Proceedings of CONF-MLA 2025 Symposium: Intelligent Systems and Automation: AI Models, IoT, and Robotic Algorithms
DOI: 10.54254/2755-2721/2025.LD26953

"Hawkeye" Target Tracking System --- An Air-Ground Distributed Computing Intelligent Aerial Robot Target Tracking System

Jichang Ju

The Second High School Attached to Beijing Normal University, Beijing, China
pgdo4_youjingxi@163.com

Abstract. In recent years, rotor drones have been increasingly applied in disaster rescue scenarios, yet there remains a significant shortage of skilled drone operators. Therefore, research on drone target tracking systems is of great significance for intelligent drone-based rescue operations. This paper proposes an intelligent aerial robot target tracking system based on air-ground distributed computing ("Hawkeye" target tracking system), enabling drones to be applied to emergency rescue tasks in an automated, low-cost, and long-endurance manner. The system uses YOLOv8 target detection algorithm on the ground station to search for targets and the KCF tracking algorithm on the drone for visual target tracking. Finally, visual servo tracking is performed based on the target's position in the image. By employing air-ground distributed computing, this method significantly reduces the computational demands on the airborne side, lowers costs, and improves drone endurance. Simulation and real flight tests demonstrate the reliability and cost-effectiveness of the system. Compared to the Jetson Nano, our system achieves similar results with an 87% cost reduction and a 92% weight reduction.

Keywords: target detection, target tracking, distributed computing, drone, visual servoing

1. Introduction

Rotor drones have been increasingly used in disaster relief, showing great potential in rapid disaster assessment, search and rescue, supply delivery, and communication relay. However, major challenges remain, including a shortage of skilled drone operators, high costs, and poor endurance. Thus, studying automated, lightweight, and low-cost drone rescue systems is critical to enhancing efficiency in intelligent rescue.

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2025.LD26953



Figure 1. Extensive applications of rotary-wing UAVs in disaster rescue scenarios

Currently, drones have been applied in emergency rescue operations, but they mainly rely on skilled drone operators for manual operation. Although traditionally human-operated drones have improved rescue efficiency to some extent, they are still limited by the number of operators, their response speed, and differences in skill levels. In sudden emergency rescue scenarios, the shortage of drone operators makes it impossible to keep them on standby at all times; waiting for operators to arrive and then launch the drone takes too long—and in rescue, "time is life." Therefore, developing drone systems with autonomous search and rescue capabilities can not only significantly improve rescue efficiency but also save more lives at critical moments. Moreover, the intelligent computing power of autonomous search and rescue drone systems mainly resides in two components: the onboard system and the ground station. Intelligent computing at the ground station suffers from high latency, while the onboard side faces issues such as limited computing power, high cost, and heavy weight. Thus, how to leverage the computing power of both onboard and ground systems to build a low-cost, lightweight, and high-performance intelligent system has become a key research focus.

This paper proposes an intelligent aerial robot target tracking system based on air-ground distributed computing ("Hawkeye" target tracking system). Applied to drones, this system enables autonomous search and rescue, eliminating reliance on operator numbers, response times, and skill variability. Additionally, the system runs the computationally intensive YOLOv8 target detection algorithm on the ground station to search for targets, while the drone executes the lightweight KCF target tracking algorithm for visual tracking—achieving collaborative air ground distributed computing. Tracking tasks, which are latency-sensitive but computationally light, are handled via onboard edge computing; whereas search tasks, which are computationally heavy but less latency-sensitive, are processed on the ground station via image transmission. This method not only reduces the computing requirements on the drone but also lowers manufacturing costs, reduces weight, and improves flight endurance. Finally, visual servo tracking is performed based on the target's position in the image.

Main contributions:

1. We propose an air-ground distributed intelligent aerial robot target tracking system ("Hawkeye") and implement it in a drone with autonomous rescue capability.

2. By distributing tasks (tracking onboard, detection on the ground), we reduce airborne computing requirements, thereby lowering cost and weight and improving endurance.

3. Flight tests show the system achieves similar results to Jetson Nano with 87% lower cost and 92% less weight.

DOI: 10.54254/2755-2721/2025.LD26953

2. Related work

2.1. Target detection

Faster R-CNN [1], YOLO [2], and SSD [3] are among the most representative algorithms in recent years. Faster R-CNN introduced the Region Proposal Network (RPN), enabling end-to-end object detection and significantly improving detection speed and accuracy. The YOLO algorithm adopts a single forward pass approach to achieve real-time object detection, greatly reducing computational complexity while maintaining high accuracy. SSD combines the advantages of both Faster R-CNN andYOLO by performing dense sampling on feature maps of different scales, further enhancing the precision and speed of object detection.

In addition to these classical algorithms, many other research directions are being actively explored. For example, some research teams are developing unsupervised and self-supervised learning methods to reduce dependence on large amounts of labeled data. Others are investigating the integration of object detection with other computer vision tasks—such as segmentation and tracking—to enable more comprehensive scene understanding and analysis.

Overall, current implementations of object detection on drones mostly rely on GPU computation, which consumes significant computing power. Therefore, in this study, object detection is performed on the ground station.

2.2. Target tracking

Drone visual guidance based on optical images primarily uses single-object video tracking interpretation techniques, which can be categorized into two types: traditional methods based on correlation filtering and intelligent interpretation algorithms based on deep learning.

Traditional target tracking algorithms based on correlation filtering include MOSSE [4], CSK [5], KCF [6], and DSST [7]. These methods rely on manually designed feature extraction to match the search region in the video stream with the target template specified in the first frame. Such algorithms offer high computational speed and can run with low overhead on edge devices like drones. However, they often rely on strong assumptions and are only effective in certain scenarios, making it difficult to adapt to rapidly changing real-world environments.

