased safety, efficiency, and convenience. One of the most crucial aspects is the ability to accurately detect and interpret road signs, one of them being stop signs. Stop sign detection is a fundamental feature for AVs, which is critical for regulating traffic flow and preventing collisions. Accurate stop sign detection under varying environmental conditions is paramount to ensure the safety of passengers, pedestrians, and other road users.

LTU has been a part of the Intelligent Ground Vehicle Competition (IGVC)'s Self-Drive category since 2017. Recent rule changes[1] required AVs to not only detect stop signs but also avoid fake stop signs. For example, a sign that says "SOUP or "SELL". Previously, the team used a feature based Haar Cascade algorithm[2] (HAAR) to detect stop signs. This algorithm depended on identifying features like color and shape (red circles) that are within a certain size range. Under perfect lighting conditions, it is very fast and accurate from a distance. However, it fails as soon as any other variables are introduced; Different text, reflections off of shiny surfaces, brake lights at night, and snow are such examples. The Haar algorithm has high false-positive detections as well. For example, the red base in Fig. 1 is falsely detected. Moreover, it identifies signs that are similar to stop signs, but are ultimately not stop signs (or fake in our use case). To adapt the algorithm to all these variables was not user friendly due to the high complexity of the feature identification system. It required lengthy manual tuning of parameters and in depth knowledge of the algorithm. Additionally, adding features such as text recognition and classification between real, obstructed, vandalized or fake stop signs is difficult.

[Figure 1: HAAR detection of a stop sign.] This photo illustrates how the HAAR algorithm detects stop signs on a road.

Our main functional objectives for the AV application are:

- Detecting stop signs from a reasonable distance
- Analyzing if they are real, fake, vandalized, or obstructed
- Calculating the pixel area (size) of the detection to use as a measure of when to stop the vehicle.

In this project, we explore using YOLOv8, one of the latest advances in object detection models, to try and improve our AVs performance. We integrated our custom trained YOLOv8m model using its python API into our AV platform's modular ROS stack as a direct replacement of the HAAR algorithm. Both algorithms take in the camera feed as input and output whether a stop sign is detected and how big it is. Since the AV operates on detecting whether there is a real stop sign present or not, we will compare the results using this as a metric.

DATASET & MODEL

Dataset

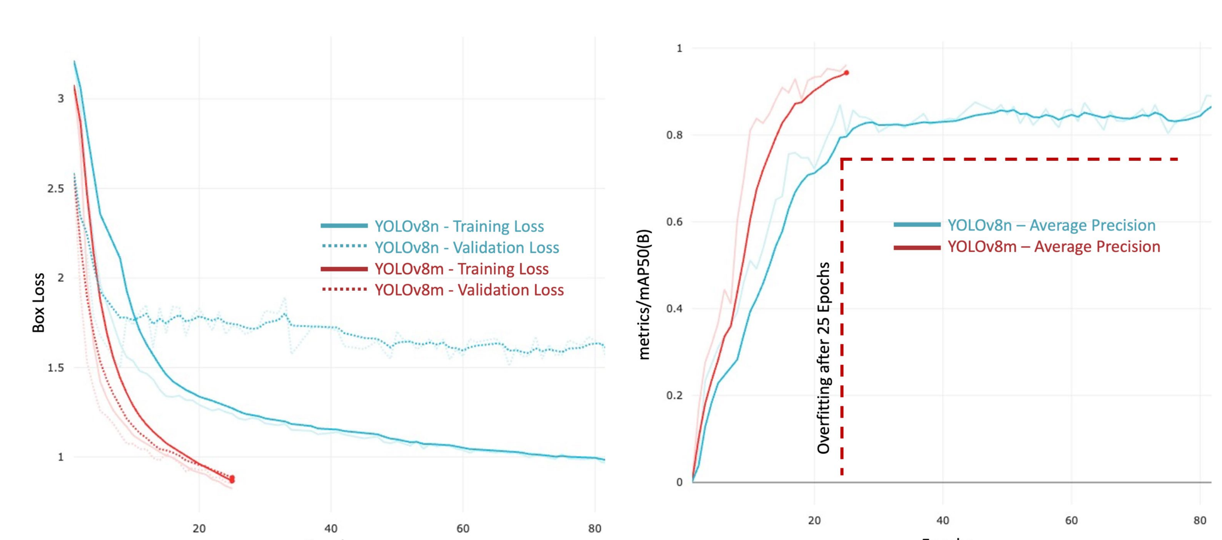
Since we are aiming to detect stop signs that are not in ideal conditions, there were no available datasets that contained stop signs in bad conditions. There are many large datasets that contain traffic signs, but not nearly enough samples of just stop signs in bad conditions. There was a small dataset provided from the previous team's work, but more data was needed to address the following four categories of stop signs:

- Regular Stop Signs were quite easy to find in other traffic sign datasets, though they needed manual separation from the other non-stop signs. Several sources include free Kaggle datasets, published research papers, general Google image searches, and stock images.
- Vandalized Stop Signs were rarely in any dataset, so these images had to be manually collected from Google image searches, stock photos, and even via manually photo editing images of regular stop signs.
- Obstructed Stop Signs were not present in any datasets found, so these images were also manually collected from Google image searches, stock photos, and photo editing.
- Fake Stop Signs, even though they resemble real stop signs from far, they were not in any datasets. Most of these images were manually collected from Google image searches and manual photo editing.

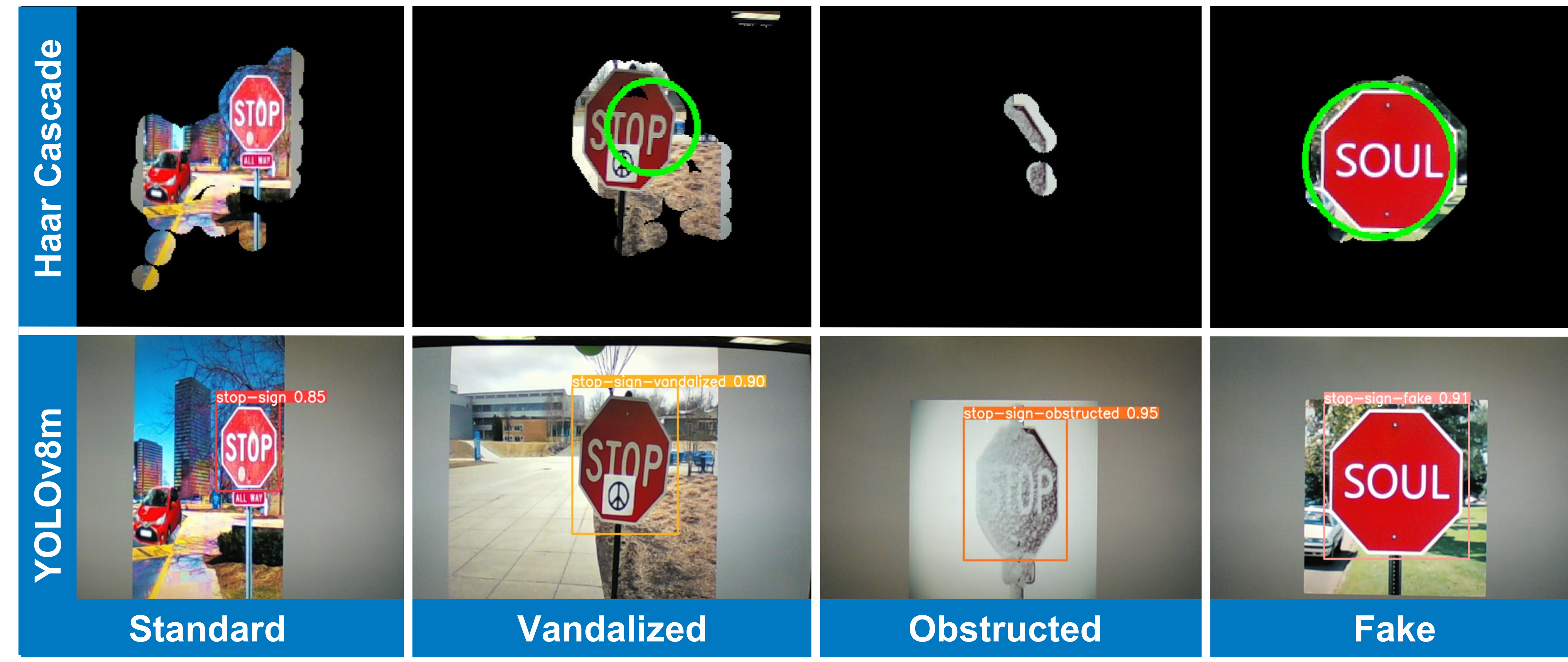
[Figure 2: (Left to right) Examples of standard, vandalized, obstructed, and fake stop signs.]

