Improving Road Awareness with a Real-Time Co-Pilot by Harnessing the Power of Dashcams

Adrian John
CSC 752 - Computer Vision
Department of Computer Science, University of South Dakota
Supervised by Dr. Lina Chato, Fall 2024

***Abstract*—Every year millions of motor vehicular accidents are reported in the United states alone, many resulting in death and other life changing injuries. This paper explores a cost-effective solution to reduce car accidents by enhancing driver awareness through a real-time co-pilot system powered by machine learning. The goal is to harness the power of the car's dashcam video feed to give drivers a second pair of eyes, alerting them about crucial traffic elements such as signs, crosswalks, and speed limits. This approach aims to make advanced driver-assistance systems accessible to the general public, helping to reduce accident rates and fatalities. For this project, a machine learning algorithm called YOLO was used for detection and classification where we saw an overall precision of 83.7 percent.**

***Index Terms*—Dashcams, Driver Awareness, Machine Learning, Real-Time Detection, Traffic Signs, Crosswalk Detection, Speed Limit, Computer Vision**

I. Introduction

According to the National Highway Traffic Safety Administration (NHTSA), in the year 2022, motor vehicle accidents in the United States led to approximately 42,795 fatalities. [1]. This represents the urgent need for advanced road safety solutions. Factors contributing to these accidents include speeding, distracted driving, and impaired driving [2]. Lets be honest, we've all missed that one stop sign just because we were focused on some other aspect on the road, or even looking at google maps to ensure we are heading in the right direction.

Automotive companies like Tesla have introduced advanced driver-assistance systems that employ machine learning and computer vision techniques to reduce accident rates by alerting drivers of traffic signals, potential hazards, pedestrians and many more [3]. However, these systems remain expensive and typically are found only in luxury or high-end vehicles, making them inaccessible to many drivers. This project seeks to address this gap by harnessing the power of the dash-cam feed available most modern vehicles. Vehicles without a dash-cam can easily install once at an affordable cost.

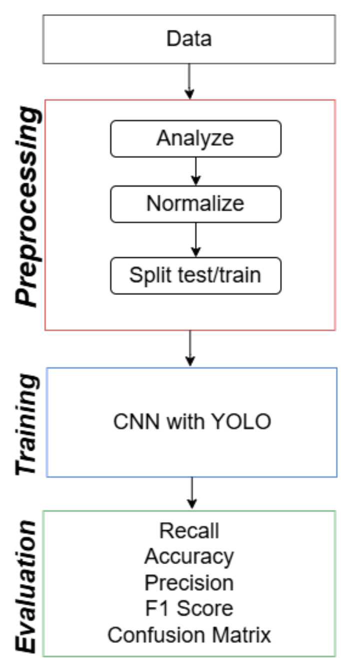
II. Objective

The primary goal of this project is to improve driver situational awareness at a low cost by providing real-time alerts about nearby traffic signs, crosswalks, and speed limits using dash-cam footage and machine learning to reduce motor vehicular accidents.

III. Related Work

Tesla's self driving system is a concrete example of autonomous vehicle technology, employing deep learning and computer vision techniques. The system relies on neural networks for object detection to control components like steering wheel, accelerator and breaks. It also provides real time updates to the display screen of potential dangers on the road. These neural networks are trained on millions of video clips collected from Tesla's extensive fleet replacing older methods using explicit programming (e.g., over 300,000 lines of C++ code). The latest version of Tesla's auto detection system is called FSD Beta v12, only comes at a premium cost. A Tesla can cost anywhere from 45000 to 120000 United States Dollars, making it unaffordable by the major.

IV. Methodology



IV.a Data Collection

This project utilizes the ”Road Sign Detection” dataset from Kaggle [1] to train the model which contains 877 annotated images in png format. Each sample is classified as ”stop sign", "speed limit", "crosswalk" and "traffic light" and the annotation is in the form of an XML file format.