Deep learning-based intelligent interpretation algorithms are currently the mainstream approach for single-object tracking. Methods based on Siamese networks—such as Siamese RPN [8], Siamese FC [9], and Siamese FC++ [10]—use relatively simple network structures to achieve single-object visual tracking with accuracy far exceeding that of traditional methods, and are suitable for a wide range of application scenarios. In recent years, single-object video tracking models based on Transformers, such as OSTrack [11] and MixFormer [12], have introduced one-stage tracking frameworks that eliminate the need to separately extract template and search region features, further simplifying the tracking process. However, due to their intensive computation and large parameter size, these methods are difficult to deploy on edge devices like drones, which are constrained by computing power and energy consumption.

2.3. Drone tracking systems

Currently, the development of drone target tracking systems is progressing slowly. DJI drones are equipped with auto-follow programs, where users can manually select a target on the smartphone (or

DOI: 10.54254/2755-2721/2025.LD26953

remote controller) screen for the drone to follow automatically. However, the target must be selected manually, and fully automatic target recognition is not yet achievable [13].



Figure 2. DJI mini 4 pro automatic follow mode publicity material

In academia, TianyeAlex released the tool tracker_kcf_ros [14] on GitHub, which implements the KCF tracking algorithm using a depth camera under the ROS framework. This allows robots to automatically follow specific targets using the KCF algorithm. When combined with a target detection algorithm, it can achieve automatic target recognition and tracking. After modifications, ROS can also be used for drone guidance.

In 2020, YugAjmera and Surya Pratap Singh used reinforcement learning and YOLOFlow [15] to implement autonomous drone-based target search and tracking. All computations were performed onboard the drone, enabling autonomous target detection and tracking. The method was validated through simulations of earthquake rescue scenarios but was not tested in actual flight, making it uncertain whether the drone's onboard computing power could support the processing requirements.

This method also requires the drone to carry both a fixed camera and an optical flow camera. Since optical flow cameras are not standard equipment on most drones, the hardware setup is complex and costly.

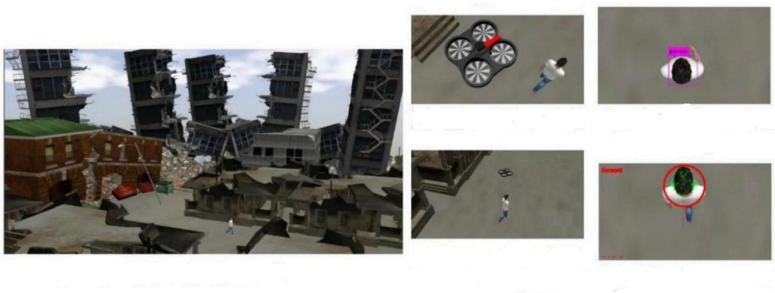


Figure 3. Methodology of ajmera and singh

In recent years, both domestic and international approaches to drone-based target tracking have typically involved running target tracking or detection algorithms directly on the drone. This requires high computational power, resulting in heavy and costly onboard computers.

Such methods make it difficult to achieve automatic target recognition and tracking using low-cost, lightweight onboard computers.

In response, this paper explores a method in which the ground station performs target detection to provide target cues for the drone, while the drone's onboard computer runs a tracking algorithm to

DOI: 10.54254/2755-2721/2025.LD26953

follow the indicated target. This approach reduces the performance demands on the drone's onboard computer, enabling low-cost, high-efficiency target tracking.

3. Hawkeye target tracking system

3.1. System workflow

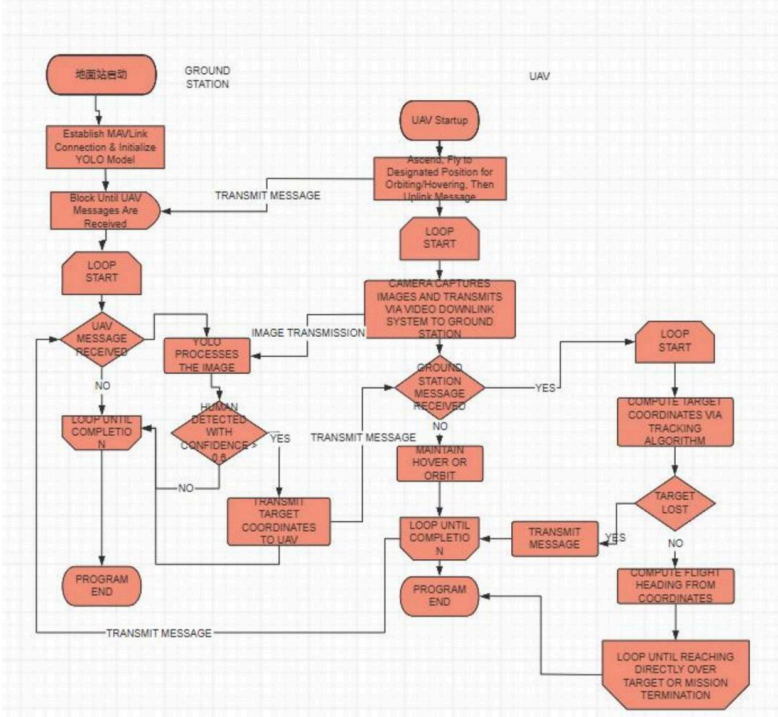


Figure 4. Task processing flow diagram

In the intelligent drone rescue system, the drone's onboard camera continuously captures real-time image data of the scene, which is transmitted back to the ground control station via an efficient image transmission system. As illustrated in Figure 4, once the onboard computer sends a command to initiate target inspection, the ground station immediately applies an advanced target detection algorithm to analyze the received images in depth, aiming to accurately identify the designated target to be tracked, with the confidence level exceeding a predefined threshold.