Model

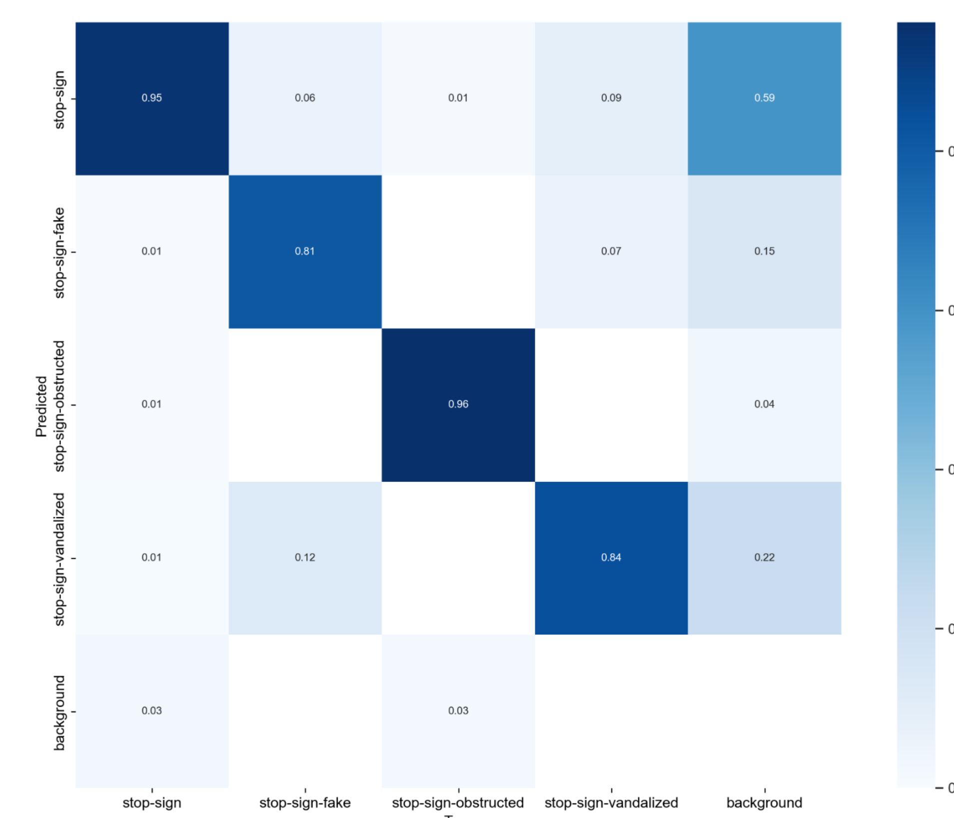
Compared to varying lighting conditions, hues or exposures, we found image resolution to drastically affect how far the model is able to detect a sign. For training our model, we had to closely match dataset image resolution with the camera feed in the AV. However, we also trained on close ups with lower resolution so that our model can be used alongside other models. After realistic data augmentation like rotation, sheer, contrast and exposure, our YOLOv8m model reliably achieved over 80% confidence when 25ft from the sign. Detecting fake, obstructed or vandalized signs is where our model is significantly better compared to the Haar algorithm. Moreover, using the built in PyTorch API allowed us to offload the inference to the GPU, and thus we were able to preserve the fast cycle time of the Haar algorithm.



