Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5, 708-726
2024
Publisher: Learning Gate
DOI: 10.55214/25768484.v8i5.1737
© 2024 by the authors; licensee Learning Gate

Comparative analysis of UAV detection and tracking performance: Evaluating YOLOv5, YOLOv8, and YOLOv8 DeepSORT for enhancing anti-UAV systems

Kamphon Suewongsuwan[1], Natchanun Angsuseranee[2*], Prasatporn Wongkamchang[1], Khongdet Phasinam[3]
[1]Faulty of Engineering, Navaminda Kasatriyadhiraj Royal Air Force Academy, Bangkok, Thailand; kamphon.s@thingsanalytic.com (K.S.); prasatporn_w@rtaf.mi.th (P.W.).
[2]Faculty of Engineering and Architecture, Rajamangala University of Technology Suvarnabhumi, Phranakhon Si Ayutthaya, Thailand; natchanun.a@rmutsb.ac.th (N.A.).
[3]Faculty of Food and Agricultural Technology, Pibulsongkram Rajabhat University, Phitsanulok, Thailand; phasinam@psru.ac.th (K.P.)

Abstract: This article presents a comprehensive comparative analysis of the performance of three prominent object detection and tracking models, namely YOLOv5, YOLOv8, and YOLOv8 DeepSORT, in the domain of UAV detection and tracking. The study aims to assess the effectiveness of these models in enhancing anti-UAV systems. A series of experiments were conducted using diverse datasets and evaluation metrics to evaluate the detection and tracking capabilities of each model. The results provide valuable insights into the strengths and limitations of YOLOv5, YOLOv8, and YOLOv8 DeepSORT, shedding light on their potential applications in anti-UAV systems. The findings of this study contribute to the advancement of UAV detection and tracking technologies and serve as a guide for researchers and practitioners in the field of anti-UAV systems.

Keywords: *Convolutional neural network, Deep learning, Object detection, Object tracking, Real-time Tracking with a deep association metric, Simple online Unmanned aerial vehicle.*

1. Introduction

Over the past few years, unmanned aerial vehicles (UAVs) have gained popularity worldwide, particularly for recreational activities and photography. It's important to recognize that the military sector leads the way in UAV technology, as military drones are built to resemble full-sized aircraft and are equipped with advanced features for long-range and high-altitude missions. The industry's growth in the last decade reflects the widespread use of UAVs across different sectors, including government, private organizations, and the general public.

However, current applications demonstrate deficiencies in terms of legality and scrutiny. Instances have been observed wherein UAVs are misused, resulting in infringements upon privacy rights. Additionally, the utilization of this technology by criminals can potentially lead to significant incidents. Nevertheless, anti-UAV technology is predominantly confined to military applications, with limited implementation in the private or household sectors. Therefore, it is imperative to establish a comprehensive system that encompasses all aspects to effectively prepare for the impending circumstances.

The anti-UAV system used in this project is divided into three phases: the Long-range for the detection phase, the Middle range for the detection and identification of the UAV model phase, and the Jamming phase. The detection process is split into two stages because the current technology has limitations in obtaining clear information through remote sensing.

© 2024 by the authors; licensee Learning Gate
* Correspondence: natchanun.a@rmutsb.ac.th

The objective of this research is to develop a comprehensive anti-aircraft system in the scope of the middle range that can be used in all sectors. This system will integrate an image processing component that will help identify different types of Unmanned Aerial Vehicles (UAVs), allowing appropriate responsive actions to be taken. The research evaluates the performance of the You Only Look Once (YOLO) model in detecting and tracking UAVs, as well as identifying their models for various purposes. The focus is specifically on embedded-based object detection using microcontroller units (MCUs) such as the NVIDIA Jetson MCU, The Jetson MCU was chosen for its availability to both private and public sectors, making it an ideal choice for anti-UAV development. Its affordability and ease of procurement also contribute to its significance, as it can be widely adopted at an affordable cost. This is particularly crucial for the second phase of the anti-UAV system, which involves middle-range detection and identification of UAV models.

2. Background and Related Work

The YOLO algorithm, which enables real-time object detection, is commonly trained on large datasets such as the Microsoft Common Objects in Context (MS COCO) dataset. The MS COCO dataset is extensively used and consists of images of complicated everyday scenes with over 80 labeled object categories per image. By training YOLO on the MS COCO dataset, it is able to accurately detect a broad range of objects in real-world situations. The combination of YOLO and the MS COCO dataset has been applied in various areas such as surveillance, autonomous driving, and robotics.

2.1. YOLOv5

YOLOv5, renowned for its exceptional effectiveness and precision, is a framework specifically designed for object detection. It leverages deep neural networks to identify and categorize items within images or videos. This is accomplished by creating a grid from the input image and generating predictions for bounding boxes and class probabilities for each grid cell. The real-time, multi-object detection capability enabled by this approach renders YOLOv5 particularly well-suited for applications that demand swift and accurate object recognition.

One of the primary advantages of YOLOv5 is its superior performance in comparison to earlier iterations. By incorporating advancements in deep learning techniques such as feature pyramid networks and anchor-based object detection, the framework achieves a more streamlined design. Its exceptional accuracy and rapid inference times make YOLOv5 an optimal choice for real-time object detection workloads.

Furthermore, YOLOv5 exhibits the ability to recognize objects of diverse sizes, shapes, and orientations, encompassing a wide range of object types. This versatility, coupled with its extensive training data, allows YOLOv5 to generalize effectively and achieve precise detections even in challenging circumstances. As a consequence, the framework finds significant application in various domains, including robotics, autonomous driving, surveillance systems, and object tracking, where accurate and real-time object recognition plays a crucial role. The adaptability, speed, and accuracy of YOLOv5 contribute to its popularity among developers and researchers engaged in object identification applications.

In this research, the YOLOv5 model is employed as a real-time object detection framework renowned for its remarkable speed and accuracy. One notable aspect of YOLO is its unique capability to detect overlapping objects using a sophisticated grid structure within each smaller layer, efficiently transferring relevant information across multiple layers. Currently, YOLOv5 has advanced to version 7, presenting several model options for evaluation. YOLOv5 comprises five distinct variants, namely n, s, m, l, and x, each offering different levels of accuracy and detection performance.

2.2. YOLOv8

YOLOv8 is an enhanced version of the You Only Look Once (YOLO) algorithm for real-time object detection. It employs a feature pyramid network (FPN) and a spatial attention mechanism in

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5: 708-726, 2024
DOI: 10.55214/25768484.v8i5.1737
© 2024 by the authors; licensee Learning Gate

combination to enhance its detection accuracy and efficiency. The following are the notable features of YOLOv8: Table 1. Font sizes of headings. Table captions should always be positioned *above* the tables.

Table 1.
YOLOv8 versus YOLOv5 +FPS comparison on different environments, Input Size: 640*640 [1].

YOLOv8/v5 variant name	CPU, Core i5-10500H	GPU, NVIDIA GTX 1650 Max Q	Jetson nano developer kit
YOLOv5s	15 FPS	45 FPS	28 FPS
YOLOv5n	7.5 FPS	36 FPS	15 FPS
YOLOv5m	3.3 FPS	25 FPS	6 FPS
YOLOv8s	13 FPS	50 FPS	N/A
YOLOv8n	7.1 FPS	46 FPS	N/A
YOLOv8m	2.6 FPS	25.8 FPS	N/A

- YOLOv8 adopts a Feature Pyramid Network (FPN) architecture that can detect objects of different sizes, making it effective in detecting both small and large objects within an image.
- It also uses a spatial attention mechanism that enables the model to concentrate on crucial areas of the image, thereby reducing false positives and improving object detection accuracy.
- YOLOv8 uses the lightweight CSPDarkNet backbone network to minimize computation and memory usage.
- It utilizes anchor-free detection to identify objects without requiring pre-defined anchor boxes, which increases its efficiency.
- YOLOv8 can handle occlusions better than earlier versions of YOLO, making it more robust in real-world scenarios. In summary, YOLOv8 can detect smaller objects with high accuracy in complex scenes while maintaining real-time performance.