Once the target is successfully detected, the ground station promptly transmits the specific position and bounding box information of the target to the drone’s onboard computer via the telemetry radio. Upon receiving this information, the onboard computer immediately activates the target tracking algorithm and uses the received target position and bounding box as initialization parameters to precisely execute the algorithm for continuous tracking, thereby enabling the drone to steadily follow the target.

If tracking is lost due to environmental factors or dynamic changes in the target, the onboard computer will autonomously send a new target detection request to the ground station to ensure the continuity and accuracy of the rescue mission.

DOI: 10.54254/2755-2721/2025.LD26953

3.2. Detection algorithm

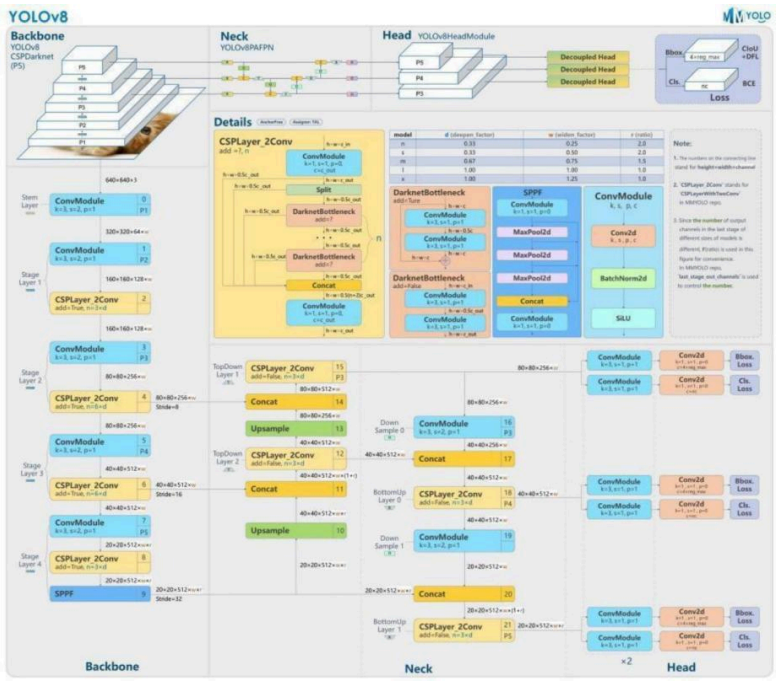


Figure 5. YOLOv8 model schematic

In this study, we selected the You Only Look Once algorithm (YOLO), developed by Ultralytics. YOLO is a one-stage object detection algorithm. Compared to two-stage algorithms—which first perform object proposal to locate potential objects and then classify each one—YOLO achieves end-to-end object detection through a single convolutional neural network (CNN) model. Specifically, YOLO's CNN divides the input image into an S×S grid, where each grid cell is responsible for detecting objects whose center points fall within it. Each grid cell predicts C class probability values, which represent the likelihood that the object within the predicted bounding box belongs to each of the C categories.

By integrating the predictions from all grid cells, a prediction tensor map is formed, which ultimately produces the object detection result. Furthermore, we employed an improved version of YOLO—YOLOv8. Compared to its predecessor YOLOv5, YOLOv8 offers improved detection accuracy. Based on the performance level of the GPU equipped on the ground station, we selected high-precision models from the YOLOv8 series for local object detection. Specifically, we used the YOLOv8x model for high-performance GPUs and the YOLOv8m model for mid-range GPUs.

All of these models were retrained to adapt to specific object recognition tasks (such as human body detection) under wide-angle camera settings and simulated image transmission conditions, which include image quality issues like fisheye distortion and scanline snow screen artifacts.

DOI: 10.54254/2755-2721/2025.LD26953

3.3. Tracking algorithm

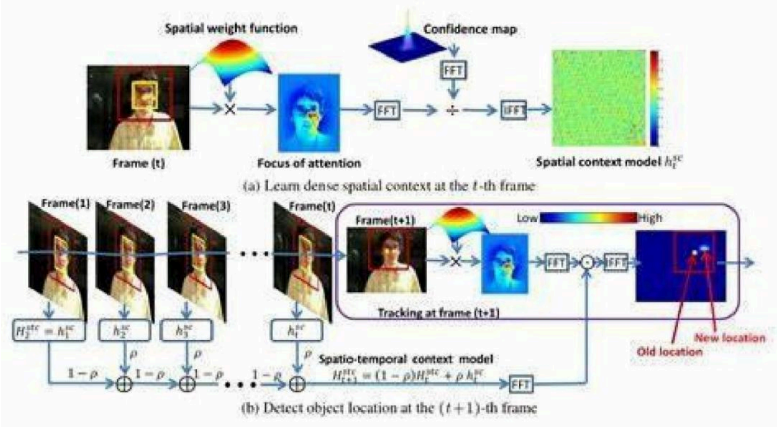


Figure 6. Diagram of the KCF algorithm

In this study, we conducted an in-depth exploration and adopted the Kernelized Correlation Filters (KCF) algorithm as the core method for target tracking. KCF is a discriminative tracking technique that is distinguished by its ability to dynamically train and optimize a target detector across consecutive frames during tracking. This mechanism allows the KCF algorithm to adapt in real time to changes in the appearance of the target within a video sequence, thereby ensuring accuracy and robustness in tracking.

Specifically, during the tracking process, the KCF algorithm first selects a region containing the target in the current frame as a positive sample, and then selects a series of surrounding regions as negative samples. The selection of these samples follows a principle: the closer a region is to the target, the more likely it is to be labeled as a positive sample, while regions farther from the target are more likely to be labeled as negative samples. In this way, the KCF algorithm effectively leverages features of both the target and its surrounding environment to provide rich and meaningful data for training the target detector.