IV.b Data Preparation

- Extraction and normalization was performed on the XML data for bounding boxes coordinates from the XML files
- The dataset is split into training and testing sets

IV.c Model Development

YOLO is a pre-trained model renowned for its efficiency and accuracy in real-time object detection that is capable of classifying multiple classes simultaneously. YOLO can classify over 80 classes which varies from cars, animals and even food [5].

The model was initialized with the YOLO weights and it trained on the 877 images. No adjustments were made on the architecture of the YOLO model besides the weights.

IV.d System Integration

To simulate real-time video analysis, we utilized a high-resolution YouTube video featuring dash-cam footage recorded in Los Angeles in California at 60 frames per second (FPS). Python was employed to process the video frame by frame, ensuring each frame was iteratively passed through our trained model for object detection and analysis. This approach allowed us to evaluate the model's performance in a dynamic, real-world scenario. **For example:**

```
model = YOLO('model.pt')
camera_feed = cv2.VideoCapture(0)

while True:
    results = model(frame)
    play_audio_based_on_result(model(frame))
```

V. Experiments and Results

The example below shows a comparison between YOLO default weights and the new wights in attempting to make a prediction.

Fig. 1. Example

The model was evaluated on the test dataset which contained 176 images, containing a total of 236 instances across four classes: traffic light, stop sign, speed limit, and crosswalk. The evaluation metrics include Precision (P), which measures the accuracy of the model's positive detections, and Recall (R), which measures the model's ability to detect all actual instances of each class. 2.

Class	Images	Instances	Precision	Recall
all	176	236	0.837	0.851
trafficlight	18	25	0.648	0.64
stop	13	13	0.77	0.923
speedlimit	137	157	0.958	0.994
crosswalk	36	41	0.972	0.849

Fig. 2. Evaluation Results on Test Dataset

Overall, the model achieved a Precision of 0.837 and a Recall of 0.851. This indicates a balanced performance, where the model correctly identifies a high proportion of instances with relatively few false positives.

Traffic Light: With a Precision of 0.648 and Recall of 0.64, the model performed lower on this class, suggesting it may struggle to accurately identify and capture traffic lights.

Stop Sign: Achieving 0.77 Precision and a high Recall of 0.923, the model successfully detects nearly all instances of stop signs, though some false positives reduce the precision slightly.

Speed Limit: This class sees excellent performance with Precision of 0.958 and Recall of 0.994, indicating the model consistently and accurately identifies speed limit signs with minimal error.

Crosswalk: With Precision of 0.972 and Recall of 0.849, the model performs well in detecting crosswalks, demonstrating strong accuracy with only a slight decrease in recall.

These results suggest that while the model performs well across most classes, traffic light detection remains an area for potential improvement. Overall, the high Precision and Recall for stop signs, speed limits, and crosswalks indicate the model's effectiveness in detecting these features in the test dataset. The result of this is more clearly shown in the confusion matrix 3. This suggest the model is trying predicting traffic signs when there are no traffic signs.

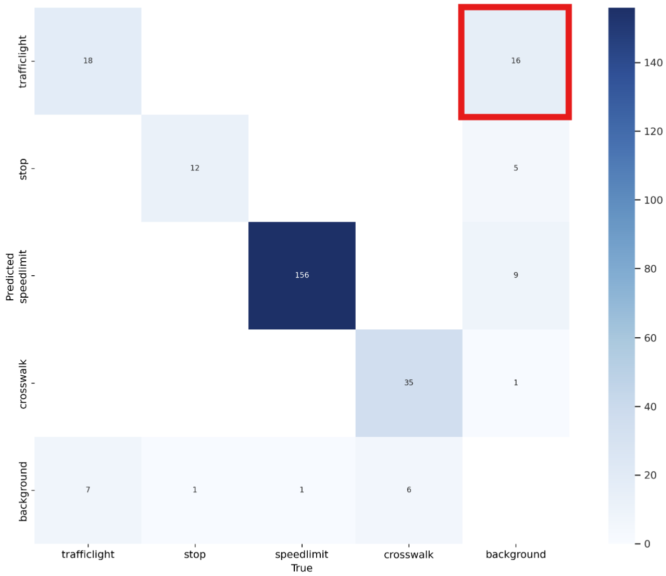


Fig. 3. Confusion Matrix

VI. Future works

VI.a Data Augmentation

To get a more robust and diverse dataset, we can use data augmentation techniques to artificially expand the dataset and improve the model's ability to generalize to unseen data. This includes operations such as rotations, resizing, flipping, cropping, adjusting brightness and contrast, adding noise, and applying various transformations like scaling or shearing. These techniques help simulate real-world variations and enhance the model's robustness in diverse conditions.