For this research, YOLOv8 will be evaluated as a real-time object detection model that exhibits exceptional speed and accuracy. The main feature that sets YOLOv8 apart is its capability to detect overlapping objects by utilizing a complex grid structure in each smaller layer and then propagating this information to subsequent layers. The pre-trained models of YOLOv8, namely Detect, Segment, Pose, and Classify, have been trained using the COCO and ImageNet datasets. YOLOv8 offers five variants (n, s, m, l, and x) with varying levels of accuracy and detection performance.

2.3. CSPDarkNet53

The CSPDarkNet53 backbone is a variant of the DarkNet neural network architecture that finds frequent utilization in computer vision tasks, particularly in models designed for object detection. The acronym "CSP" within CSPDarkNet53 represents Cross Stage Partial connections, which entails the integration of skip connections facilitating efficient information flow across multiple stages of the network. This architectural design enhances gradient propagation during training and promotes effective feature reuse, thereby contributing to superior model performance.

The CSPDarkNet53 backbone denotes the layer count of the underlying network architecture. The "53" in CSPDarkNet53 represents the composition of residual blocks, pooling layers, and convolutional layers. The specific arrangement and configuration of these layers significantly contribute to the network's ability to capture and represent intricate visual patterns. In essence, the extensive feature extraction capabilities offered by the CSPDarkNet53 backbone assumes a pivotal role in object detection.

2.5. NVIDIA Jetson Series

Raspberry Pi and Arduino are popular microcontroller boards, but in this context, let's focus on the

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5: 708-726, 2024
DOI: 10.55214/25768484.v8i5.1737
© 2024 by the authors; licensee Learning Gate

difference between Raspberry Pi and NVIDIA Jetson. Raspberry Pi offers more advanced capabilities compared to Arduino, with a powerful CPU, GPU, and versatile features suitable for a wide range of applications. On the other hand, Arduino is a simpler microcontroller board designed for embedded systems and projects prioritizing low power and real-time control. The choice between Raspberry Pi and Arduino depends on specific project requirements, computational needs, and budget considerations.

Raspberry Pi and NVIDIA Jetson are prominent platforms for embedded computing, but they exhibit distinct differences in terms of their capabilities and target applications. Here is a comparative analysis of the two platforms:

Performance: In terms of performance, NVIDIA Jetson boards generally surpass Raspberry Pi. Jetson devices incorporate robust GPUs specifically designed for AI and ML tasks, delivering accelerated computing capabilities. Conversely, Raspberry Pi relies on a more general-purpose CPU and GPU combination, suitable for a wide range of applications but potentially lacking the performance of Jetson in AI-specific workloads.

AI and ML Capabilities: NVIDIA Jetson platforms are purpose-built for AI and ML applications, boasting optimized hardware and software support for deep learning frameworks. They excel in tasks such as AI inference, neural network training, and computer vision. While Raspberry Pi can run AI and ML models, it may not offer the same level of performance and dedicated hardware acceleration tailored for these tasks.

GPU Acceleration: NVIDIA Jetson devices feature dedicated GPUs that provide substantial computational power for parallel processing and accelerating AI workloads. In contrast, Raspberry Pi possesses a GPU of lesser power and lacks the same emphasis on AI acceleration.

Use Cases: Raspberry Pi finds common usage in general-purpose computing, IoT projects, home automation, media centers, and educational initiatives. On the other hand, NVIDIA Jetson is more commonly employed in AI-driven applications, robotics, computer vision, autonomous vehicles, drones, and other tasks that necessitate high-performance AI inference or training.

To summarize, Raspberry Pi offers a cost-effective and versatile embedded computing platform suited for regular projects, while NVIDIA Jetson furnishes more potent AI-specific capabilities with hardware acceleration. The choice between the two hinges on the specific requirements of your project, performance needs, and budgetary considerations.

To select an appropriate NVIDIA Jetson series for this work, there are various factors to consider in the computational requirements of running the YOLOv5 and 8 model and the specific needs of your UAV detection application. Here are some factors to consider when selecting a Jetson model:

- Processing Power: YOLOv8 is a deep neural network, and to function effectively, it needs a specific amount of processing power. Think about the trained model's complexity and layer count. More demanding models can be handled by the stronger Jetson models.
- Real-Time Processing: If real-time processing is essential for the UAV detection application, consider a Jetson model that has the processing power necessary to complete the task in the allotted time.
- Power Consumption: Power efficiency is frequently crucial for UAV applications. Make sure the Jetson model's power requirements are in line with the system's power limitations by taking this into account.
- GPU Performance: For quick inference, YOLOv5 and 8 benefit from a strong GPU. Because it directly affects how quickly the detection model can draw conclusions, consider the Jetson model's GPU performance.

Based on these considerations, here are some Jetson models that can consider:

- Jetson Nano: The Jetson Nano is an entry-level model that provides a good balance between price, power consumption, and computational performance. It is suitable for small-scale deployments and applications that don't require real-time processing.
- Jetson Xavier NX: The Jetson Xavier NX offers a higher level of performance than the Jetson Nano. It has a more powerful GPU and higher power consumption but provides better

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5: 708-726, 2024
DOI: 10.55214/25768484.v8i5.1737
© 2024 by the authors; licensee Learning Gate

computational capabilities. It can handle more complex models and real-time processing.

- Jetson AGX Xavier: The Jetson AGX Xavier is the most powerful Jetson model currently available. It offers the highest computational power and is suitable for demanding applications that require real-time processing and advanced AI capabilities.

However, the choice depends on the specific requirements of the UAV detection application and budget. Consider the trade-offs between computational power, real-time processing, power consumption, and budget to select the most appropriate Jetson model for the application.

2.6. YOLOv5 Versus YOLOv8

Two widely recognized object identification algorithms, namely YOLOv5 and YOLOv8, share several key components that contribute to their impressive performance. Firstly, both algorithms utilize the CSPDarknet53 architecture as their foundation, providing a robust basis for efficient and accurate detection. Moreover, anchor boxes, a commonly employed technique, are utilized by both algorithms to enhance object detection accuracy. The adoption of Non-Maximum Suppression (NMS), a technique aimed at suppressing redundant detections of the same object, is another prominent feature of both algorithms. Additionally, post-processing techniques are employed by YOLOv5 and YOLOv8 to refine the accuracy of their object detection outputs. The training processes of both methods leverage the widely used Adam optimizer for optimization purposes. Lastly, the architectural designs of both YOLOv5 and YOLOv8 incorporate the Mish activation function, facilitating effective information propagation and enabling reliable feature extraction. These similarities and differences in the object detection field between YOLOv5 and YOLOv8 are worth highlighting.

YOLOv5 has gained recognition for its exceptional accuracy in object detection. An impressive performance has been achieved, with an average precision of 50.5% on the COCO dataset. Notably, YOLOv5 has overcome the challenge of detecting small objects, which was a limitation in earlier versions of the YOLO framework. Furthermore, YOLOv5 has demonstrated its effectiveness in reliably identifying pedestrians in video feeds, showcasing its utility in real-world scenarios.

YOLOv8 represents a significant advancement in accuracy compared to YOLOv5, surpassing its predecessor in the YOLO series. The YOLOv8s model achieved an average precision of 51.4% on the COCO dataset, while the YOLOv8m model achieved an even higher average precision of 54.2%. Notably, YOLOv8 exhibits improved performance in detecting small objects while addressing various limitations of YOLOv5. These enhancements in accuracy and overall performance significantly enhance the potential and utility of the YOLOv8 framework.

When it comes to real-time applications, YOLOv5 excels in optimization, boasting an impressive frame rate per second (FPS). Among the different versions, the 'n' version of YOLOv5 offers the highest FPS, making it particularly suitable for real-time applications where speed is crucial.

In comparison, YOLOv8 exhibits a slightly lower FPS than YOLOv5 when executed on CPUs. However, it still maintains a respectable FPS for real-time applications and outperforms YOLOv5 in terms of FPS on certain GPUs. Notably, the 'n' version of YOLOv8 is particularly well-suited for embedded devices such as the Jetson Nano, delivering optimal performance within resource-constrained environments.