During the training phase, the KCF algorithm uses kernelized correlation filtering to compute the correlation between positive and negative samples, thereby constructing a detector capable of distinguishing the target from the background. This detector is then used in the subsequent tracking process to determine whether the predicted target position in the next frame is accurate. If the detector deems the position to be correct, the algorithm updates the training set and incorporates the new detection result to further refine the detector. Through this iterative process, the KCF algorithm achieves stable tracking of the same target across consecutive frames.

It is worth noting that the KCF algorithm employs efficient computational strategies, such as Fast Fourier Transform (FFT), which enable it to maintain low computational complexity while still achieving real-time tracking performance. This characteristic makes the KCF algorithm particularly well-suited for real-time applications such as drone-based target tracking, where high responsiveness is essential.

In summary, as a discriminative tracking method, the KCF algorithm demonstrates outstanding performance in the field of target tracking. Its dynamic training of the detector, strategic sample selection, and efficient computation collectively form the foundation for high-speed, accurate tracking of the same target across continuous video frames.

DOI: 10.54254/2755-2721/2025.LD26953

3.4. Visual servo tracking

In the drone visual servo tracking system, we skillfully employed the Kernelized Correlation Filters (KCF) algorithm to accurately capture and continuously track targets within the camera's field of view. Renowned for its efficiency and accuracy in the field of target tracking, the KCF algorithm can swiftly lock onto target positions even in complex and dynamic environments. Once the KCF algorithm successfully acquires the target's position, the system immediately compares that position with the center of the camera's field of view. Based on the result of this comparison, the drone can adjust its heading and the vertical orientation of its camera in real time, ensuring that the target remains steadily centered in the frame.

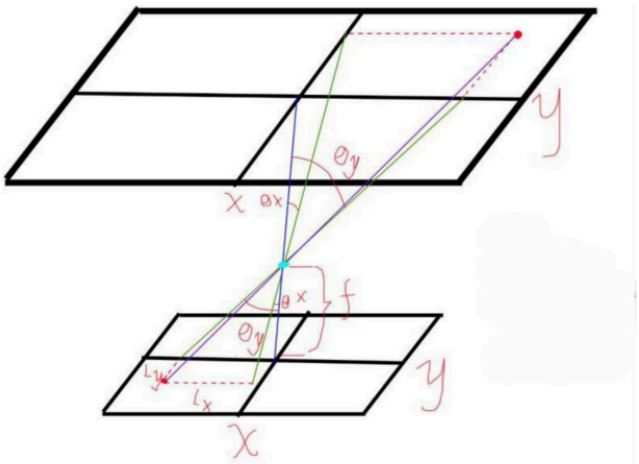
This dynamic adjustment mechanism not only improves the stability and accuracy of tracking but also significantly enhances the drone's adaptability. Notably, when the camera is tilted vertically downward to 90°, indicating that the drone is hovering and preparing for precise target localization, it will further adjust its position based on the target location provided by the KCF algorithm to ensure the target remains within the field of view. At this point, it can be considered that the drone has successfully arrived directly above the target, thereby being fully prepared for subsequent tasks such as precision strikes or data collection.

However, in real-world applications, the KCF algorithm may sometimes fail to track the target due to various complex factors, such as changes in lighting or target occlusion. To address this challenge, we designed a comprehensive fault-handling mechanism. If the KCF algorithm loses the target, the system will immediately send a command to the ground station, triggering a new round of target detection. In this way, even if unexpected situations arise during tracking, the system can quickly recover and continue executing the mission.

In addition, the angle between the tracked target and the camera's optical axis can be calculated using the following method (as shown in the figure below):

$$\theta_x = \arctan\left(\frac{L_x}{f}\right)$$

$$\theta_y = \arctan\left(\frac{L_y}{f}\right)$$



f:Focal Length Lx,Ly:Position of object within camera's field of view(Calculated by multiplying the x,y coordinates in the image by the pixel width to obtain)

Figure 7. Target-heading angle calculation

DOI: 10.54254/2755-2721/2025.LD26953

3.5. Air-ground communication

In this study, image transmission between the drone and the ground station is carried out using an analog signal at a frequency of 5.8 GHz. This data transmission process is managed by the telemetry radio module integrated into the flight control system. To ensure the effectiveness and compatibility of data transmission, the communication protocols between the onboard computer and the flight control system, as well as between the onboard computer and the ground station, all adhere to the MAVLink V2.0 standard.

MAVLink is a lightweight messaging protocol specifically designed for communication between ground control terminals (ground stations) and drones, as well as between internal components of onboard drone systems. Compared to MAVLink 1, which has a packet overhead of only 8 bytes (including start markers and packet loss detection), MAVLink 2 expands the overhead to 14 bytes. Nevertheless, it offers enhanced security and scalability.

Since the MAVLink protocol does not require additional framing structures, it is particularly suitable for bandwidth-limited communication scenarios.

Since its inception in 2009, MAVLink has been widely adopted for communication between various vehicles, ground stations, and other nodes, and has performed reliably even in challenging communication environments characterized by high latency and noise. Moreover, MAVLink includes features such as packet loss and corruption detection, as well as packet authentication, further enhancing the reliability and security of communication [16,17].

In the specific implementation of this study, in addition to transmitting regular packets (such as heartbeat packets), the ground station also sends an additional message called TARGET_POS to the onboard computer. This message contains four fields: X1, Y1, X2, and Y2, which respectively represent the X and Y coordinates of the top-left and bottom-right corners ofthe target's bounding box within the field of view.

Correspondingly, the onboard computer sends back a message to the ground station called MISSION_CONDITION, which includes a single field. When this field is set to 1, it indicates that the drone is currently performing target detection.

When the drone is set to Guided mode, the onboard computer uses the DroneKit library to send flight heading and camera orientation commands to the flight control system, enabling precise target tracking and imaging [18].