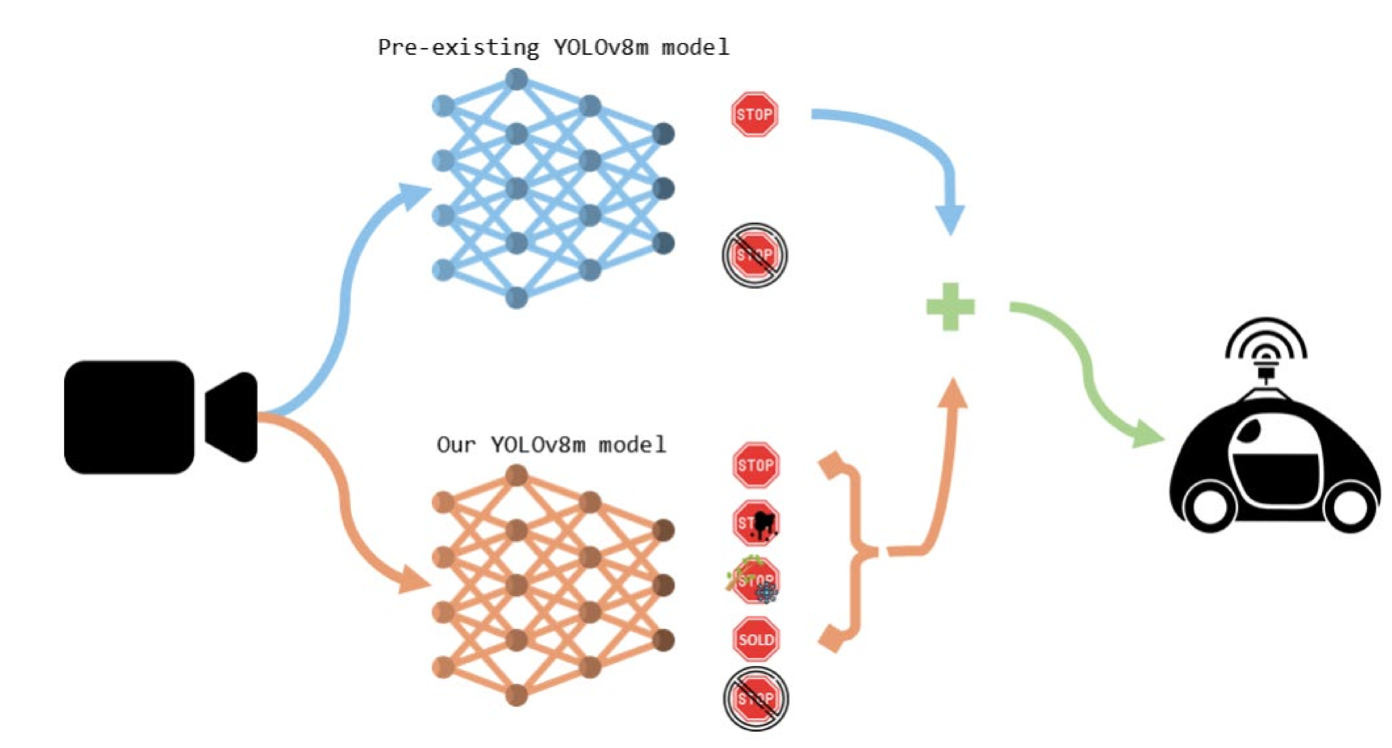
RESULTS



[Figure 3: Stop sign detection of Haar Cascade and YOLOv8m models.] This side-by-side comparison shows that the YOLOv8m model detects of all 4 types of stop signs better than Haar.



[Figure 4: Confusion matrix of YOLOv8m model.]



[Figure 5: Future Improvements to Stop Sign Detection.]

DISCUSSION

As seen from the results, our model successfully distinguishes between different types of stop-signs. We had a few issues with making this model work in realistic scenarios due to certain biases in the dataset as well as some of the design features of the model. Ultimately, we went through about 5 different versions of our dataset; at first our model was detecting random things, such as ceiling squares, as a 'vandalized stop-sign' due to our examples mostly containing close up images of stop signs. Another important issue, that was then taken into account, was that the distance at which the stop-sign appears in the image plays a big role in real-time accuracy. To address this, we added images of stop-signs within realistic ranges. In addition to continually improving the dataset, we also experimented with the model design, as we trained a variety of structures ranging from 3.2M to 68.2M parameters, as well as experimentation with image resolution, batch size and number of training epochs. We decided YOLOv8m (25.9M) was the best sized model for the detection speed required in our application.

The model could be improved and scaled up when it comes to distinguishing between fake stop-signs and vandalized stop-signs. Since this project is unique in the sense that there are not many public models or datasets that look at other use cases, such as fake stop-signs, one of our main focuses was to collect/create this inclusive dataset. The dataset is our largest contribution thus far and is publicly available[3] for future research. With that being said, the current work completed on this project will be passed onto other students for future analysis and work for the IGVC team to further improve upon and utilize for LTU's autonomous vehicle.

References

1. http://www.igvc.org/2023rules.pdf
2. http://www.igvc.org/design/2019/36.pdf
3. https://universe.roboflow.com/stopsign detection/stopsigndetection