

Improving YOLOv5 for Drone and UAV detection

Shahar Shpitaler, Alon Bar Koter, Noa Gerson-Golan, Andrei Plotkin

August 31, 2024

Abstract

This project addresses the challenge of detecting and classifying Unmanned Aerial Vehicles (UAVs) and Drones using the YOLOv5 object detection model. As UAVs become increasingly prevalent in various sectors, the need for effective detection systems has grown. We have enhanced YOLOv5 by incorporating advanced neural network components, including C3TR, GhostConv, and Focus layers, and by fine-tuning hyperparameters to optimize performance. Our experimental results indicate that while the baseline YOLOv5 model provides high accuracy, integrating enhancements such as GhostConv and Focus layers achieves a favorable balance between detection performance and computational efficiency. The trade-offs between accuracy, model complexity, and inference speed are discussed, offering insights into selecting an optimal model for practical deployment. This work contributes to the ongoing advancement of UAV detection technologies and provides a foundation for future research in this area. You can find the full code and details in our GitHub repository.

1 Introduction

The arrival of aerial technology has led to a significant increase in the use of UAVs and Drones across various sectors. By 2024, the market for these aerial devices has burgeoned to a valuation of 30.2 billion USD, with more than 5.42 million units deployed worldwide [1]. Their applications are diverse, encompassing delivery systems, security and surveillance, agricultural monitoring, photography, videography, mapping, data transmission, and military operations.

Despite their utility, the proliferation of UAVs and Drones raises critical issues pertaining to airspace safety, privacy, and security [3]. The potential for their misuse in criminal activities, such as smuggling, unauthorized surveillance, and interference with commercial air traffic, underscores the urgent need for effective detection and classification systems [4].

In this context, our project is dedicated to enhancing the YOLOv5 object detection model to accurately identify and differentiate between Drones and UAVs. YOLOv5, known for its real-time processing capabilities, serves as an excellent foundation for developing a system that can operate efficiently in dynamic and complex environments where these aerial devices are commonly encountered.

The objectives of our project are multifaceted. Firstly, we aim to train a custom YOLOv5 model to establish a robust baseline for the performance of Drone and UAV detection. Secondly, we seek to augment the architecture of YOLOv5 to improve inference speed and accuracy. This involves the incorporation of advanced neural network components such as C3TR (Transformer Enhanced C3), GhostConv, and Focus layers. Furthermore, we intend

to refine our model through hyperparameter tuning, which is anticipated to enhance the model’s ability to generalize across varied scenarios [5].

2 Project Goals

The primary objectives of our project are outlined as follows:

1. **Train a Custom YOLOv5 Model:** Establish a baseline for Drone and UAV detection performance to serve as a reference point for subsequent enhancements.
2. **Enhance Architecture:** Improve the YOLOv5 model by integrating advanced architectural components such as C3TR, GhostConv, and Focus layers to increase inference speed and accuracy.
3. **Hyperparameter Tuning:** Optimize hyperparameters including Shear, Scale, Mixup, and Translation to improve the model’s robustness and detection capabilities.

Each goal is aimed at pushing the boundaries of what is possible with YOLOv5 for Drone and UAV detection, ensuring that our model can operate effectively in real-world scenarios.

3 Data Preparation

The data preparation phase involved sourcing images from two distinct datasets available on Kaggle [6]. One dataset contained images of Drones, and the other consisted of UAVs, with both sets featuring images captured from various distances, sizes, and angles. We meticulously reviewed the images to ensure we understood the labeling and that it was logical, subsequently merging them into a single dataset. Since the datasets were initially separate, with the default class label "0", we reassigned the UAV images to class "1" to create two distinct classes and mixed them together. The data was then split into training and test sets (80-20 ratio).

Each image is accompanied by its corresponding bounding box annotation, which marks the object’s location within the image and its class. For example, as shown in this sample annotation, each bounding box is defined by its X and Y coordinates, width, height, and the class of the object. In this case, class "1" represents a UAV.

	Drones	UAV'S
Train	3155	450
Test	788	112




Figure 1: Our dataset

4 Background on YOLO

YOLO (You Only Look Once) is an influential real-time object detection system that has significantly impacted the field of computer vision. YOLO's approach differs from traditional object detection methods by combining the tasks of bounding box prediction and object classification into a single neural network model, enabling it to predict at high speeds suitable for real-time applications.

The fundamental concept of YOLO is to partition the input image into a grid, with each grid cell tasked with predicting objects that fall within its area. Each cell is responsible for predicting several bounding boxes and associated confidence scores that indicate the likelihood of object presence. These predictions encompass both the coordinates of the bounding box and the probabilities of object classes.

