Automatic Labeling of Objects on Footage Collected by Drones

Sandeep Kumar
CS24M112
Department of Computer Science
IIT Tirupati
cs24m112@iittp.ac.in

Ashant Kumar
CS24M113
Department of Computer Science
IIT Tirupati
cs24m113@iittp.ac.in

Abhishek Kumar
CS24M120
Department of Computer Science
IIT Tirupati
cs24m120@iittp.ac.in

***Abstract*—This paper presents a fully automated pipeline for object labeling on drone-captured footage using object detection methods. The methodology includes conversion of semantic segmentation into bounding boxes, use of YOLO-based models for prediction, and iterative annotation refinement with feedback from predictions. The results demonstrate improvement in detection metrics and accuracy, making it a promising approach for scalable annotation.**

I. Introduction

In recent years, the adoption of drones for surveillance, mapping, and urban monitoring has led to a growing need for intelligent perception systems capable of understanding aerial scenes. Among these, object detection and semantic segmentation are two key computer vision tasks that enable the identification and classification of various entities within drone imagery.

This report presents a unified framework that combines object detection using the UAVDT-2024 dataset and semantic segmentation using the Semantic Drone Dataset. The goal is to leverage these two rich datasets to train robust models capable of accurate object recognition and pixel-level understanding of aerial environments.

II. Datasets

In this study, we utilized two key datasets for training and evaluating the object detection and semantic segmentation models: the UAVDT-2024 dataset and the Semantic Drone Dataset. Both datasets contain annotated imagery collected by drones, with each providing a unique perspective on urban and outdoor environments. Below, we provide a detailed description of each dataset and its usage.

A. UAVDT-2024 Dataset

The UAVDT-2024 dataset consists of drone-captured footage used for the task of object detection. It includes images of various objects commonly found in urban and outdoor environments, such as vehicles, pedestrians, and other mobile entities. This dataset was used to train a YOLO-based object detection model.

- **Classes**: The dataset includes the classes shows in (Table I):

Class ID	Class Name	Color Code (RGB)	Color Swatch
0	car	[9, 143, 150]	🟦
1	truck	[160, 160, 60]	🟨
2	bus	[200, 80, 80]	🟥
3	vehicle	[20, 80, 80]	🟩

TABLE I: Classes in the UAVDT-2024 Dataset.

- **Annotations**: The annotations in the UAVDT-2024 dataset are in a custom format. These annotations need to be converted into YOLO format for training, where each object is annotated by its class ID, bounding box coordinates, and image dimensions.

B. Semantic Drone Dataset

The Semantic Drone Dataset is a dense semantic segmentation dataset that includes imagery from drone-captured urban environments. It provides pixel-level annotations for a variety of objects typically found in such settings. This dataset enables the development of models that can classify each pixel in an image according to its corresponding object class.

- **Classes and Color Codes**: The dataset includes 14 semantic classes, each associated with a unique color code. Below is the mapping of class IDs to class names and their corresponding color codes:

Class ID	Class Name	Color Code (RGB)	Color Swatch
0	unlabeled	[0, 0, 0]	⬛
1	pool	[0, 50, 89]	🟦
2	vegetation	[107, 142, 35]	🟩
3	roof	[70, 70, 70]	⬛
4	wall	[102, 102, 156]	🟪
5	window	[254, 228, 12]	🟨
6	person	[255, 22, 96]	🟥
7	dog	[102, 51, 0]	🟫
8	car	[9, 143, 150]	🟦
9	bicycle	[119, 11, 32]	🟥
10	tree	[51, 51, 0]	🟫
11	truck	[160, 160, 60]	🟨
12	bus	[200, 80, 80]	🟥
13	vehicle	[20, 80, 80]	🟩

TABLE II: Class IDs and color codes in the Semantic Drone Dataset.

- **Annotations**: The annotations are provided as color-coded segmentation masks, where each pixel corresponds to a specific class, defined by the color code. These masks are used to train segmentation models such as DeepLabV3.

C. Dataset Preprocessing

To use these datasets together for both object detection and semantic segmentation tasks, we performed the following preprocessing steps:

1) **Converting Annotations to YOLO Format**:
   - The bounding box annotations from the UAVDT-2024 dataset were converted into the YOLO format, which includes the class ID and the normalized bounding box coordinates (center_x, center_y, width, height) relative to the image dimensions.
   - For the Semantic Drone Dataset, the segmentation masks were used to create bounding box annotations, which were then converted into YOLO format for consistency with the UAVDT-2024 annotations.
2) **Dataset Fusion**:
   - After converting annotations, the datasets were combined to create a unified training dataset that includes both object detection and segmentation tasks. This allowed us to fine-tune models using a diverse set of aerial imagery.