The measurement of Frames Per Second (FPS) holds great significance in the field of image processing as it dictates the rate and continuity of image or video processing. A higher value of FPS signifies the processing of a greater number of frames per second, thereby resulting in a more seamless and immediate visual output. The FPS parameter plays a crucial role in image processing by directly influencing the smoothness, responsiveness, and real-time capabilities of the system. Consequently, it becomes a pivotal aspect to consider for applications reliant on swift and accurate image processing, as it profoundly influences their overall performance and user experience.

However, attaining a high frame rate per second (FPS) can present a formidable challenge due to the necessity for efficient hardware and optimized algorithms. This endeavor frequently involves

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5: 708-726, 2024
DOI: 10.55214/25768484.v8i5.1737
© 2024 by the authors; licensee Learning Gate

striking a delicate equilibrium between computational resources, encompassing processing power and memory, and the targeted degree of accuracy and intricacy demanded by image processing tasks.

Table 1, show the comparison of the efficiency of the speed FPS of YOLO v5 and v8 in s, n, and m models with the processing device.

2.7. Simple Online and Realtime Tracking with a Deep Association

The Simple Online and Real-time Tracking (SORT) [2] algorithm is a method for tracking objects that employs fundamental techniques, including Kalman filters and Hungarian algorithms, and purports to surpass numerous other online trackers. SORT consists of four principal components: detection, estimation, data association, and track identity creation/deletion [3]. During the detection stage, an object detector identifies objects in the frame to be tracked, which are subsequently passed to the next stage. In the estimation stage, the current frame's detections are propagated to the next frame by estimating the target's position using a constant velocity model. When a detection is associated with a target, the target state is updated based on the detected bounding box, with the velocity components optimally determined via the Kalman filter framework. The data association step computes a cost matrix, utilizing the intersection-over-union (IOU) distance between each detection and all predicted bounding boxes from existing targets. The Hungarian algorithm is employed to optimally solve the assignment problem. Assignments with an IOU below a specified threshold, referred to as IOUmin, are rejected, thereby addressing occlusion issues and maintaining track identities. The creation and deletion of track identities module is responsible for managing the creation and deletion of identities [4]. Unique identities are established and terminated based on the IOUmin criterion. If the overlap between a detection and a target is below IOUmin, it signifies an untracked object [4]. Tracks are terminated if they remain undetected for a specified number of frames, denoted as TLost. Tracking resumes under a new identity if an object reappears.

Among the various algorithms used for tracking multiple objects, Simple Online and Real-time Tracking with a Deep Association Metric (DeepSORT) [2] has emerged as one of the most efficient and robust approaches [5]. Initially developed as the SORT algorithm [3], it was designed to provide a streamlined approach to detection based online tracking, with a focus on effectively associating object detections in each frame. Leveraging the powerful capabilities of convolutional neural networks in object detection, SORT algorithm capitalized on their reputation for accuracy. Additionally, the algorithm incorporates two well-established methods in motion prediction and data association: the Hungarian algorithm [6] and the Kalman filter [7]. These algorithms enable the prediction of future object positions based on their current positions, as well as the estimation of more accurate current positions than what the sensor or algorithm alone can provide. This enhances the overall association process. For a visual representation of the entire algorithm, refer to Fig. 1.

2.8. Anti-UAV System Concept

The research presents the Anti-UAV System concept, which is structured into three distinct phases, refer in Fig. 2. The initial phase focuses on the detection of remote unmanned aerial vehicles (UAVs). Long-range detection plays a pivotal role in UAV surveillance. When a suspected UAV is detected within the detector's range, it triggers an alert for the mid-near UAV detection system, constituting the second phase of the anti-UAV system. Continuing from phase one, the principle lies in determining the direction from which the suspected UAV is approaching. This phase is dedicated to the identification of UAVs, enabling the anti-UAV system to thoroughly analyze and optimally respond to such suspected UAVs. This phase holds importance within the anti-UAV concept in this research.

It is crucial to highlight that employing standard anti-UAV systems through forceful means or regular jamming techniques can result in damage to ground property. This includes tourist attractions like parks or historic sites, as well as locations with flammable objects that may explode upon impact from an unmanned aerial vehicle's lithium polymer battery colliding with a solid surface. The purpose of anti-UAV jamming, constituting the third phase of the anti-UAV system, is different. By acquiring

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5: 708-726, 2024
DOI: 10.55214/25768484.v8i5.1737
© 2024 by the authors; licensee Learning Gate

identity information from suspected unmanned aerial vehicles and assessing potential responses, the system can effectively mitigate damage. If the system possesses a protocol database of such unmanned aircraft, it can devise procedures to minimize harm efficiently.

2.8.1. Anti-UAV System Concept

- The UAV Detection Radar System is a technology specifically designed for detecting and tracking unmanned aerial vehicles (UAVs) or drones. This system uses radar waves to locate UAVs in the airspace. It includes a radar transmitter that emits electromagnetic signals and a receiver that captures the signals reflected by the UAVs. By analyzing the Doppler shift and time delay of these signals, the system can determine the presence, position, and speed of the UAVs. The UAV Detection Radar System can detect both small and large UAVs and is capable of operating in different weather conditions. It is a crucial tool for monitoring airspace, ensuring border security, and protecting critical infrastructure. With its long-range capabilities and accurate tracking, the UAV Detection Radar System offers reliable detection and monitoring of UAV activity, facilitating effective response and countermeasures if necessary.
- The Acoustic Drone Detection System, also known as the UAV Detection Acoustic System, is an advanced technology used to detect and track unmanned aerial vehicles (UAVs) or drones. Unlike traditional radar systems that rely on radio waves, this system uses sound wave detection. It works by deploying a network of strategically placed acoustic sensors or microphones in a specific area. These sensors capture the unique sound signatures emitted by UAVs, including the distinct noises made by their propellers or motors. By analyzing the collected acoustic data, the system provides real-time information about the presence, exact location, and movement of UAVs. The UAV Detection Acoustic System performs exceptionally well in situations where radar-based systems face obstacles, such as environments with high levels of electromagnetic interference or when the drone's radar signature is significantly reduced. This technology is particularly valuable for identifying small UAVs operating at low altitudes, which can be challenging to detect visually or with traditional radar approaches.

2.8.2. Middle-Range Detection

- The RGB camera, also referred to as a color camera, is an imaging device designed to capture and record images or video within the visible light spectrum. The term RGB represents the primary colors, Red, Green, and Blue, which are combined to produce a wide range of colors perceivable by the human eye [9]. RGB cameras hold significant importance in the field of image capture, enabling the acquisition of high-quality color images. They find extensive applications across various domains, catering to diverse needs and requirements.

In the context of this research, RGB cameras were employed to collect image data for the purpose of monitoring suspected UAVs. However, it is essential to acknowledge that their effectiveness may be hindered by brightness levels during daylight. Consequently, these devices offer exceptional utility during daytime operations when ample sunlight allows for optimal detail and information clarity, surpassing the capabilities of alternative devices. Nonetheless, it is worth noting that RGB cameras are not ideally suited for nighttime use, as the absence of sunlight poses limitations in capturing objects or UAVs with sufficient clarity and detail.

- A thermal camera is a device that uses infrared radiation to capture and create images based on the temperature differences of objects or surfaces. Unlike RGB cameras that rely on visible light, thermal cameras detect and measure the infrared radiation emitted by objects, allowing for the visualization of heat patterns. In the context of UAV detection, thermal cameras offer unique advantages. They can detect the heat signatures emitted by UAVs, even in low light or nighttime conditions when visual detection may be challenging. This makes thermal cameras particularly useful for identifying UAVs that may be operating covertly or trying to evade

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5: 708-726, 2024
DOI: 10.55214/25768484.v8i5.1737
© 2024 by the authors; licensee Learning Gate

visual detection. Additionally, thermal cameras can be effective in distinguishing between UAVs and other objects, as the heat signature of a UAV differs from that of surrounding elements example case, in the case of a UAV moving through the building in the afternoon the temperature of the building is almost or equal to UAV, due to the heat that the building accumulates during the day.

However, thermal cameras also have limitations. They are primarily sensitive to temperature differences rather than providing fine details of objects like RGB cameras. Additionally, environmental factors such as weather conditions and background temperature can affect the accuracy of thermal imaging.

Despite these limitations, thermal cameras offer valuable capabilities for UAV detection and surveillance, especially in scenarios where visual or RGB-based methods may not be sufficient or reliable.

