COMP9444 Project Summary

Object Detection for Autonomous Drones

z5521416 Mengfei Li

z5470782 Yansong Zhu

z5540457 Zhuoyu Wang

z5452207 Zhaoyu Yan

z5319476 Jiayang Jiang

I. Introduction

This project aims to develop an advanced object detection system for autonomous drones to address specific challenges faced by drones during task execution. Object detection is a fundamental task in computer vision, essential for applications such as autonomous driving, surveillance, and environmental monitoring. However, drones face unique challenges compared to traditional object detection systems, including severe changes in perspective, frequent occlusions, and varying object scales due to altitude and movement. These factors significantly impact detection accuracy and reliability, revealing shortcomings in existing models.

In this project, we constructed a robust object detection system using deep learning, capable of handling these challenges. By training on the VisDrone dataset, our system demonstrated high accuracy, precision, and recall in real-world scenarios, ensuring reliable detection performance. The contribution of this project lies in developing a powerful detection framework that provides better adaptability in dynamic environmental conditions, supporting applications such as surveillance, traffic management, search and rescue, and disaster response. This approach marks an important step in improving the adaptability and reliability of object detection for autonomous drones.

II. Related Work

Research on drone-based object detection mainly focuses on solving key challenges such as fast and efficient detection. Key contributions include the VisDrone dataset and object detection models such as YOLO, Faster R-CNN, SSD, DETR, and Swin, which have laid the foundation for detecting objects of various scales and aspect ratios.

VisDrone-DET2018 (Zhu et al., 2018)[1] introduced a benchmark dataset for drone object detection, focusing on multi-scale challenges. While it laid the foundation for models like YOLO and Faster R-CNN, its limitations in occlusion and scale diversity hinder broader applicability. YOLOv3 (Redmon & Farhadi, 2018)[2] optimized real-time detection with multi-scale features, making it ideal for high-speed applications such as drone detection. However, challenges remain in detecting small objects and occlusion in complex aerial scenes. SSD (Liu et al., 2016)[3] provides a fast, single-network solution with a balanced speed-accuracy trade-off but is less precise for small objects and dense scenes. To improve benchmarking, VisDrone-DET2019 (Du et al., 2019)[4] introduced more complex images, although it still requires refinement for environmental factors such as weather and altitude. DETR (Carion et al., 2020)[5] proposed a transformer-based, end-to-end detection framework that significantly improves accuracy, especially for long-tail data. The SWIN Transformer (Liu et al., 2021)[6] introduced a hierarchical design with a shifted window approach, improving efficiency and outperforming traditional methods in object detection and semantic segmentation tasks. These efforts highlight ongoing challenges in drone-based object detection, emphasizing the need for advanced models and diverse datasets to overcome issues like occlusion, scale variation, and small object detection.

YOLO focuses on speed, making it suitable for real-time detection; however, it struggles with detecting distant and small objects. SSD offers a balance between speed and accuracy but performs poorly in dense scenes. Faster R-CNN, although highly accurate, is too slow, limiting its applicability in real-time scenarios. DETR improves accuracy through its Transformer-based framework, particularly for long-tail data, but suffers from slower processing speeds, making it unsuitable for real-time applications. The SWIN Transformer utilizes a hierarchical design and shifted window approach to enhance efficiency, outperforming traditional methods; however, it still faces challenges with large-scale images and fast-changing scenes.

This project builds on these studies by enhancing object detection accuracy and efficiency for drones through the updated VisDrone dataset, addressing issues such as occlusion, scale variation, and small object detection.

III. Methods

We adopted three classical object detection models: YOLO, SSD, and Faster R-CNN.

YOLOv8 is a one-stage object detection model that strikes a good balance between speed and accuracy, making it ideal for real-time detection tasks. The SSD model uses VGG16 as the backbone network and a multi-scale detection strategy to enhance detection capabilities. Faster R-CNN is a two-stage detection model that combines Region Proposal Networks (RPN) and Region of Interest (ROI) pooling to achieve high precision in object classification and localization, making it suitable for tasks that require high precision.