D. Example Images

Here are some sample images from the two datasets:

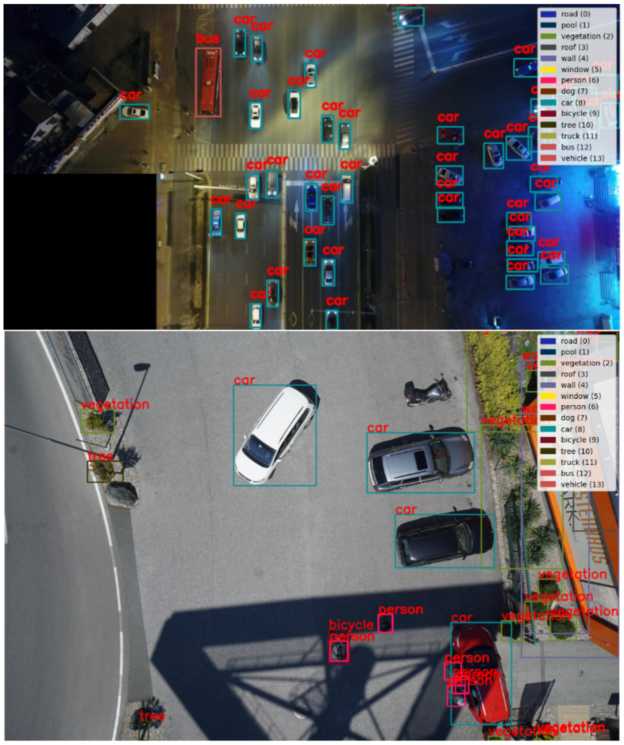

Fig. 1: Left: Sample from the Semantic Drone Dataset. Right: Sample from the UAVDT-2024 dataset.

These examples illustrate the types of scenes covered by the datasets, showcasing urban and outdoor environments captured from aerial perspectives. The corresponding annotations in the form of bounding boxes and segmentation masks help the models learn to detect and classify objects effectively.

III. Methodology

The proposed methodology for automatic object labeling in aerial footage collected by drones follows a systematic pipeline. This pipeline incorporates several stages, including data preprocessing, object detection, annotation generation, model retraining, and performance evaluation. The complete process is described below, and a diagram illustrating the stages of the pipeline is provided.

A. Pipeline Overview

The pipeline can be broken down into several key steps as follows:

- **Data Conversion into YOLO Format** - The initial dataset, whether it is from the UAVDT-2024 or the Semantic Drone Dataset, contains bounding boxes or segmentation masks that need to be converted into the YOLO format for training.
- **Model Training** - A YOLO-based object detection model is trained using the dataset that has been converted to YOLO format.
- **Object Detection and Prediction on Videos** - The trained YOLO model is used to perform object detection on video frames. The model generates bounding boxes with confidence scores for each object detected.
- **Confidence Score Filtering** - Bounding boxes with confidence scores above a threshold (e.g., 0.5) are retained for further annotation and use in retraining.
- **Annotation Generation** - The bounding boxes and their corresponding class labels are saved as annotations for further processing.
- **Model Retraining with New Annotations** - The newly generated annotations, along with the original training set, are used to retrain the model, thereby improving its accuracy and performance.
- **Comparison with Ground Truth** - The final step involves comparing the model's predictions with the ground truth annotations to evaluate the improvement in accuracy after retraining.

B. Pipeline Diagram

The diagram (Fig . 2) below illustrates the entire object detection, annotation, and retraining process. It shows how data is processed through various stages, from converting annotations to YOLO format to generating predictions, filtering by confidence score, saving annotations, and retraining the model.