- EO/IR/LRF (Electro-Optical/Infrared/Laser Range Finder) sensors play a critical role in an Anti-UAV system, combining multiple technologies to effectively detect, track, and identify unmanned aerial vehicles (UAVs). These sensors encompass EO, IR, and LRF capabilities, each serving specific purposes. EO sensors capture high-resolution images and video using visible light and near-infrared wavelengths, functioning similarly to RGB cameras, providing detailed visual information in daylight conditions. IR sensors operate in the infrared spectrum, detecting thermal radiation emitted by objects, excelling in low-light or nighttime situations, and tracking UAVs based on heat signatures. LRF sensors emit laser beams and measure their reflection time to determine precise distances, enabling accurate positioning of UAVs within the Anti-UAV system. The integration of EO, IR, and LRF sensors offers comprehensive and robust detection, tracking, and identification capabilities for UAVs in Anti-UAV systems.

Combining these three sensor technologies, EO/IR/LRF sensors offer comprehensive surveillance capabilities for Anti-UAV systems. They visually detect UAVs, track them through thermal signatures, and accurately determine their distances. These sensors provide numerous advantages, including 24/7 surveillance capabilities, detection at various ranges, and enhanced situational awareness. The integration of laser range finders improves target tracking and engagement accuracy.

However, it's important to consider limitations. Adverse weather conditions and environmental factors can affect sensor performance. Additionally, each sensor's detection range and accuracy may vary, EO/IR/LRF sensors are vital for Anti-UAV systems, providing a comprehensive approach to detect, track, and counter UAV threats. By integrating visual, thermal, and range-finding technologies, these sensors enhance situational awareness and improve the effectiveness of Anti-UAV operations.

2.8.3. Jamming Phase

- RF Jamming in an anti-UAV system involves the intentional interference or disruption of radio frequency (RF) signals to counter the unauthorized or malicious use of Unmanned Aerial Vehicles (UAVs). By transmitting signals or noise on the same frequency bands used by UAV communication systems, RF Jamming aims to disable or neutralize the UAV. This can be achieved by overpowering or blocking the UAV's remote-control signals or telemetry data, effectively preventing the UAV from functioning appropriately. RF Jamming techniques can disrupt the communication links between the UAV and its operator, rendering UAVs inoperable or lost control.
- Protocol-Specific Jamming focuses on targeting the communication protocols specific to UAV control and telemetry systems. UAVs often utilize a variety of wireless communication protocols, such as Wi-Fi, Bluetooth, or proprietary protocols, for command and control, data transmission, and navigation. Protocol-Specific Jamming aims to exploit vulnerabilities or weaknesses in these protocols to disrupt or manipulate the communication between the UAV and Remote controller by disrupting the protocol messages, which the Anti-UAV system can interfere with the UAV's operation, disrupt the navigation system, or take control of the UAV.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5: 708-726, 2024
DOI: 10.55214/25768484.v8i5.1737
© 2024 by the authors; licensee Learning Gate

- Cellular jamming for an anti-UAV system refers to intentionally interfering with the cellular signals utilized by UAVs for communication and control purposes. UAVs heavily depend on cellular networks to establish communication links between the remote pilot and UAV, as well as to transmit telemetry data and receive commands by employing a dedicated cellular jamming system is feasible to disrupt or block the cellular signals that UAVs rely on, So UAVs is unable to establish connections with remote pilots or receive commands. This involves employing specialized equipment capable of transmitting interfering signals within the frequency bands allocated for cellular networks. These interfering signals overpower or disrupt legitimate cellular signals, preventing UAVs effectively from establishing connections or maintaining communication systems with remote pilots or ground control stations.

3. Work Research Methodology

3.1. Data Preparation

3.1.1. Video was Converted into an Image Frame

Videos were converted into image frames using the "Scene video filter" of VLC, an automatic screenshot of the video was taken at an interval of half a second [10]. For example, if the video had a frame rate of 60 FPS, one image frame was taken every 30 frames. For a 20 FPS video, one image frame was taken every 10 frames. which dataset is provided from This research received support from various sources, including the contributions of the following researchers and authors: 1. Anti-UAV: A Large MultiModal Benchmark for UAV Tracking" by N. Jiang, K. Wang, X. Peng, X. Yu, G. Li, and Z. Han, affiliated with the University of Chinese Academy of Sciences (UCAS) in Beijing, China, in 2021 [11]. 2. The 17th IEEE International Conference on Advanced Video and Signal-based Surveillance in 2021 [12]. 3. A Deep Learning Approach to Drone Monitoring" by Yueru Chen, Pranav Aggarwal, Jongmoo Choi, and C.-C. Jay Kuo, associated with the University of Southern California in California, USA, in 2017. These contributions have provided valuable insights and findings that have contributed to the advancement of this research [13].

Next was the elimination of images without UAV from the dataset. After that, the remaining image frames were 5,477 (640x512 resolution) images and 2,150 images (1920 × 1088 resolution) giving a total of 7,627 images. The dataset was further expanded through augmentation.

3.1.2. Labelling

Bounding box labeling was done using Supervisely. Its labeling interface allows precise and efficient labeling, reducing the possibility of human error in labeling ground truth data. However, the export format of the labels was only in Supervisely format (JSON). Other open-source Python-based annotation tools exported directly in YOLO format include LabelImg [14].

Roboflow was used to convert the Supervisely format labels to the desired frameworks (such as Darknet, Tensorflow, PyTorch, etc.). An alternative to Roboflow is an open-source method of converting Supervisely labels to YOLO format [9].

3.2. Data Augmentation

In order to enhance the effectiveness of the UAV detection system in various scenarios, it is crucial to include a diverse range of UAV representations, such as UAVs flying through buildings heated by the sun, in the dataset. However, it is important to acknowledge that there may be inherent biases in the dataset that the researcher may not immediately notice, leading to overfitting on the training dataset. To address this concern, it is assumed that more valuable information can be extracted from the training dataset by transforming the images in different ways. This process, known as data augmentation, aims to simulate a wider range of UAV data in the sky, thereby preventing potential overfitting. However, the researcher must carefully select the appropriate data augmentation techniques for the specific situation. The two main categories of transformations in data augmentation are spatial-

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5: 708-726, 2024
DOI: 10.55214/25768484.v8i5.1737
© 2024 by the authors; licensee Learning Gate

level and pixel-level, each offering different approaches to modifying the images during the augmentation process. An example of the data augmentation are shown in Fig. 3.

Pixel-level transformations involve altering the pixel values within the image while preserving the overall spatial structure. Examples of pixel-level transformations include changing brightness, adjusting contrast, adding noise, blurring, or applying color transformations. These modifications primarily focus on individual pixels or small local neighborhoods within the image.

On the other hand, Spatial-level transforms involve adjusting the overall spatial properties of the image. The entire image or some portions of it may be rotated, scaled, cropped, flipped, or sheared during these modifications. The overall spatial arrangement can be considerably affected by spatial-level modifications that change the shape, orientation, or positioning of objects within the image.

However, the study utilized both spatial-level transformations and data splitting to enhance the performance of object detection systems. It was found that spatial level transformations were particularly effective in improving system performance. The following image transformations:

3.2.1. Random Flip (Only horizontal)

Implementing continuous horizontal flip and vertical flip settings aims to mitigate the impact of cloudy image datasets. However, it is crucial to consider the potential adverse effects of allowing detectors to learn features through unconventional methods, as this could cause confusion and hinder their overall learning process.

3.2.2. Random Rotation Between -40% to +40 %

In order to prevent the occurrence of vertical inversion, the rotation setting has been configured to operate between a range of -25 to +25 percent. The Unmanned Aerial Vehicle (UAV) tilt appears visible within the UAV image dataset at particular angles. Excessive rotation settings may cause the detection to falsely learn that the UAV is upside down, which may compromise the detection's ability.

3.2.3. Random Brightness Adjustment from -25% to +25%

The brightness settings encompass a spectrum from a negative 25% to a positive 25% increase. In real-world situations, specific camera or sensor orientations demand the ability to identify things both during the day and at night. It is essential to keep in mind that exposure to sunlight might affect how bright a signal the camera or sensor receives.

