Chase Hall

Dr. Sarkar

CS 440-01 Drones

10 December 2025

Road Sign Detection with YOLOv11:
Object Detection for Drone-Based Mapping

Abstract

This project explores a use case of YOLOv11 for automatic road sign detection in street level imagery. A publicly available Roboflow dataset that contained four classes, which were crosswalk, speed limit sign, stop sign, and traffic light, was used to train and evaluate a YOLOv11n model in Google Colab. This dataset was exported in YOLO format, and the model was trained for 25 epochs at 640x640 resolution with a batch size of 16. On the validation set of 174 images and 232 labeled signs, the model achieved a mean precision of 0.968, mean recall of 0.943, mAP@0.5 of 0.957, and mAP@0.:0.95 of 0.811, indicating strong performance across the classes. The report analyzes these results, highlights the challenges of small or distant signs, and reflects on how a detector could be integrated into drone and mapping workflows for tasks like updating road sign inventories and supporting a smart city infrastructure.

Introduction

Road signs are a critical part of road safety and navigation. Human drivers rely on road signs for speed limits, crossing zones, and right of way rules. Modern driving assistance and autonomous systems also need to be able to interpret this visual information. Automating the road sign detection process is therefore an important computer vision problem with applications in intelligent transportation systems, infrastructure monitoring, and mapping.

In this project, I focused on object detection of road signs using the YOLOv11 model family. The goal was to detect and localize multiple types of signs in single images. Rather than building a model from scratch I use a publicly available annotated dataset found on Roboflow and a modern implementation of YOLO to explore how well a lightweight model can perform on a task like this.

Beyond evaluating detection performance, an additional goal is to connect this work back to drones and aerial mapping. Mini drones can capture imagery at low altitudes, while 3D mapping tools can place detections into geographic context. A road sign detector could then support automated map updates, asset management and safety inspections. This report describes the dataset, model, and training setup, analyzes the results, and reflects on how a system like this could be deployed in a drone-based mapping workflow

Methodology

I used a Roboflow universe dataset focused on road signs, exported in YOLO format. The dataset contains four object classes: crosswalk, speed limit, stop sign, and traffic light. The images depict urban street scenes with a variety of angles, distances, and lighting conditions. The Roboflow export supplies files that include train, val, and test splits, with each image paired with a YOLO label file containing one line per annotated sign.

The validation split, which I used for reporting final metrics, contains 174 images and 232 labeled road sign instances, as reported by the YOLOv11 validation summary. There were about 1-2 signs per image on average, with some containing none and some containing up to 5 signs. The dataset has several challenges common withing real world traffic scenes, where some signs are far away, some are partially covered, and others are affected by glare, shadows, or background clutter such as buildings, trees, and other vehicles. I manually checked several annotated images within the Roboflow dataset to obtain a better understanding of what the acutal dataset looked like.

For the model I used YOLOv11n the nano variant of the YOLOv11 family provided by the Ultralytics library. YOLOv11n is designed to be lightweight and fast, which makes it good for deployment on edge devices such as a small onboard computer on a drone. All experiments were run on Google Colab with T4 GPU.

The dataset was configured via a data.yaml file generated by Roboflow during exportation. This file specified the paths to the train, validation, and test folders and the four class names, Images were then resized to 640x640 during training. The configuration I went with was YOLOv11n for 25 epochs with a batch size of 16, a typical choice that balances convergence quality, GPU memory constraints, and practical training time in Colab

The choice of 25 epochs reflects a balance between learning capacity and overfitting risk on a dataset of this size. For small to medium dataset, YOLO models reach good performance within 20-30 epochs, after gains are minimal and risk of overfitting increases. The batch size of 16 was selected to fit comfortably in GPU memory for 640x640 images while still having a stable gradient estimate. Smaller batches would lead to noisier updates and slower training, while large batches would exceed the memory limits within Colab's environment.

During training, the Ultralytics logs reported standard metrics such as training and validation losses and intermediate mAP scores. The best model checkpoint was saved and reloaded for evaluation on validation set and generated qualitive prediction examples on test images.