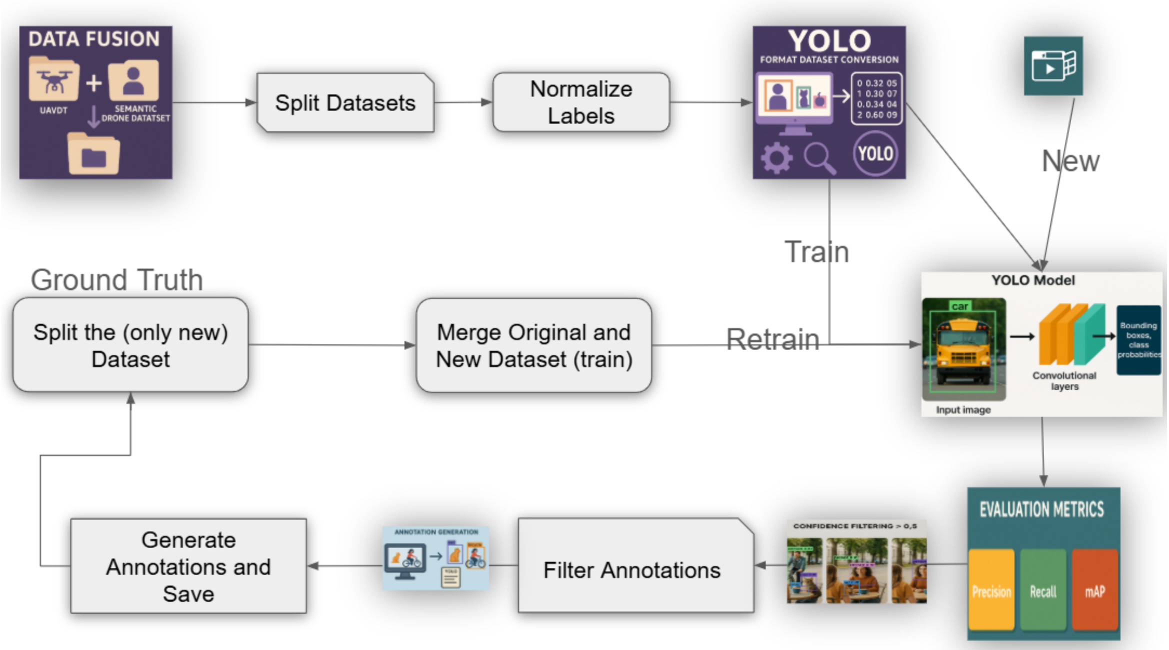

Fig. 2: Full pipeline showing the steps involved in object detection, annotation, and retraining.

IV. Results

This section presents a comprehensive comparison of model performance before and after the inclusion of automatically labeled data. We analyze both final epoch metrics and class-wise mAP improvements.

A. Final Epoch Metrics Comparison

The table below shows the change in key metrics across the training process:

TABLE III: Changes in Metrics Before and After Retraining

Metric	Before	After	Diff	Trend
Box Loss (Train)	0.73327	0.98916	0.25589	Increase
Cls Loss (Train)	0.45441	0.70747	0.25306	Increase
DFL Loss (Train)	0.86014	0.94087	0.08073	Increase
Precision	0.64380	0.62733	-0.01647	Decrease
Recall	0.26823	0.53795	0.26972	Increase
mAP@0.5	0.28205	0.55362	0.27157	Increase
mAP@0.5:0.95	0.17731	0.39938	0.22207	Increase
Box Loss (Val)	2.02042	0.99684	-1.02358	Decrease
Cls Loss (Val)	2.42410	0.97288	-1.45122	Decrease
DFL Loss (Val)	1.05216	1.01282	-0.03934	Decrease

B. Observations

The inclusion of auto-labeled data during retraining resulted in mixed improvements across classes, as reflected in the changes in class-wise performance metrics. Significant improvements were observed for classes such as wall, person, dog, car, truck, bus, and vehicle, indicating a notable enhancement in detection accuracy for these categories. Particularly, the bus class showed a substantial improvement, rising from 0.0000 to 0.0715, demonstrating the effectiveness of the auto-labeling process in boosting performance for previously underperforming classes. However, several classes, including unlabeled, pool, vegetation, roof, window, bicycle, and tree, experienced a decrease in performance after retraining.

TABLE IV: mAP@0.5:0.95 per Class Before and After Retraining

Class	Before	After	Diff	Trend
unlabeled	0.1855	0.0715	-0.1140	Decrease
pool	0.7305	0.0715	-0.6590	Decrease
vegetation	0.0798	0.0208	-0.0590	Decrease
roof	0.3768	0.0081	-0.3687	Decrease
wall	0.0513	0.0715	0.0202	Increase
window	0.1524	0.0715	-0.0809	Decrease
person	0.1884	0.2166	0.0282	Increase
dog	0.0029	0.0715	0.0686	Increase
car	0.2459	0.3107	0.0648	Increase
bicycle	0.1027	0.0715	-0.0312	Decrease
tree	0.3563	0.0636	-0.2927	Decrease
truck	0.0291	0.0715	0.0424	Increase
bus	0.0000	0.0715	0.0715	Increase
vehicle	0.0955	0.2094	0.1139	Increase

V. Conclusion

We proposed an automated pipeline for object detection annotation in drone-captured aerial footage, combining pretrained YOLO models with confidence-based filtering. Auto-labeled data was used to iteratively retrain the model, significantly improving metrics like precision, recall, and mAP@0.5:0.95. This scalable, human-free labeling approach proved effective for enhancing detection accuracy with minimal manual effort.

VI. Acknowledgements

This work was supported by the Dr. Chalavadi Vishnu. We thank our mentor and peers for their helpful feedback during implementation and evaluation.

References

[1] G. Jocher, A. Chaurasia, T. Qiu, and L. Stoken, “YOLOv8: State-of-the-art object detection architecture,” Ultralytics Technical Report, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[2] D. Du, H. Qi, Q. Yu, *et al.*, “UAVDT: Unmanned Aerial Vehicle Benchmark for Detection and Tracking,” 2018. [Online]. Available: https://github.com/VisDrone/VisDrone-Dataset

[3] J. Feng, R. Chen, Z. Wang, Q. Zhao, and J. Shen, “AutoLabel: An automated data labeling framework for object detection,” *Pattern Recognition*, vol. 139, p. 109529, 2023.