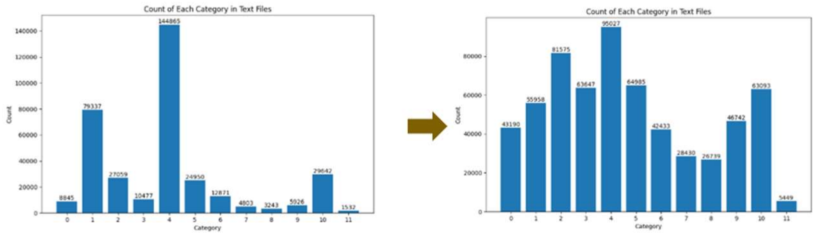
These three models were chosen based on a comprehensive assessment of task requirements, hardware constraints, and model performance. YOLOv8, with its fast detection capabilities and balanced accuracy, is well-suited for real-time object detection in drone applications. The SSD model achieves a good trade-off between speed and accuracy but struggles with small object detection. Faster R-CNN provides excellent localization precision but is slower, making it more suitable for tasks with higher precision requirements. Although DETR and Swin perform well in complex environments, their high computational cost and long training times make them less suitable for low-latency real-time detection tasks. Therefore, YOLO, SSD, and Faster R-CNN were selected for experimentation after considering these factors.

To accelerate training and improve detection performance, all models were fine-tuned using pretrained weights on the VisDrone dataset. YOLOv8 achieved a good balance between model complexity and hardware constraints by using pretrained weights. SSD utilized pretrained weights from ImageNet, with Xavier initialization applied to the newly added detection layers to ensure stable training. Faster R-CNN loaded pretrained ResNet weights and further optimized the model through a multi-task loss function.

IV. Experimental Setup

The VisDrone dataset was used for evaluation, which is widely applied in drone object detection tasks. The dataset contains two main categories of files: images/ (e.g., 001.jpg, 002.jpg) and annotations/ (e.g., 001.txt, 002.txt). The dataset consists of 8599 images, of which 6471 are used for training, 548 for validation, and 1580 for testing. The dataset includes 12 object categories, including ignored regions (0), pedestrians (1), people (2), bicycles (3), cars (4), vans (5), trucks (6), tricycles (7), awning-tricycles (8), buses (9), motorcycles (10), and others (11). However, the ignored region and "others" categories are excluded during evaluation. The dataset is challenging due to class imbalance, with some categories (e.g., pedestrians and cars) being dominant, while others (e.g., tricycles and awning-tricycles) have fewer instances.

To address class imbalance in the dataset, we applied offline augmentation techniques. First, we reduced the data for dominant categories (such as pedestrians and cars) to lower the training overhead. Then, we used the Mosaic augmentation strategy to expand the dataset for less frequent categories (such as bicycles, trucks, tricycles, awning-tricycles, and buses). The Mosaic strategy combines four images into one, randomly selecting from the dataset to generate over 50,000 possible combinations, resulting in 1,020 new augmented samples. These preprocessing steps increased dataset diversity and improved the class distribution.



Initial dataset
(Before-6471 images)

Processed dataset
(After-5962 images)

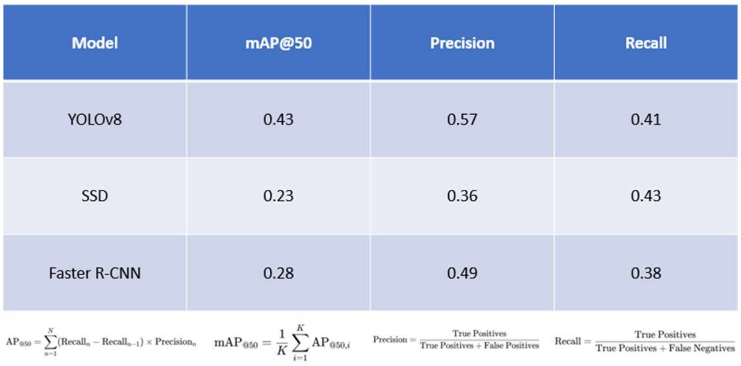
Figure 1

We used mAP@50 (mean Average Precision at IoU threshold 0.5) as the primary evaluation metric to calculate the average precision across all object categories. mAP@50 provides a comprehensive assessment of detection accuracy. Additionally, precision, recall were used to analyze false positives and missed detections, allowing a complete evaluation of model performance.

The models were optimized based on their characteristics. SSD used the Adam optimizer with learning rate decay (0.001), employing multi-scale localization and classification losses to improve detection accuracy, along with random cropping, horizontal flipping, and color jitter for enhanced robustness. YOLOv8 utilized the Adam optimizer with a cosine annealing scheduler, leveraging Mosaic augmentation, random scaling, flipping, and brightness/contrast adjustments, with a batch size of 16. Faster R-CNN used the SGD optimizer (learning rate 0.005) with classification and bounding box regression losses for multitask training, with ResNet as the backbone and data augmentations like cropping, flipping, and color adjustments.