4. System implementation

4.1. Hardware platform



Figure 8. Test aircraft

The system includes:

DOI: 10.54254/2755-2721/2025.LD26953

Ground GPU: NVIDIA RTX4050
Onboard computer: Raspberry Pi Zero 2W
Frame: 450mm
Camera: 1800TVL
Gimbal: 2-axis
Image transmission: 5.8GHz, 1W
Flight controller: PX4

Table 1. Flight control system details

Component	Model/Specification
Ground Station GPU	NVIDIA RTX4050
Onboard Computer	PI ZERO 2W
Frame	Wheelbase 450mm
Camera	1800TVL
Gimbal	2-Axis
Video Transmitter (VTX)	5.8GHz, 1W Power
Flight Controller	PX4

The drone operation system is equipped with a high-performance ground station graphics card, the NVIDIA RTX4050, to support complex image processing and data analysis tasks. The onboard computer utilizes the lightweight PI ZERO 2W, ensuring stable operation under low power consumption. The drone's frame has a wheelbase of 450mm, providing a stable flight platform and sufficient payload capacity. The camera features a high-definition specification of 1800 TVL, ensuring high-quality image capture. Meanwhile, the 2-axis gimbal design effectively enhances shooting stability and flexibility. The video transmission system operates at a frequency of 5.8 GHz with a power output of 1W, guaranteeing real-time and stable image data transmission. The flight control system adopts the advanced PX4, whose powerful computing capability and stability provide a solid foundation for the drone’s autonomous flight and target tracking.

4.2. Detection training

The object detection algorithm was trained using the pre-trained weights of YOLOv8m and YOLOv8x, with the number of detectable object classes adjusted to one (Person). The dataset consists of images and Person labels from the COCO dataset, as well as annotated frames containing people extracted from drone video transmissions.

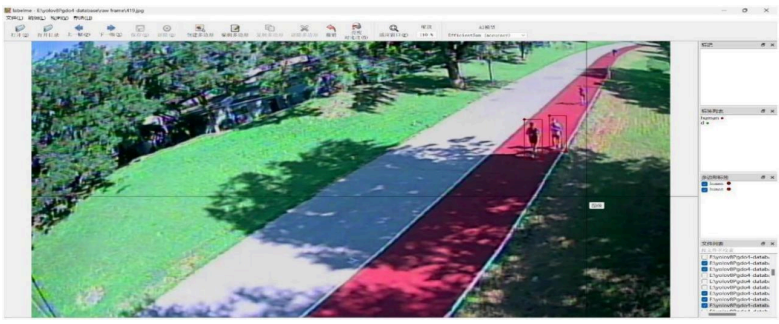


Figure 9. labelme labeled training data

DOI: 10.54254/2755-2721/2025.LD26953

Under the specified training parameters—100 training epochs, images with a resolution of 1024×60 pixels, a batch size of 21 images, and 8 threads enabled for parallel processing—the YOLO model achieved remarkable performance in object detection. As shown in Figure 8, we can clearly observe that with the increase in training epochs, both the bounding box deviation (box_loss, specifically indicated by the loss values near 1 at the top-left and bottom-left corners) and the classification deviation (cls_loss, likewise showing values close to 2 at the top-left and bottom-left corners) significantly decreased and eventually approached zero. This result strongly demonstrates that a well-trained YOLO model can achieve high-precision target detection under video transmission conditions.

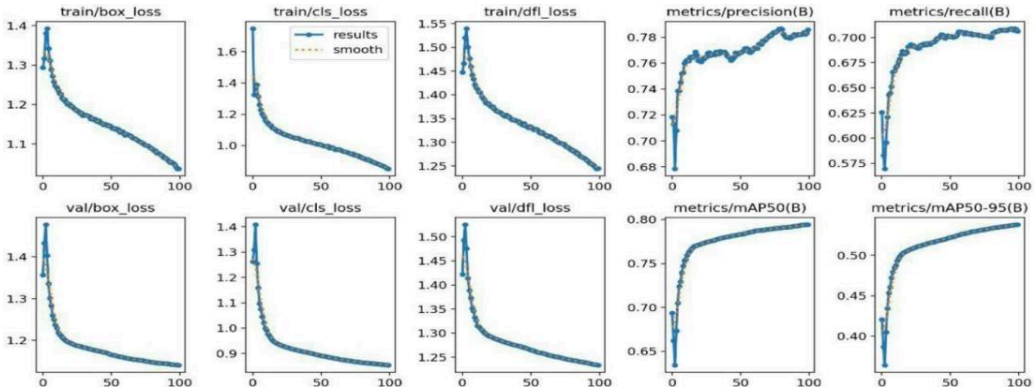


Figure 10. YOLOv8 model training loss convergence

Compared to the original artificial intelligence models, the YOLO model not only maintained stable performance in the simulated video transmission environment but also showed a significant improvement in target confidence. This means that, for the transmitted image data, the YOLO model can more accurately identify and locate target objects, thereby enhancing the reliability and practicality of object detection.

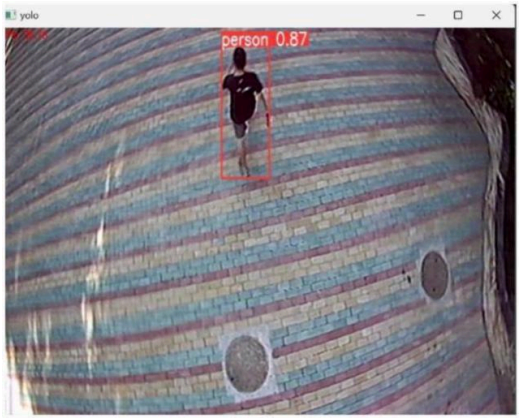


Figure 11. Target detection test on transmitted images

In addition, it is worth noting that if the amount of training data under video transmission conditions is further increased, we have reason to believe that the performance of the YOLO model will improve even further. More training data will help the model better learn the image features and target information present during the video transmission process, thereby further enhancing the accuracy and robustness of object detection. Therefore, in future work, we

DOI: 10.54254/2755-2721/2025.LD26953

may consider continuing to expand the training dataset to further optimize the YOLO model's object detection performance under video transmission conditions.