Results & Discussion

Metric	Value	Interpretation
Mean Precision (mP)	0.968	About 97% of predicted boxes were correct
Mean Recall (mR)	0.943	About 94% of real signs were detected
mAP@0.5	0.957	High accuracy at loose IoU (0.5)
mAP@0.5:0.95	0.811	Good performance even under stricter IoU thresholds

On the validation set of 174 images with 232 labeled signs, the YOLOv11n model achieved strong overall performance. The mean precision of 0.968 shows that the predicted boxes mostly correspond to actual signs, with relatively few false positives. The mean recall of 0.943 shows that the model is able to find the majority of signs present within the images. The mAP@0.5 of 0.957 confirms that the detection and classification are highly accurate at a standard IoU threshold, while the stricter mAP@o.5:0.95 of 0.811 shows that localization quality is good even when more precise boundaries are required

Looking at per class metrics from the YOLO summary, speed limit and stop signs show the highest accuracy, with precision and recall values near 0.99-1.00 and mAP@0.5 close to 1.

Cross walks also perform well but have a slightly lower recall which means some crosswalks are missing. Traffic lights are the most challenging class with lower mAP@0.5:0.95 reflecting the difficulty of precisely localizing small bright objects in clutter scenes. This pattern fits with the dataset's characteristics: traffic lights tend to ben smaller in the image and be surrounded by poles wires, and buildings causing interference.

Qualitative analysis of test predictions shows that the model produces clean bounding boxes around most speed limits and stop signs. However, errors still occur in several situations like small or distance signs sometimes ignored, glare or lighting can cause misplacement of bounding box, and clutter where signs near visually similar objects increases.

Drone & Mapping Reflection

Although this project was done on static images, the motivation is to integrate detection into a drone-based mapping workflow. An idea would be a mini drone equipped with a camera that could fly along roads and campus paths, capturing images. A model like the trained YOLOv11n detector could run either on the drone or offline after the flight, automatically identifying locations of cross walks, speed limit signs, stop signs, and traffic lights.

When combined with mapping tools like Map Made Easy detections could be geotagged and overlaid onto a 2D or 3D map. This would allow for automatically maintaining a road sign inventory for a campus, checking for missing or damaged signs, or flagging of inconsistencies between physical and digital maps

However, there are limits. From higher altitudes, signs would occupy fewer pixels in each image, which directly hurts detection performance. Flight path planning, altitude, and camera resolution all become critical design choices. Running onboard a small drone requires careful model selection and hardware considerations, as larger YOLO variants may not meet real time constraints on limited hardware

Conclusion & Future Work

This project showed that a lightweight model like YOLOv11n can achieve strong performance on a multi class road sign detection task. The model reached high precision and recall, with mAP@0.5 and mAP@0.5:0.95 values indicating that both detection and localization are generally reliable. The best performance was on speed limit and stop signs, while traffic lights were still detected but contained more errors.

The results suggest that modern object detection architectures can serve as a core component in drone assisted mapping systems for road infrastructure. At the same time, the error analysis indicates that more diverse training data and better handling of small objects are necessary for real world deployment.

Future work could be expanding the dataset with more images of different heights angles and lightning; Implementing more road sign types into the data; experimenting with larger models or fine tuning the current model; and building a prototype that integrates detection GPS and mapping tools to create a live georeferenced road sign layer for a campus or urban area.

Works Cited

Hall, Chase. *Final_project.ipynb.* Google Colab notebook, 2025. Google Drive, https://colab.research.google.com/drive/1W15mOAieLFEE1uNIWY6Tizgj8ZkTfNHV#scrollTo=F1cWpdvbYn-I. Accessed 8 Dec. 2025.

Belaroussi, Faliha, et al. "Road Sign Detection in Images: A Case Study." *Proceedings of the International Conference on Machine Vision Applications (MVA)*, 2010. ResearchGate, https://www.researchgate.net/profile/Rachid-Belaroussi/publication/48418194_Road_Sign_Detection_in_Images_A_Case_Study/links/09e41508eb141a5690000000/Road-Sign-Detection-in-Images-A-Case-Study.pdf. Accessed 8 Dec. 2025.

Roboflow. *road-sign.* Roboflow Universe, https://universe.roboflow.com/new-workspace-vvjjt/road-sign. Accessed 8 Dec. 2025.

Ultralytics. "YOLOv11 Models – Ultralytics Documentation." Ultralytics Docs, https://docs.ultralytics.com/models/yolo11/. Accessed 8 Dec. 2025.