3.2.4. Random Grayscale: Apply to 25% of Images

During the gray-scale augmentation process, an input image undergoes a random transformation, converting it into a single-channel gray-scale result image. Consequently, the model assigns relatively less significance to color as a signal.

3.2.5. Random Blur: Up to 2px

The dataset is considered to apply a blur setting of up to 2 pixels considering the dataset provided to the detector, such as a thermal camera, for learning purposes containing cloudy images, particularly thermal images.

Consider:

- The camera remains stationary while the objects it is tasked to detect are frequently in motion, such as in the case of a camera observing traffic.
- The camera is in motion while the objects are designed to detect and remain stationary, as exemplified by a user utilizing a mobile app to scan a book cover.
- Both the camera and the objects it is detecting are in motion, as demonstrated by a drone flying over a windy field of crops to carry out detection tasks. The images were adjusted to different sizes, including 416 × 416, 512 × 512, 608 × 608, and 640 × 640. To maintain the aspect ratios of the images, black padding was added to prevent any distortion in the aspect ratio of the UAV. By

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5: 708-726, 2024
DOI: 10.55214/25768484.v8i5.1737
© 2024 by the authors; licensee Learning Gate

using augmentation techniques, the dataset was expanded significantly from 7,627 images to a whopping 73,180 images.

3.3. Data Splitting

As mentioned in the previous section, the dataset was comprised of 55,884 images. Researcher adopted a rule-of-thumb in data splitting for dataset sizes from 100 to 10,000, which was 70:30 for training and validation sets. This is typical for datasets that have even distribution, meaning the training and validation set are not too different from each other. However, in this study, the dataset had uneven distribution. High-resolution images were used in training the neural networks to enable them to detect smaller objects. Then, the trained neural networks were tested on the target application of this study, which was lower-resolution mobile phone images [9]. One may think that validating the trained model with a dataset that has a different distribution does not truly evaluate the performance of the trained model. Researchers can measure the learning performance to be true.

Subsequently, the dataset underwent division into four distinct portions, adhering to a ratio of 70:10:10:10, denoted as the training set, training validation set, validation set, and test set. Within the training and training-validation sets, the inclusion comprised high-resolution image frames. The training validation set encompassed high-resolution image frames that were unseen during the training process and served the purpose of assessing whether the trained model exhibited signs of overfitting solely to the training images.

3.4. Transfer Learning with YOLO Model

Within the scope of the research study, the researcher opted to employ transfer learning as a methodology, leveraging pre-trained models, specifically focusing on the YOLOv5 model series (YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x) and the YOLOv8 model series (YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x). To align the model with the research objectives, the original MS COCO class was replaced by a custom dataset class encompassing three distinct UAV models: DJI Mavic, DJI Phantom, and DJI Inspire. Further categorization of each UAV model was carried out based on the type of image utilized: Thermal Image, RGB Image, and Grayscale Image. By adopting this methodology, the researcher was able to refine the pre-trained YOLOv8 model using the custom dataset, thereby enabling precise detection and classification of the specific UAV models utilizing diverse image types.

Dataset classification, the researcher has organized the data into nine distinct classes, each representing a specific combination of UAV model and image type. The classes include DJI Inspire with Grayscale Image, DJI Inspire with IR Image, DJI Inspire with RGB Image, DJI Mavic with Grayscale Image, DJI Mavic with IR Image, DJI Mavic with RGB Image, DJI Phantom with Grayscale Image, DJI Phantom with IR Image, and DJI Phantom with RGB Image.

The utilization of this classification methodology enriches the research process by facilitating a focused analysis and insightful interpretation of the dataset. This classification technique empowers researchers to conduct a comprehensive exploration of the dataset, enabling them to investigate and compare different UAV models and image types in a meticulous manner. Consequently, the dataset becomes a valuable resource that can be thoroughly explored and leveraged to draw significant and conclusive findings.

3.5. Object Detection Training

To assess how well the YOLO model series performs, the researcher thoroughly examined each variant of the model (n, s, m, l, and x). The models were put through rigorous testing using different image sizes (416x416, 512x512, 608x608, 640x640). This systematic approach allowed for a comprehensive comparison of how well the model performed under various conditions. The results of this analysis are important in understanding the best combination of the YOLO model series and image

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5: 708-726, 2024
DOI: 10.55214/25768484.v8i5.1737
© 2024 by the authors; licensee Learning Gate

size for specific use cases. This information provides valuable guidance for future implementations and helps enhance the overall performance of the YOLO model in different scenarios.

3.6. Object Tracking with SORT

Normally, The YOLO model can detect and track the target object, but SORT enhances the tracking performance to reduce object loss tracking.

4. Experimental Results

4.1. Dataset Labeling Using Roboflow

The researchers make use of Roboflow as a data labeling tool in the process of classifying the data. Within this procedure, the researchers employ object classification accompanied by bounding boxes to guarantee the accurate identification of each model of the unmanned aerial vehicle (UAV) as shown in Fig. 4.

4.2. Experimental Results of Optimized Training Method of YOLOv5, YOLOv8, and YOLOv8 with DeepSORT Model on Custom Data by Using Weights and Biases Display

The Weights and Biases display was used to assess the experimental outputs of the optimum training method for YOLOv5, YOLOv8, and YOLOv8 with DeepSORT models on specialized data at each data size 416 pixels presented in Tables 2-4, data size 512 pixels presented in Tables 5-7, data size 608 pixels presented in Tables 8-10, and data size 640 pixels presented in Tables 11-13.

Table 2.
Detail training - performance comparison within data size 416 pixels in each model of YOLOv5.

Model	Metric/ Precision	Metric/ Recall	Metric/ mPA 0.5:0.95	Metric/ mPA 0.5
YOLOv5s	0.5514	0.9143	0.3276	0.6758
YOLOv5n	0.9142	0.9381	0.4082	0.9449
YOLOv5m	0.9670	0.9445	0.4425	0.9700
YOLOv5l	0.9657	0.9664	0.4437	0.9699
YOLOv5x	0.9460	0.9880	0.6450	0.9780

Table 3.
Detail training - performance comparison within data size 416 pixels in each model of YOLOv8.

Model	Metric/Precision	Metric/ Recall	Metric/mPA - 0.5:0.95	Metric/mPA 0.5
YOLOv8n	0.9500	0.9760	0.6620	0.9820
YOLOv8s	0.9460	0.9800	0.6610	0.9830
YOLOv8m	0.9540	0.9780	0.664	0.9810
YOLOv8l	0.9500	0.9820	0.667	0.9810
YOLOv8x	0.9470	0.9850	0.6680	0.9820

Table 4.
Detail training - performance comparison within data size 416 pixels in each model of YOLOv8 with DeepSORT of YOLOv8.

Model	Metric/ Precision	Metric/ Recall	Metric/ mPA 0.5:0.95	Metric/ mPA 0.5
YOLOv8n + DeepSORT	0.9400	0.9860	0.6650	0.9790
YOLOv8s + DeepSORT	0.9410	0.9840	0.6540	0.9800
YOLOv8m + DeepSORT	0.9370	0.9870	0.6600	0.9780
YOLOv8l + DeepSORT	0.9400	0.9840	0.6540	0.9600
YOLOv8x + DeepSORT	0.9320	0.9790	0.6570	0.9800

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5: 708-726, 2024
DOI: 10.55214/25768484.v8i5.1737
© 2024 by the authors; licensee Learning Gate

Table 5.
Detail training - performance comparison within data size 512 pixels in each model of YOLOv5.

Model	Metric/ Precision	Metric/ Recall	Metric/mPA - 0.5:0.95	Metric/ mPA 0.5
YOLOv5s	0.4366	0.9767	0.9761	0.9719
YOLOv5n	0.4280	0.9709	0.9637	0.9712
YOLOv5m	0.4425	0.9779	0.9796	0.9820
YOLOv5l	0.4059	0.9647	0.9582	0.9658
YOLOv5x	0.9440	0.9440	0.6480	0.9810

Table 6.
Detail training - performance comparison within data size 512 pixels in each model of YOLOv8.