VI.b YOLO Optimization

Since no modifications was made to YOLO, we can possible enhance its performance by introducing modifications to its architecture. For instance, we could freeze the fully connected layers of YOLO and add add a custom set of fully connected layers. This additional layer can be fine tuned and experimented with, allowing the model to better adapt to the dateset

VII. Conclusion

In conclusion, this paper aims to implement a real-time copilot system leveraging machine learning and dash cam footage to enhance driver awareness and potentially reduce vehicular accidents. Through the use of dashcam video feeds, our solution provides an accessible and cost-effective approach to implementing advanced driver-assistance systems features. By detecting essential traffic elements like traffic signs, crosswalks, and speed limits, this system serves as a supplementary "second pair of eyes" for drivers, helping them stay more informed and aware of their surroundings.

The experimental results reveal that the model performs well on detecting stop signs, speed limits, and crosswalks with high precision and recall, though further optimization is necessary for accurate traffic light detection. Overall, this approach demonstrates the feasibility of creating a low-cost driver awareness system that could benefit the general public.

Acknowledgment

We thank Dr. Lina Chato for her guidance and support throughout this course and project.

References

[1] A. Mvd, "Road Sign Detection Dataset," *Kaggle*, 2020. [Online]. Available: https://www.kaggle.com/datasets/andrewmvd/road-sign-detection.

[2] National Highway Traffic Safety Administration (NHTSA), "Early Estimates of Motor Vehicle Traffic Fatalities for the First Half (January–June) of 2022," Traffic Safety Facts, 2023. [Online]. Available: https://www.nhtsa.gov.

[3] Insurance Institute for Highway Safety (IIHS), "Fatality Statistics," 2022. [Online]. Available: https://www.iihs.org/topics/fatality-statistics.

[4] Tesla, Inc., "Autopilot and Full Self-Driving Capability," Tesla, 2022. [Online]. Available: https://www.tesla.com/autopilot.

[5] P. J. Redmon, "YOLO: Real-Time Object Detection," 2018. [Online]. Available: https://pjreddie.com/darknet/yolo/.

[6] Ao, B., Ren, J. and Guo, C., "Impact of image corruptions on the reliability of traffic sign recognition using machine learning technique," in *Proceedings of the 2019 3rd International Conference on Compute and Data Analysis*, pp. 6–8, 2019. doi:10.1145/3314545.3314547.

[7] Kanchana, B. et al., "Computer Vision for Autonomous Driving," *2021 3rd International Conference on Advancements in Computing (ICAC)*, 2021. doi:10.1109/icac54203.2021.9671099.

[8] National Safety Council, "Motor Vehicle - Introduction," *Injury Facts*, 2024. Available at: https://injuryfacts.nsc.org/motor-vehicle/overview/introduction/ (Accessed: 20 September 2024).

[9] Tang, H. et al., "Deep Unsupervised Key Frame Extraction for Efficient Video Classification," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 19, no. 3, pp. 1–17, 2023. doi:10.1145/3571735.

[10] Tesla, Inc., "Vehicle Safety Report," Tesla. Available at: https://www.tesla.com/VehicleSafetyReport (Accessed: 20 September 2024).

[11] Viktor, M. and Honggui, H., "Traffic Sign and Vehicle Classification Based on Machine Learning," in *2020 2nd International Conference on Video, Signal and Image Processing*, pp. 35–41, 2020. doi:10.1145/3442705.3442711.

[12] Avatar for Fred Lambert Lambert, F.,"Tesla starts pushing dumb-down version of Full Self-Driving v13" https://electrek.co/(Accessed: 20 October 2024).