4.3. Camera calibration

During the experiment, we used a Misence brand 19x19mm 1800TVLboard camera, which greatly facilitated the experiment with its wide field of view. To convert the NTSC signal output by the camera into digital images, we used a dedicated decoding board that decoded the signal into 480×320-pixel images, which were then sent to the onboard computer for subsequent processing.

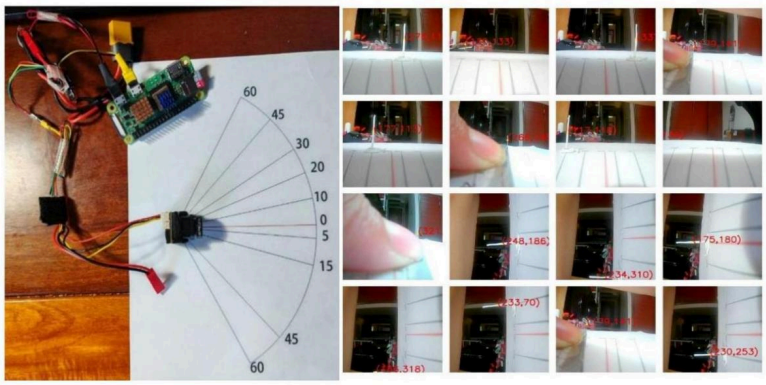


Figure 12. Camera angle calibration

In practical operation, we derived calculation formulas for the X-axis and Y-axis angles based on experimental data. For the X-axis angle, the calculation formula is:

$$Degree_x = 0.000000725564x^3 - 0.0005224061106x^2 + 0.2745658755429x - 45.8354154819057$$

For the Y-axis angle, due to a certain offset, we added a constant of 80 to the y-coordinate for correction. The calculation formula is:

$$Degree_y = 0.000000725564(y+80)^3 - 0.0005224061106(y+80)^2 + 0.2745658755429(y+80) - 45.8354154819057$$

Using these two formulas, we can accurately calculate the angles of objects in the image along the X and Y axes, providing strong support for subsequent experimental analysis.

4.4. Program design

Ground and drone-side programs were designed for seamless detection-tracking coordination.