V. Results

We evaluated three object detection models—YOLOv8, SSD, and Faster R-CNN—on the VisDrone dataset. YOLOv8 achieved the highest mAP@50 of 0.43 and recall of 0.41, making it ideal for real-time tasks with a good balance of precision and recall. SSD had a precision of 0.36 and a recall of 0.43, but its lower mAP@50 of 0.23 resulted in more missed detections. Faster R-CNN performed moderately with mAP@50 of 0.28, precision of 0.49, and recall of 0.38, showing a trade-off between accuracy and speed, suitable for complex scenes but slower in inference.



Model	mAP@50	Precision	Recall
YOLOv8	0.43	0.57	0.41
SSD	0.23	0.36	0.43
Faster R-CNN	0.28	0.49	0.38

$\mathrm{AP}_{@50} = \sum_{n=1}^{N} (\mathrm{Recall}_n - \mathrm{Recall}_{n-1}) \times \mathrm{Precision}_n$ $\quad \mathrm{mAP}_{@50} = \frac{1}{K} \sum_{i=1}^{K} \mathrm{AP}_{@50,i}$ $\quad \mathrm{Precision} = \frac{\mathrm{True\ Positives}}{\mathrm{True\ Positives} + \mathrm{False\ Positives}}$ $\quad \mathrm{Recall} = \frac{\mathrm{True\ Positives}}{\mathrm{True\ Positives} + \mathrm{False\ Negatives}}$

Figure 2

The results show that YOLOv8 offers the best balance of precision and recall, making it suitable for real-time detection tasks. While SSD had higher precision, its lower recall led to more missed detections, especially for small objects. Faster R-CNN, despite being more accurate in complex scenes, is slower and not ideal for real-time applications.

For real-world applications, YOLOv8 stands out due to its balance of speed and accuracy, making it suitable for autonomous drones or surveillance systems. SSD is faster but misses more detections, and Faster R-CNN, though accurate, has slow inference, limiting its real-time use.

Compared to other solutions and related research, YOLOv8 performs better in terms of both speed and accuracy, making it ideal for dynamic, real-time environments. Its performance is better than SSD's in detection coverage and faster than Faster R-CNN, making it the most practical choice for many applications.

VI. Conclusions

We focused on improving object detection using the VisDrone dataset, primarily addressing data imbalance by filtering occluded data and using Mosaic to augment underrepresented categories, thus balancing the dataset and improving the overall model performance.

The key strength of the solution lies in the selection of YOLOv8, which offers a good balance between speed and accuracy, making it suitable for real-time applications. The ensemble of SSD and YOLOv8 optimizes detection results, hyperparameter tuning ensures optimal learning, and the lightweight Faster R-CNN improves inference speed without sacrificing accuracy.

However, YOLOv8's performance still has room for improvement, especially in complex scenes with false positives. Data augmentation could also be further expanded to enhance generalization.

To further improve overall performance, we propose several strategies. First, data augmentation can be enhanced using techniques like rotation and color jitter to improve model generalization. Second, model ensemble can be used, with SSD refining YOLOv8's output to reduce false detections. Additionally, hyperparameter tuning, such as adjusting the learning rate, can further increase accuracy. Finally, developing a lightweight version of Faster R-CNN can improve inference speed while minimizing the loss of accuracy.

Reference

[1]. Zhu, P. *et al.* (1970a) *Visdrone-DET2018: The Vision meets drone object detection in image challenge results*, *SpringerLink*. Available at: https://link.springer.com/chapter/10.1007/978-3-030-11021-5_27 (Accessed: 16 November 2024).

[2]. Redmon, J. and Farhadi, A. (2018) *Yolov3: An incremental improvement*, *arXiv.org*. Available at: https://arxiv.org/abs/1804.02767 (Accessed: 16 November 2024).

[3]. Liu, W. *et al.* (2016) *SSD: Single shot multibox detector*, *arXiv.org*. Available at: https://arxiv.org/abs/1512.02325 (Accessed: 16 November 2024).

[4]. Zhu, P. *et al.* (2021) *Detection and tracking meet Drones Challenge*, *arXiv.org*. Available at: https://arxiv.org/abs/2001.06303 (Accessed: 16 November 2024).

[5]. Pu, Y. *et al.* (2023) *Rank-detr for high quality object detection*, *arXiv.org*. Available at: https://arxiv.org/abs/2310.08854 (Accessed: 16 November 2024).

[6]. Liu, Z. *et al.* (2021) *Swin Transformer: Hierarchical vision transformer using shifted windows*, *arXiv.org*. Available at: https://arxiv.org/abs/2103.14030 (Accessed: 16 November 2024).