The YOLO architecture incorporates several key elements:

- **Residual Blocks:** These blocks facilitate the training of deep networks by allowing gradients to flow through the architecture, mitigating the vanishing gradient problem.
- **Bounding Box Regression:** YOLO employs a regression approach to directly predict the bounding box coordinates.
- **Intersection Over Union (IoU):** IoU is a metric used to evaluate the accuracy of the predicted bounding box by measuring its overlap with the ground truth box.
- **Non-Maximum Suppression (NMS):** As a post-processing step, NMS removes redundant bounding boxes, ensuring that each detected object is uniquely identified.

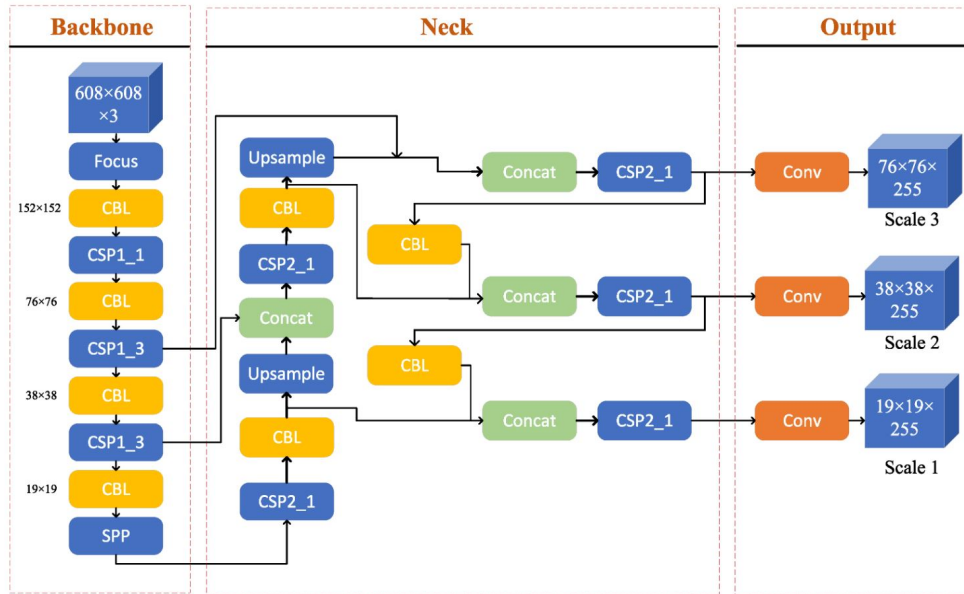


Figure 2: The architecture of the YOLOv5 method

5 YOLOv5 Enhancements

In our project to improve the YOLOv5 model, we implemented several enhancements. These enhancements include C3, C3TR, GhostConv, and Focus layers. Below is a detailed description of each of these components:

5.1 C3

The C3 module, short for Cross Stage Partial Network (CSPNet) stage 3, is designed to enhance feature fusion capability while reducing computational complexity. The C3 module consists of the following key concepts:

- **Partial Convolution:** This technique splits the input feature map into two parts. One part undergoes a series of convolutional operations, while the other part bypasses these operations. This split reduces the number of convolution operations required, making the process more efficient.
- **Cross-Stage Connection:** After processing one part of the feature map with convolutions, it is merged back with the bypassed part using a cross-stage hierarchy. This ensures that the gradient flow is maintained throughout the network, improving the learning process and preserving essential feature information.
- **Efficient Learning:** By reducing the number of parameters and ensuring better gradient flow, the C3 module allows the network to learn more efficiently. It captures diverse and rich features from the input data without significantly increasing computational load.

5.2 C3TR

The C3TR module is an extension of the C3 module, integrating transformer layers to capture long-range dependencies in the feature maps. The components of C3TR include:

- **Transformer Integration:** Transformer layers are incorporated within the C3 architecture to model global context information. Transformers are adept at capturing long-range dependencies and relationships within data, which is beneficial for object detection tasks.
- **Enhanced Feature Representation:** By combining the strengths of convolutional neural networks (CNNs) and transformers, C3TR provides a robust representation of input features. This hybrid approach leverages local feature extraction by CNNs and global context modeling by transformers.
- **Improved Object Detection:** The ability to capture both local and global features enhances the model's performance in detecting objects, especially those with complex relationships or occlusions.

5.3 GhostConv

GhostConv is an innovative convolutional operation designed to reduce computational cost while maintaining high accuracy. It operates based on the following principles:

- **Primary and Ghost Feature Maps:** GhostConv first applies a standard convolution to obtain a small number of primary feature maps. It then generates additional feature maps, known as ghost feature maps, through inexpensive linear operations.

