Lower-altitude and Last-mile Drone Obstacle Detection using YOLOv5

Vihaan Bhaduri | vihaan.bhaduri@gmail.com | Saratoga High School | 10th Grade

Introduction

Currently, there is a lot of focus in the AI industry on terrestrial Autonomous Vehicles (AV) but minimal attention on aerial AV using drones. On top of that, an Internet and Kaggle survey shows the major focus on drone obstacle detection using the birds-eye-view (Nadir's view) dataset at higher altitudes. This paper attempts to not only steer the focus on '*AV-in-the-sky*' (drones) but also to address the autonomy of drones flying at lower altitudes and navigating the last mile. This is essential for the residential and commercial delivery of goods like medicines, food, and emergency equipment. The first step to achieving that is successful object detection by drone cameras at lower altitudes and last-mile datasets, which is the primary focus of this paper.

Methodology

Building dataset: One of the biggest hurdles in this project was the lack of public low-altitude drone datasets that were labeled based on common obstacles (Table 1) that are most probable to occur near residential locations. Kaggle [3, 4] and other public dataset websites provided drone images that typically focused on a top-down perspective (bird's eye/nadir view) rather than an oblique or horizontal field of view, which wouldn't help train the model meant for lower altitude maneuverability that is essential for safe navigation in crowded regions. Hence, resorting to dataset creation from scratch, focusing on 6 different classes - buildings, houses, fences, poles, street signs, and trees - was the only option. Based on these goals, three locations were chosen for image acquisition: Brandywine Drive (residential), Saratoga High School (school), and downtown Los Gatos (commercial). A DJI Mavic Mini Drone was flown in these locations at varying ranges of altitudes from near-ground-level to aerial views, around 5-30 meters above the ground. The gimbal orientation varied between 10° - 90°, bringing different perspectives between horizontal to vertical.

Labeling: Images were labeled on Roboflow [1], a computer vision data curation and labeling site. 228 PNG images were uploaded in Roboflow after downloading them from the drone. Every minute object in the image was not labeled to save time; rather, the focus was on more prominent objects. The polygon tool was used to perform instance segmentation by hand.

Tree	Building	Fence	House	Pole	Street Sign
1757	503	239	248	456	113

Table 1: Non-uniform number of instance segmentations per class

Training: To train the model, a technique, called transfer learning, was used on an existing YOLOv5 model with pre-trained weights. YOLOv5 architecture was chosen, since it is fast and lightweight and is well suited for small drone form factors. As stated in PyTorch's article [2] , "*YOLOv5 is designed to be fast, accurate, and easy to use, making it an excellent choice for a wide range of object detection, instance segmentation, and image classification tasks.*" Initially, the model yielded poor detections, since it was trained on a slim dataset. Resorting to data augmentation to increase the size of the dataset greatly improved the models performance. As shown in Table 1, a 4-fold data augmentation synthetically tweaked the noise, brightness, exposure, and blur of the original images and increased the size of the training dataset. This, in turn, generalized the model during fine-tuning, which led to better predictions and confidence in the test set.

Noise	Brightness	Exposure	Blur
Up to 5%	-25% → 25%	-17% → 17%	Up to 1.5 pixels

Table 1: Parametric variations for data augmentation

What went well and what didn't: Manual labeling was a time-consuming and tedious process and data augmentation provided a quick way to scale up the size of the image dataset, also generalizing it by introducing variability, preventing overfitting. The dataset used a 3-fold augmentation in Roboflow across noise, exposure, brightness, and blur parameters. Augmentation deployed only on the instance segmentations did not improve the detection performance, but augmentation on the whole image performed better. The initial hypothesis that removing non-performant classes, like street signs and poles, would lead to better accuracy proved incorrect, as the prediction accuracy went down. Hence, the training was reverted to include all 6 classes, as shown in Table 2,3.

3. Results and Analysis

Although it was difficult to achieve a perfect model capable of performing well in all classes, the later models generally performed much better than the earlier ones. To emphasize the observation, two contrasting data are presented. Results from an earlier fine-tuned model (model 2) showed higher training accuracy compared to a later fine-tuned model (model 74), as shown in Table 2, 3 below. When the same fine-tuned models are used for inference on an unseen test set, a contrasting outcome was observed. Despite the higher training accuracy, model 2 performed poorly on the test set. It failed to detect many objects and had lower confidence due to overfitting. As shown in Table 3 (Fig. 3), model 74 performed much better, detecting objects with greater confidence on the test set, despite lower training accuracy than model 2.

Buildings	Fences	Houses	Poles	Street Signs	Trees
65%	56%	81%	41%	0%	48%

Table 2: Training accuracy with an initial model 2 (initial experiments)

Buildings	Fences	Houses	Poles	Street Signs	Trees
53%	60%	78%	30%	27%	59%

Table 3: Training accuracy with model 74 (later experiments)

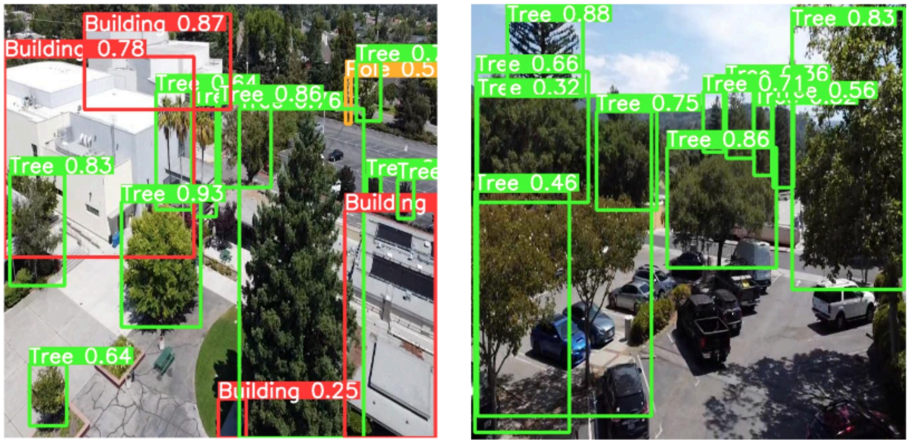

Fig 3: Object detection accuracy and confidence on the test set using model 74

Conclusion and Future Work

A successful low-altitude and last-mile obstacle detection model was built using drone images. Data augmentation, a catalyst for the rapid improvement in detection accuracy, led to an increase in prediction accuracy, as the final fine-tuned model from experiment 74 rarely misclassified. Additionally, this model was able to detect objects with high confidence, achieving the baseline project goal and creating a working base model for future enhancements. As a part of future work, I would like to increase the size of the dataset, take closer shots of slimmer objects, like street signs and poles, and use the "segment-anything" [5] tool from Meta to enhance labeling efficiency. I would also like to address the class imbalance (Table 1) and have a more uniform and balanced dataset across all the classes. Additionally, I would like to contribute to the open-source Kaggle datasets and publish my code on GitHub publicly to help others explore the relatively unexplored domain of drone AI and make "*AV-in-the-sky*" a reality.

References

[1]: https://app.roboflow.com/drone-obstacle-detection/drone-object-detection-yhpn6/deploy/23
[2]: https://pytorch.org/hub/ultralytics_yolov5/
[3]: https://www.kaggle.com/datasets/bulentsiyah/semantic-drone-dataset?rvi=1
[4]: https://www.kaggle.com/datasets/nunenuh/semantic-drone
[5]: https://github.com/facebookresearch/segment-anything