DOI: 10.54254/2755-2721/2025.LD26953

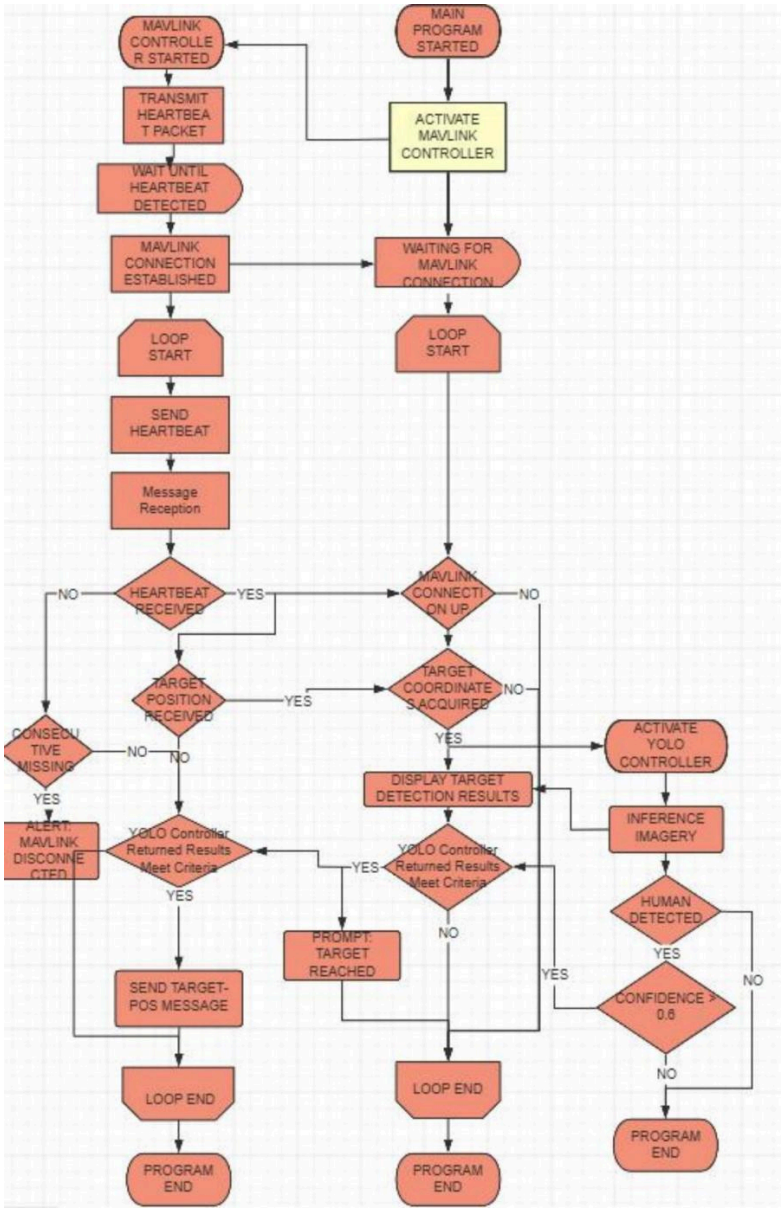


Figure 13. Ground station program design

DOI: 10.54254/2755-2721/2025.LD26953

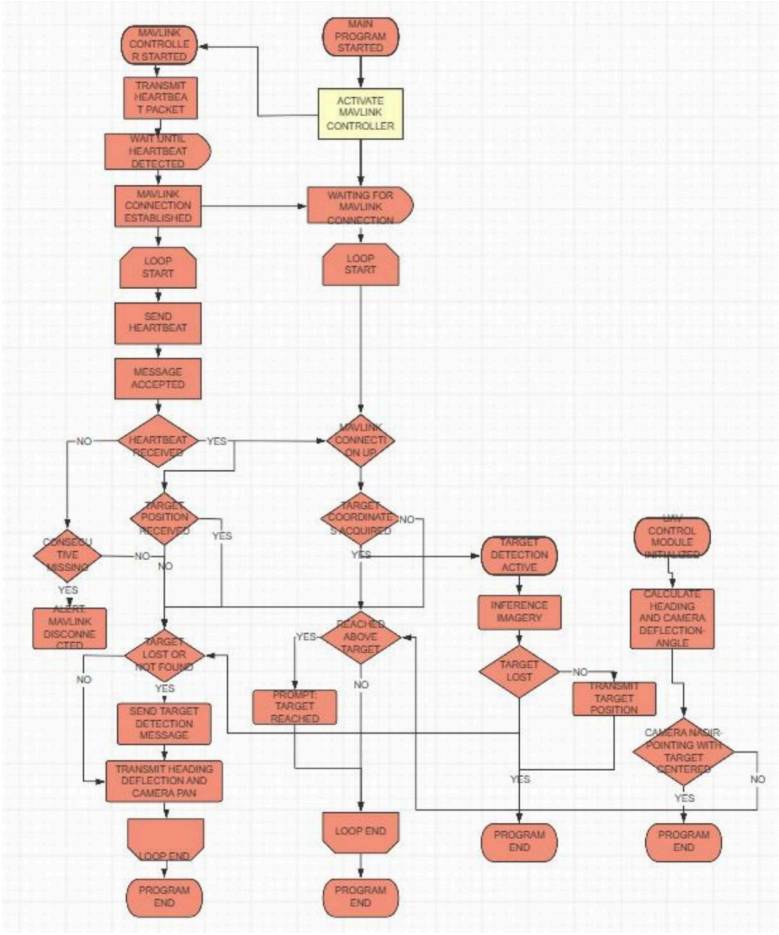


Figure 14. UAV program design

5. Flight experiments

5.1. Scenario design

The drone starts 3m from the target at 3m altitude. After takeoff, the ground station runs the object detection algorithm. Once a person is detected, the target tracking algorithm is initiated, and the drone performs visual servoing to track the moving human target until it flies directly above the target.

5.2. Process

After being programmatically controlled to take off, the drone enters a hovering state. The camera's video stream is transmitted back to the ground station, where object detection is performed on the video. The detection results are shown in Figure 15.

DOI: 10.54254/2755-2721/2025.LD26953



Figure 15. Target detection results at ground station

The detection results from the ground station are then sent to the drone as the template frame for initializing target tracking. Subsequently, the drone begins KCF (Kernelized Correlation Filter) target tracking, with the tracking results shown in Figure 16.



Figure 16. Target tracking outcome at UAV

Based on the target tracking results, visual servoing is carried out, and the results are presented in Figure 17. The first column of images shows the drone's third-person view, while the second column shows the drone's first-person view. The red circle represents the tracking center point.

DOI: 10.54254/2755-2721/2025.LD26953

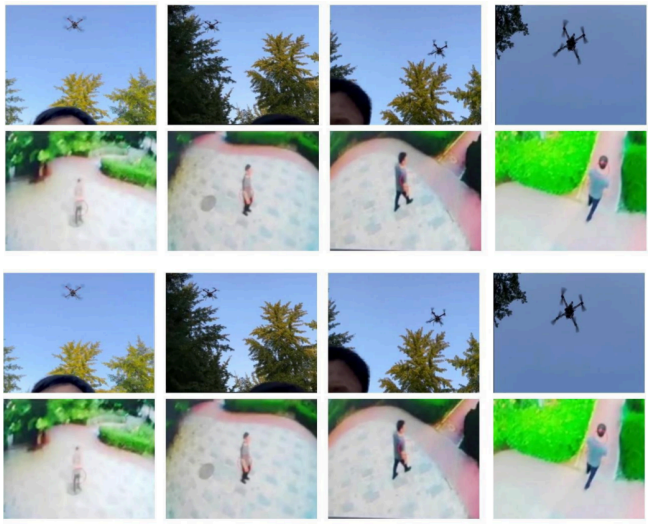


Figure 17. Visual servoing target tracking

5.3. Results and analysis

5.3.1. Target velocity analysis

The flight experiment was successfully completed as expected. The drone was able to stably follow the target at a flight altitude of 3 meters and an initial distance of 3 meters from the target.

Table 2. The experimental results are as follows

Evaluation Metric	Result
Target Detection Frame Rate	60
Target Tracking Frame Rate	5
Maximum Trackable Target Speed	3m/s

The drone can stably follow the target when it moves at a speed of 3 m/s. However, it is unable to follow the target at speeds exceeding 3 m/s, mainly due to the low frame rate of the KCF tracking algorithm, which prevents the video from tracking the target effectively. Nevertheless, in emergency rescue scenarios, targets are generally stationary or moving slowly. Therefore, considering cost factors, our system can achieve emergency rescue functionality in a low-cost manner.

5.3.2. Cost and weight analysis

“Hawkeye” Target Tracking System uses a Raspberry Pi Zero 2W as the onboard computer and accomplishes fully autonomous target identification and tracking through ground station-based object detection and drone-based target tracking.