- **Computational Efficiency:** By generating more feature maps from fewer intrinsic feature maps, GhostConv significantly reduces the number of parameters and computational overhead. This makes the model more efficient without compromising on accuracy.
- **Maintaining Performance:** Despite the reduction in computation, GhostConv ensures that the model’s performance remains high. The generated ghost feature maps retain essential information needed for accurate object detection.

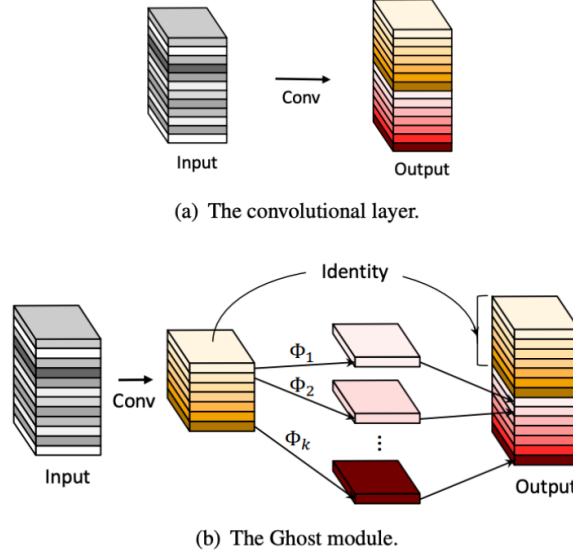


Figure 3: Ghost model

5.4 Focus

The Focus module is designed to reduce the spatial resolution of input feature maps efficiently. It works through the following mechanism:

- **Patch Extraction:** The Focus module extracts patches from the input image. This is done by dividing the input image into smaller patches and rearranging them along the channel dimension.
- **Channel Concatenation:** The extracted patches are concatenated along the channel dimension before being fed into the subsequent convolutional layers. This approach allows the model to capture fine-grained information from the input images.
- **Efficiency and Speed:** By reducing the spatial size of the input feature maps, the Focus module speeds up the processing. It enhances the model’s ability to detect small objects and improves overall efficiency.

To enhance the efficiency of our neural network, we conducted an experiment

6 Model Comparison and Results

Our experimental evaluation aimed to compare the baseline YOLOv5 model with various enhanced versions, focusing on improvements in detection accuracy and computational efficiency for Drone and UAV detection. We measure *Mean Average Precision (mAP)* at different IoU thresholds: mAP at IoU of 0.50 (mAP50), which reflects the precision of the model when the predicted bounding box and the ground truth have an overlap of 50%, and mAP at IoU thresholds ranging from 0.50 to 0.95 (mAP50-95), which averages the precision across a range of IoU thresholds, providing a more comprehensive assessment of the model’s accuracy.

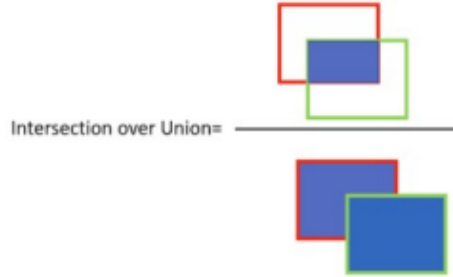


Figure 4: IoU, green- predicted b- box, red- truth b-box

Additionally, we consider the *inference speed*, quantified in *Giga Floating Point Operations Per Second (GFLOPS)*, which indicates how quickly the model can process images and make predictions. A lower GFLOPS value is desirable for our model as it means it can run efficiently on less powerful devices and is more energy-efficient, making it practical for real-world applications. We examine the *model complexity* by counting the total number of parameters in the model. A higher number of parameters often suggests a more complex model that could potentially capture more detailed information, but may also require more computational resources and could be prone to overfitting.

The table below presents the results of our model comparisons:

Model	Map50	Map50-95	GFLOPS	Model size
Baseline	0.96	0.65	4.2	1,761,871
Baseline - with HP	0.82	0.34	4.1	1,761,871
Ghost c3 and conv	0.86	0.48	2.3	939,275
Ghost c3 and conv - with HP	0.54	0.2	2.3	939,275
C3TR instead of c3	0.88	0.49	4.1	1,762,063
C3TR instead of c3 - with HP	0.6	0.21	4.1	1,762,063
Focus	0.9	0.49	N/A	1,761,871
Focus - with HP	0.55	0.21	N/A	1,761,871
Ghost c3 and conv with focus	0.86	0.48	2.3	943,683

Table 1: Comparison of YOLOv5 baseline and enhanced models.