Model	Metric/ Precision	Metric/ Recall	Metric/mPA - 0.5:0.95	Metric/ mPA 0.5
YOLOv8n	0.9490	0.9780	0.6650	0.9770
YOLOv8s	0.9480	0.9750	0.6610	0.9820
YOLOv8m	0.9470	0.979	0.6650	0.9840
YOLOv8l	0.9500	0.9810	0.6740	0.9840
YOLOv8x	0.9490	0.9830	0.6690	0.9830

Table 7.
Detail training - performance comparison within data size 512 pixels in each model of YOLOv8 with DeepSORT of YOLOv8.

Model	Metric/ Precision	Metric/ Recall	Metric/mPA - 0.5:0.95	Metric/ mPA 0.5
YOLOv8n + DeepSORT	0.9410	0.9740	0.6660	0.9810
YOLOv8s + DeepSORT	0.9390	0.9840	0.6640	0.9840
YOLOv8m + DeepSORT	0.9440	0.9830	0.6640	0.9820
YOLOv8l + DeepSORT	0.9460	0.9830	0.6650	0.9810
YOLOv8x + DeepSORT	0.9470	0.9860	0.6680	0.9830

Table 8.
Detail training - performance comparison within data size 608 pixels in each model of YOLOv5.

Model	Metric/ Precision	Metric/ Recall	Metric/ mPA 0.5:0.95	Metric/ mPA 0.
YOLOv5s	0.9444	0.9575	0.3869	0.9600
YOLOv5n	0.9469	0.9490	0.3909	0.9564
YOLOv5m	0.9510	0.9496	0.4160	0.9545
YOLOv5l	0.9704	0.9680	0.4209	0.9645
YOLOv5x	0.9490	0.9890	0.6420	0.9810

Table 9.
Detail training - performance comparison within data size 608 pixels in each model of YOLOv8.

Model	Metric/ Precision	Metric/ Recall	Metric/ mPA 0.5:0.95	Metric/ mPA 0.5
YOLOv8n	0.9430	0.9750	0.6670	0.9780
YOLOv8s	0.9360	0.9630	0.6650	0.9770
YOLOv8m	0.9430	0.9900	0.6640	0.9830
YOLOv8l	0.9500	0.9810	0.6740	0.9840
YOLOv8x	0.9270	0.9700	0.6510	0.9780

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5: 708-726, 2024
DOI: 10.55214/25768484.v8i5.1737
© 2024 by the authors; licensee Learning Gate

Table 10.
Detail training - performance comparison within data size 608 pixels in each model of YOLOv8 with DeepSORT of YOLOv8.

Model	Metric/ Precision	Metric/ Recall	Metric/ mPA 0.5:0.95	Metric/ mPA 0.5
YOLOv8n + DeepSORT	0.9430	0.9710	0.6600	0.9790
YOLOv8s + DeepSORT	0.9420	0.9810	0.6640	0.9800
YOLOv8m + DeepSORT	0.9500	0.9760	0.6620	0.9810
YOLOv8l + DeepSORT	0.9460	0.9850	0.6660	0.9830
YOLOv8x + DeepSORT	0.9500	0.9820	0.6640	0.9630

Table 11.
Detail training - performance comparison within data size 640 pixels in each model of YOLOv5.

Model	Metric/ Precision	Metric/ Recall	Metric/ mPA 0.5:0.95	Metric/ mPA 0.5
YOLOv5s	0.9758	0.9735	0.4318	0.9727
YOLOv5n	0.9449	0.9460	0.4107	0.9513
YOLOv5m	0.9796	0.9780	0.4498	0.9728
YOLOv5l	0.9799	0.9784	0.4460	0.9751
YOLOv5x	0.9470	0.9850	0.6450	0.9770

Table 12.
Detail training - performance comparison within data size 640 pixels in each model of YOLOv8.

Model	Metric/ Precision	Metric/ Recall	Metric/ mPA 0.5:0.95	Metric/ mPA 0.5
YOLOv8n	0.9410	0.9820	0.6690	0.9800
YOLOv8s	0.9500	0.9870	0.6680	0.9830
YOLOv8m	0.9580	0.9790	0.6700	0.9850
YOLOv8l	0.9530	0.9880	0.6670	0.9860
YOLOv8x	0.9540	0.9920	0.6680	0.9840

Table 13.
Detail training - performance comparison within data size 640 pixels in each model of YOLOv8 with DeepSORT of YOLOv8.

Model	Metric/ Precision	Metric/ Recall	Metric/ mPA 0.5:0.95	Metric/ mPA 0.5
YOLOv8n + DeepSORT	0.9510	0.9810	0.6680	0.9830
YOLOv8s + DeepSORT	0.9420	0.9830	0.6650	0.9810
YOLOv8m + DeepSORT	0.9520	0.9800	0.6670	0.9840
YOLOv8l + DeepSORT	0.9300	0.9590	0.6470	0.9740
YOLOv8x + DeepSORT	0.9450	0.9860	0.6720	0.9820

5. Discussion

The selection of the most suitable YOLO model relies on its precision value, which acts as a metric for evaluating its capacity to effectively detect and track small Unmanned Aerial Vehicles (UAVs) with accuracy.

Based on the findings presented in Tables 14-17, it becomes evident that the models belonging to the YOLO series demonstrate minimal discrepancies in terms of precision, with the top-performing models exhibiting comparable results. However, it is crucial to acknowledge that the evaluation conducted in this context does not encompass the assessment of Frames Per Second (FPS) values, which were not the subject of investigation in the present study.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5: 708-726, 2024
DOI: 10.55214/25768484.v8i5.1737
© 2024 by the authors; licensee Learning Gate

Table 14.
The best results of YOLO model experiments were compared across each series within data size 416 pixels.

Model	Metric/ Precision	Metric/ Recall	Metric/ mPA 0.5:0.95	Metric/ mPA 0.5
YOLOv5m	0.9670	0.9445	0.4425	0.9700
YOLOv8m	0.9540	0.9780	0.6640	0.9810
YOLOv8s + DeepSORT	0.9410	0.9840	0.6540	0.9800

Table 15
The best results of YOLO model experiments were compared across each series within data size 512 pixels.

Model	Metric/ Precision	Metric/ Recall	Metric/ mPA 0.5:0.95	Metric/ mPA 0.5
YOLOv5x	0.9440	0.9830	0.6480	0.9810
YOLOv8l	0.9500	0.9810	0.6740	0.9840
YOLOv8x + DeepSORT	0.9470	0.9860	0.6680	0.9830

Table 16.
The best results of YOLO model experiments were compared across each series within data size 608 pixels.

Model	Metric/ Precision	Metric/ Recall	Metric/ mPA 0.5:0.95	Metric/ mPA 0.5
YOLOv5l	0.9704	0.9640	0.4209	0.9645
YOLOv8l	0.9500	0.9810	0.6740	0.9840
YOLOv8x + DeepSORT	0.9500	0.9820	0.6640	0.9630

Table 17.
The best results of YOLO model experiments were compared across each series within data size 640 pixels.

Model	Metric/ Precision	Metric/ Recall	Metric/ mPA 0.5:0.95	Metric/ mPA 0.5
YOLOv5l	0.9799	0.9784	0.4460	0.9751
YOLOv8m	0.9580	0.9790	0.6700	0.9850
YOLOv8m + DeepSORT	0.9520	0.9800	0.6670	0.9840

Nevertheless, the obtained results indicate that the YOLOv8 models, including the YOLOv8 with the DeepSORT model, showcase higher FPS rates compared to YOLOv5. This discovery highlights the potential of YOLOv8 models to surpass YOLOv5 in terms of both accuracy and speed, positioning them as promising contenders for future advancements and progressions in the field of object detection systems.