DOI: 10.54254/2755-2721/2025.LD26953

Table 3. The table below lists the price, weight, and power consumption of commonly used onboard computers for drone object detection

Onboard Equipment	Price (¥)	Weight (g)	Computing Power (FLOPS)	Power(W)
Nvidia Jeston Nano	1100	136	472G	10
Nvidia Jeston TX2	2100	85	1.33T	15
Nvidia Xavier NX	4200	136	21T	20
Raspberry Pi4 B(2G)	400	45	13.5G	8.2
Raspberry Pi4 B(4G)	440	50	31.4G	15.2
Raspberry Pi Zero 2w	140	10	5. 10G	4.8

The Raspberry Pi Zero 2W significantly outperforms other similar devices in terms of price, power consumption, and weight. Its lower power consumption directly results in reduced heat generation, thereby further minimizing the need for heatsinks and their associated additional weight. In drone applications, when devices of similar size and power specifications are installed, the reduced weight and power consumption of the Raspberry Pi Zero 2W are especially critical for improving the drone's endurance. Therefore, using ground station-based object detection in combination with drone-based target tracking effectively reduces the cost, energy consumption, and overall weight of the drone in the process of fully autonomous target identification and tracking.

Experimental results show that high-performance GPUs significantly exceed the processing speed required by a single drone for object detection. Although current experimental conditions did not allow for testing one ground station controlling multiple drones to track targets independently, it is entirely feasible in the future to use a single high-performance ground station to direct multiple drones to track different targets. This would further reduce the cost of AI-based target tracking on drones. Compared with the Jetson Nano, our system achieves nearly the same performance while reducing costs by 87% and weight by 92%.

6. Conclusion and prospect

Currently, image transmission relies on a video transmission module. In the future, with the rapid development of communication technologies such as 5G, both the speed and stability of communication have seen significant improvements. Utilizing 5G technology for communication between the drone and the ground station could theoretically greatly enhance communication stability and extend transmission range, thereby improving the overall reliability and operational efficiency of the system. Moreover, 5G technology can deliver high-quality image transmission while maintaining low latency, which in turn enhances the accuracy of object detection at the ground station.

If 5G air-to-ground communication technology is realized, this system could also be deeply integrated with cloud computing. High-performance servers could be rented only when needed for AI computation, thereby further reducing the cost of the ground station. When not performing target tracking tasks, these computing resources could be used for other tasks, effectively avoiding idle and wasted computational power.

This paper proposes an intelligent aerial robot target tracking system for emergency rescue based on air-ground distributed computing, addressing issues such as drone operator shortages and high operational costs. The effectiveness of the system has been verified through experiments.

DOI: 10.54254/2755-2721/2025.LD26953

References

[1] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi: You Only Look Once: Unified, Real-Time Object Detection. CVPR 2016: 779-788

[2] Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. NIPS 2015: 91-99

[3] Wei Liu, Dragomir Anguelov, Dumitru Erhan, SSD: Single Shot MultiBox Detector. ECCV (1) 2016: 21-37

[4] BOLME D S, BEVERIDGE J R, DRAPER BA, et al. Visual object tracking using adaptive correlation filters [C]// Proceedings of 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Francisco, CA, USA: IEEE, 2010: 2544-2550.

[5] F.HENRIQUES J, CASEIRO R, MARTINS P, et al. Exploiting the circulant structure of tracking-by-detection with kernels [C]// FITZGIBBON AW. Proceedings of 2012 European Conference on Computer Vision Part IV. Florence, Italy: Springer, 2012: 702-715.

[6] HENRIQUES J F, CASEIRO R, MARTINS P, et al. High-Speed Tracking with Kernelized Correlation Filters [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3): 583-596.

[7] DANELLJAN M, KHAN F S, FELSBERG M, et al. Adaptive color attributes for real-time visual tracking [C]// Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, USA: IEEE, 2014: 1090-1097.

[8] LI B, YAN JJ, WU W, et al. High Performance Visual Tracking with Siamese Region Proposal Network [C]// Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA: IEEE, 2018:8971-8980.

[9] BERTINETTO L, VALMADRE J, HENRIQUES J F, et al. Fully-Convolutional Siamese Networks for Object Tracking [C]// Gang H. Proceedings of 2016 European Conference on Computer Vision Part II. Amsterdam, The Netherlands: Springer, 2016: 850-865.

[10] XU Y D, WANG Z Y, LI Z X, et al. SiamFC++: Towards Robust and Accurate Visual Tracking with Target Estimation Guidelines [C]// Proceedings of the 2020 AAAI Conference on Artificial Intelligence. New York, NY, USA: AAAI, 2020: 12549-12556.

[11] Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, Xilin Chen: Joint Feature Learning and Relation Modeling for Tracking: A One-Stream Framework. ECCV (22) 2022: 341-357

[12] Yutao Cui, Cheng Jiang, Gangshan Wu, Limin Wang: MixFormer: End-to-End Tracking With Iterative Mixed Attention. IEEE Trans. Pattern Anal. Mach.Intell. 46(6): 4129-4146

[13] DJI,DJI MINI4pro Promotional Page [EB] https: //www.dji.com/cn/mini -4

[14] TianyeAlex, tracker_kcf_ros, https: //github.com/TianyeAlex/ tracker_kcf_ros, 2016.

[15] YugAjmera, Surya Pratap Singh, Autonomous UAV-based Target

[16] Belghith, and Mohamed Khalgui, Micro Air Vehicle Link (MAVLink) in a Nutshell: A Survey, Anis Koubaa, Azza Allouch, Maram Alajlan, Yasir Javed, Abdelfettah, 2019.

[17] Mavlink Documentation, https: //mavlink.io/zh/

[18] ArduPilot Documentation, https: //ardupilots.com/ardupilot/index.html [19]. DroneKit ocumentation, https: //dronekit.io/