7 Conclusions

The comparison reveals several trade-offs between accuracy, computational efficiency, and model complexity:

Accuracy vs. Efficiency: The baseline model, while offering the highest accuracy metrics, also demands more computational resources (higher GFLOPS). Enhanced models such as Ghost c3 and conv, and the Ghost c3 and conv with Focus, achieve a good balance between accuracy and efficiency, providing decent precision with significantly reduced computational demands and model size.

Model Complexity: Models with fewer parameters, like those incorporating Ghost modules, generally offer better computational efficiency but at the cost of reduced detection accuracy. On the other hand, models with more parameters, such as those using C3TR or the baseline, often achieve higher accuracy but are computationally more intensive and larger in size.

Trade-offs in Hyperparameters: Adjusting hyperparameters generally results in decreased accuracy across the board, suggesting that while efficiency improvements might be possible, they can come at the cost of model performance.

Overall, the optimal choice depends on the specific application requirements, balancing the need for high detection accuracy with the constraints of computational resources and model size. For practical deployment, especially on less powerful devices, models with lower GFLOPS and smaller sizes while maintaining acceptable accuracy are preferable.

8 Future Work

Future work on this project could explore several avenues for further improvement and application. One potential direction is the integration of additional sensor data, such as thermal or multi-spectral imaging, to enhance the model's ability to detect Drones in challenging conditions, such as low light or inclement weather. Expanding the dataset to include a broader range of Drone types and operational scenarios could also improve the model's generalization and robustness. Additionally, exploring the use of advanced techniques such as federated learning could enable the model to be trained collaboratively across multiple devices while preserving privacy and reducing computational load. Finally, deploying the model in real-world scenarios, coupled with continuous monitoring and updates based on new data, would provide valuable feedback for iterative improvements and ensure its effectiveness in practical applications.

9 Individual Contributions

We collaboratively explored potential subjects, reviewed relevant articles, and identified suitable enhancements for the YOLOv5 model. This initial phase involved selecting the right focus for our project and determining the best components to tune for improving YOLOv5 ability for Drone and UAV detection. Below is a short description of the distinct contribution of each member of the group to the project.

Alon led the Design of Experiments (DOE) part. He explored the existing YOLOv5 model, identified the ways to alter the different elements in the code, and sourced relevant data from Kaggle. He also handled the data preparation, including merging the datasets of Drones and UAVs, and ran the model with the baseline architecture. Alon also took a big part in both presentation and final report preparations.

Shahar led the technical part of running the models and solved any issue related to modifying and running heavy models on PyTorch smoothly. She ran models incorporating the Focus layer and analysed their performance. Shahar also took a big part in both presentation and final report preparations.

Andrei ran models with the C3TR instead of c3 enhancements and analysed the results. Andrei led the preparation of the final report, converted the Proposal one-pager and final report into LaTeX format and helped with preparing the presentation.

Noa specialized in the YOLOv5 background architecture and hyperparameters. She ran the models incorporating the GhostConv enhancements and evaluated their performance. She led the preparation of the presentation and took a big part in the final report preparation, ensuring accurate documentation of the project's findings.

References

- [1] Bombe, M.K. Unmanned Aerial Vehicle (UAV) Market Worth 21.8 billion by 2027- Pre and Post COVID-19 Market Analysis Report by Meticulous Research. 11 June 2020.
- [2] Kumar, R.; Kumar, P.; Tripathi, R.; Gupta, G.P.; Gadekallu, T.R.; Srivastava, G. SP2F: A secured privacy-preserving framework for smart agricultural Unmanned Aerial Vehicles. *Comput. Networks* 2021, 187, 107819.
- [3] Abro, G.E.M.; Zulkifli, S.A.B.M.; Masood, R.J.; Asirvadam, V.S.; Laouiti, A. Comprehensive Review of UAV Detection, Security, and Communication Advancements to Prevent Threats. *Drones* 2022, 6, 284.
- [4] M. Nalamati, A. Kapoor, M. Saqib, N. Sharma and M. Blumenstein, "Drone Detection in Long-Range Surveillance Videos," 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 2019, pp. 1-6, doi: 10.1109/AVSS.2019.8909830.
- [5] Zhai, X.; Huang, Z.; Li, T.; Liu, H.; Wang, S. YOLO-Drone: An Optimized YOLOv8 Network for Tiny UAV Object Detection. *Electronics* 2023, 12, 3664.
- [6] S. Shikamaru, "Amateur Drone Detection and Tracking Dataset," 2019. [Link to Dataset](#).