6. Conclusions

In conclusion, when selecting an appropriate YOLO model, precision undoubtedly holds significant importance. However, it is equally crucial to evaluate the compatibility of the YOLO model with the processing device. Factors such as the availability of internal storage and the capability to allocate data space on the processor board should be assessed. This consideration ensures support for additional features, including integration with the Internet of Things (IoT), as well as the ability to accommodate functionalities like camera orientation adjustment for tracking the movement of the UAV. By conducting a thorough evaluation of both precision and compatibility, researchers and practitioners can make well-informed decisions regarding the choice of the YOLO model, thereby optimizing the effectiveness and versatility of their UAV detection and tracking systems.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5: 708-726, 2024
DOI: 10.55214/25768484.v8i5.1737
© 2024 by the authors; licensee Learning Gate

List of Abbreviations
YOLO: You only look once; MS COCO: Microsoft Common Objects in Context; UAV: Unmanned Aerial Vehicle; SORT: Simple Online and Real-time Tracking; DeepSORT: Simple Online and Real-time Tracking with a Deep Association; RGB: Red-Blue-Green; EO: Electro-Optical; IR: Infra-Red; LRF: Laser rangefinder; RF Jamming: Radio Frequency Jamming; IOU: Intersection-over-union; MCU: Microcontroller unit; IoT: Internet-of-Thing

Acknowledgements:
We extend our deepest gratitude to the staff of the Faculty of Engineering, The Graduate School, Navaminda Kasatriyadhiraj Royal Air Force Academy for their invaluable cooperation and support throughout the course of this research. Special thanks go to N. Chaikam and S. Pummalee for their essential contributions to data collection, without which this study would not have been possible.
We are also sincerely grateful to Navaminda Kasatriyadhiraj Royal Air Force Academy and Faulty of Engineering and Architecture, Rajamangala University of Technology Suvarnabhumi for providing the necessary facilities and resources that enabled the successful completion of this research.

Authors' Contributions:
Kamphon Suewongsuwan: Formal analysis; methodology; validation; writing–original draft; writing–review and editing. Natchanun Angsuseranee: Conceptualization; data curation; resources; validation; writing–original draft. Prasatporn Wongkamchang: Conceptualization; formal analysis; resources; supervision; writing–review and editing. Khongdet Phasinam: Investigation; validation; visualization; writing–review and editing.

Copyright:
© 2024 by the authors. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

References
[1] MR Munawar, Comparing Yolov5 and Yolov8: which one should you use? (2023) https://chr043416.medium.com/comparing-yolov5-and-yolov8-which-one-should-you-use-538ca550a25d. Accessed 1 June 2023.
[2] A Bewley, Z Ge, L Ott, F Ramos, B Upcroft, in 2016 IEEE International Conference on Image Processing (ICIP), Simple Online and Realtime Tracking, (IEEE, New York, US, 2016), pp. 3464-3468.
[3] HY Chen, CH Lin, JW Lai, YK Chan, Convolutional neural network-based automated system for dog tracking and emotion recognition in video surveillance. Appl. Sci. 13, 4596 (2023).
[4] Sanyam, Understanding multiple object tracking using deepsort. (2022) https://learnopencv.com/understanding-multiple-object-tracking-using-DeepSORT. Accessed 18 June 2023.
[5] N Wojke, A Bewley, D Paulus, in 2017 IEEE International Conference on Image Processing (ICIP), Simple Online and Realtime Tracking with a Deep Association Metric, (IEEE, New York, US, 2017), pp. 3645-3649.
[6] HW Kuhn, The hungarian method for the assignment problem. Nav. Res. Logist. (NRL) 52, 7-21 (2005).
[7] RE Kalman, RS Bucy, J, New results in linear filtering and prediction theory. Basic Eng. 83, 95-108 (1961).
[8] D Avola, L Cinque, A Diko, A Fagioli, GL Foresti, A Mecca, D Pannone, C Piciarelli, MS-faster R-CNN: multi-stream backbone for improved faster r-cnn object detection and aerial tracking from UAV images. Remote Sens. 13, 1670 (2021).
[9] G Guerrer, Consciousness-related interactions in a double-slit optical system. OSF Prepr. doi:10.31219/osf.io/zsgwp (2019).
[10] AI Parico, T Ahamed, Real time pear fruit detection and counting using YOLOv4 models and deep sort. Sensors 21, 4803 (2021).
[11] N Jiang, K Wang, X Peng, X Yu, Q Wang, J Xing, G Li, J Zhao, G Guo, Z Han, Anti-UAV: A Large Multi-Modal Benchmark for UAV Tracking. In Computer Vision and Pattern Recognition, arXiv preprint arXiv:2101.08466, (2021).
[12] Institute of Electrical and Electronics Engineers, The 17th IEEE international conference on advanced video and signal-based surveillance (AVSS 2021). (2021) http://www.avss2021.org/#. Accessed 27 July 2021.

[13] Y Chen, P Aggarwal, J Choi, CCJ Kuo, in 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), A Deep Learning Approach to Drone Monitoring, (IEEE, New York, US, 2017), pp. 686-691.

[14] Parico, A.I.B., Ahamed, T. (2023). Real-Time Pear Fruit Detection and Counting Using YOLOv4 Models and Deep SORT. In: Ahamed, T. (eds) IoT and AI in Agriculture. Springer, Singapore. https://doi.org/10.1007/978-981-19-8113-5_11

Figure Legends

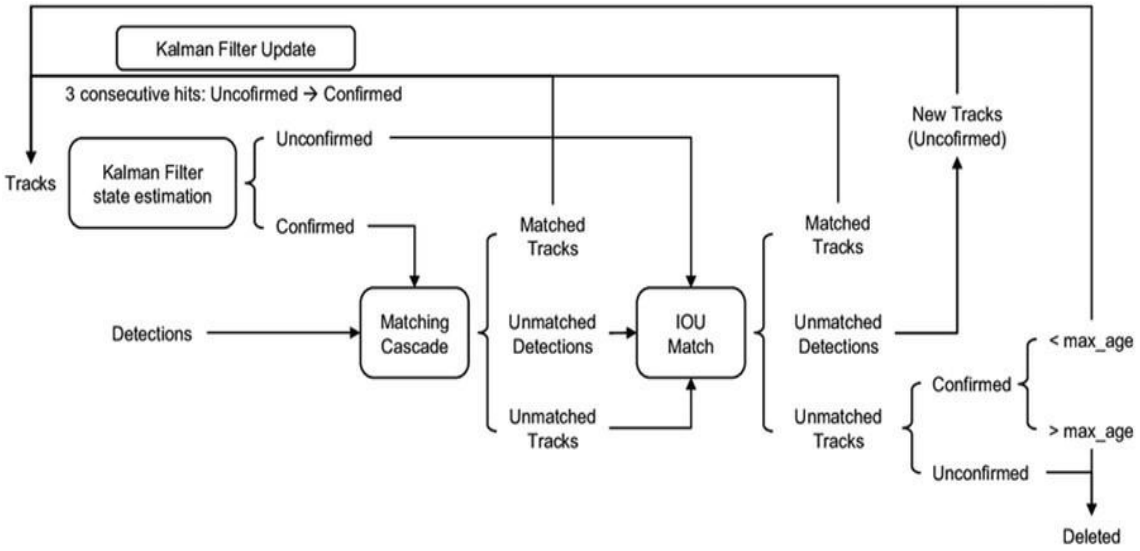

Figure 1.
DeepSort tracking algorithm flowchart [8].

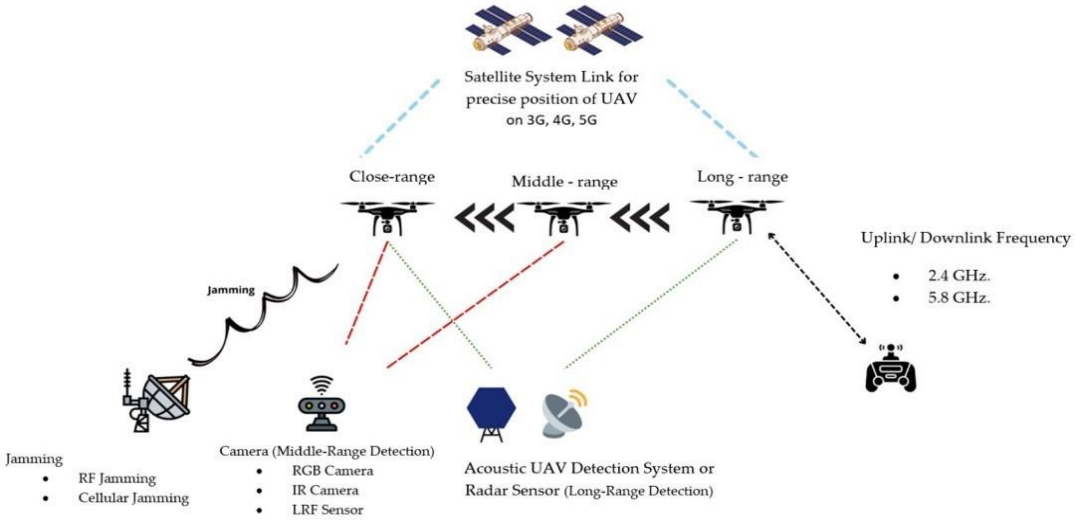

Figure 2.
The anti-UAV system conceptual.

© 2024 by the authors; licensee Learning Gate

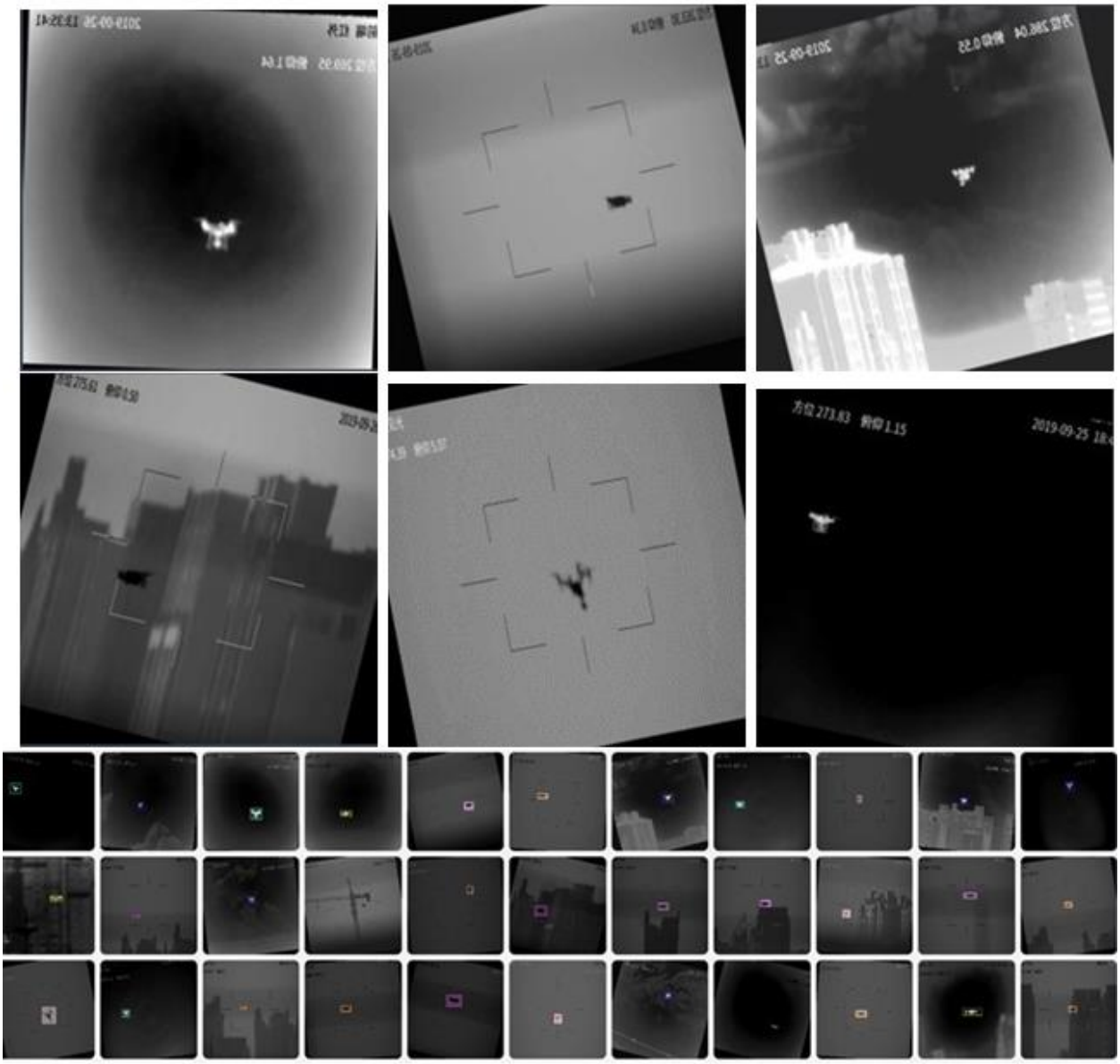

Figure 3.
Example data augmentation.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5: 708-726, 2024
DOI: 10.55214/25768484.v8i5.1737
© 2024 by the authors; licensee Learning Gate

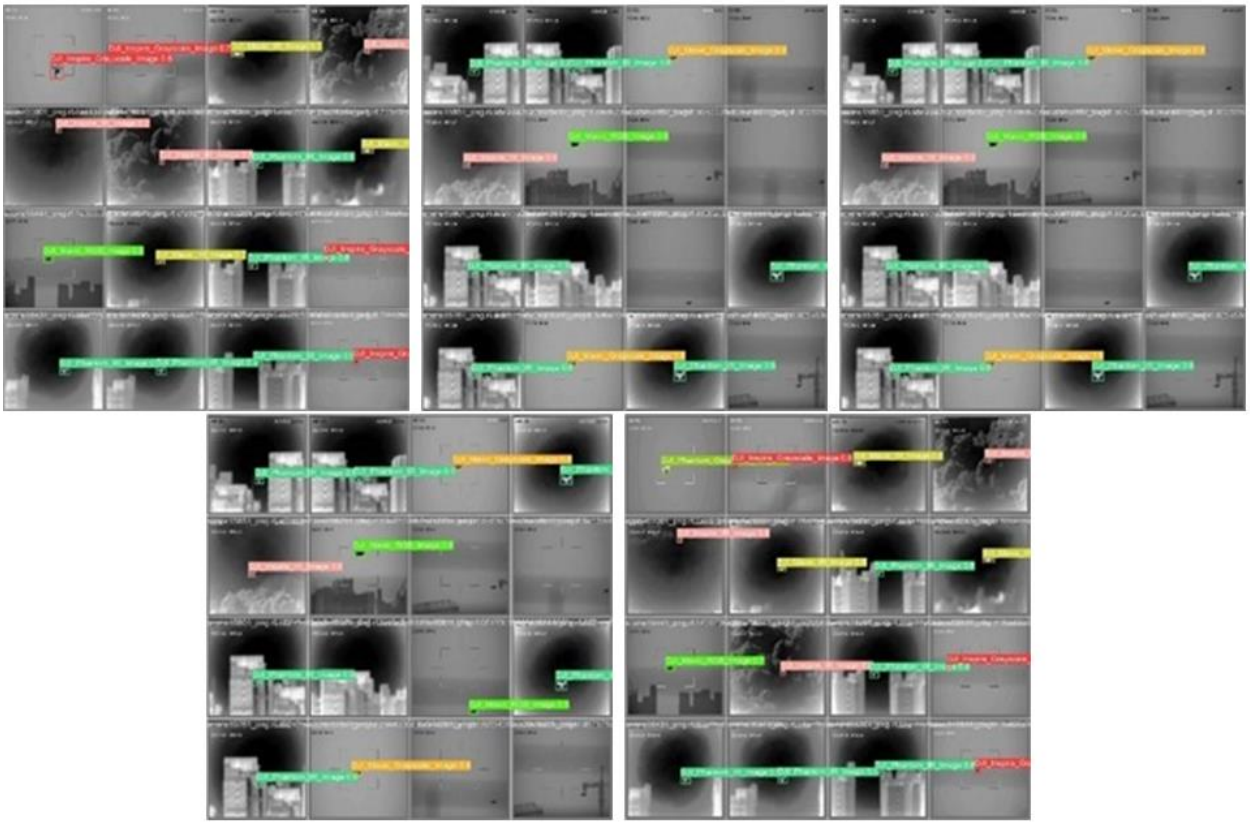

Figure 4.
Example val batch image.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5: 708-726, 2024
DOI: 10.55214/25768484.v8i5.1737
© 2024 by the authors; licensee Learning Gate