IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 September 2023, accepted 10 November 2023, date of publication 16 November 2023,
date of current version 21 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3333394

RESEARCH ARTICLE

UAV-Pose: A Dual Capture Network Algorithm for Low Altitude UAV Attitude Detection and Tracking

JIANG YOU[1,2], ZIXUN YE[2], JINGLIANG GU[2], AND JUNTAO PU[2]
[1]Graduate School, China Academy of Engineering Physics, Beijing 100088, China
[2]Institute of Applied Electronics, China Academy of Engineering Physics, Mianyang 621900, China

Corresponding authors: Zixun Ye (yeahzixun@foxmail.com) and Jingliang Gu (gavin51728@163.com)

This work was supported by the Innovation and Development Fund Project of China Academy of Engineering Physics under Grant CX2020033.

ABSTRACT This paper presents a low-altitude unmanned aerial vehicle (UAV) attitude detection and tracking algorithm, named UAV-Pose. In the context of low-altitude UAV countermeasure tasks, precise attitude detection and tracking are crucial for achieving laser-guided precision strikes. To meet the varying requirements during the tracking stages, this study designs two capture networks with different resolutions. Firstly, a lightweight bottleneck structure, GhostNeck, is introduced to accelerate detection speed. Secondly, a significant improvement in detection accuracy is achieved by integrating an attention mechanism and SimCC loss. Additionally, a data augmentation method is proposed to adapt to attitude detection under atmospheric turbulence. A self-collected dataset, named UAV-ADT (UAV Attitude Detection and Tracking), is constructed for training and evaluating the target detection algorithm. The algorithm is deployed using the TensorRT tool and tested on the UAV-ADT dataset, demonstrating a detection speed of 300 frames per second (FPS) with a map75 reaching 97.8% and a PCK (Percentage of Correct Keypoints) metric reaching 99.3%. Real-world field experiments further validate the accurate detection and continuous tracking of UAV attitudes, providing essential support for counter-UAV operations.

INDEX TERMS UAV pose detection, real-time tracking, maneuvering target, UAV countermeasures.

I. INTRODUCTION

In recent years, the rapid development of Unmanned Aerial Vehicle (UAV) technology has attracted widespread attention [1], making it a significant tool and asset in various fields such as military [2], civilian [3], and scientific research [4]. Particularly, low-altitude UAVs have been extensively used for tasks like reconnaissance [5], surveillance, and surveying due to their flexibility, stealthiness, and maneuverability. However, along with the proliferation of UAV technology, a series of new challenges have emerged, especially in the realm of military security.

Low-altitude UAVs, as a novel military threat, possess high speeds and stable flight characteristics, often making them difficult to be effectively identified and tracked by conventional defense mechanisms. Traditional radar systems have limitations in detecting low-altitude targets, and they are susceptible to interference in high-intensity electromagnetic environments [6]. Thus, an effective low-altitude UAV countermeasure system is urgently needed, and within such a countermeasure system, efficient and accurate methods for low-altitude UAV pose detection and tracking serve as the foundation for counteractions.

The rapid advancement of computer vision and image processing technology provides new avenues for addressing this challenge [7], [8], [9], [10], [11], [12], [13], [14]. By utilizing advanced visual sensors, image processing algorithms, and target tracking techniques, real-time and accurate detection and tracking of low-altitude UAV poses can be achieved. This not only enhances the precision of laser UAV countermeasure systems but also provides crucial intelligence support for military operations.

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Zhou.

In summary, this paper aims to propose an innovative method for low-altitude UAV pose detection and tracking, named UAV-Pose, and applies it to a laser UAV countermeasure system. By harnessing the power of computer vision and image processing technology, real-time and accurate detection and tracking of low-altitude UAV poses can be achieved, providing robust support for military security and offering extensive application prospects. Furthermore, this algorithm can also be applied to the realm of public safety, preventing potential risks posed by UAVs in scenarios such as urban patrols and monitoring of critical locations.

This paper designs a dual-capture network for anti-UAV target detection and tracking. Using capture networks with different fields of view in an anti-UAV system addresses various task requirements and scene changes, thus enhancing the performance and effectiveness of pose detection and tracking. The following are several advantages of employing capture networks with different fields of view:

1) Multi-scale processing: In low-altitude UAV countermeasure tasks, UAVs may exhibit significant variations in speed and distance, leading to scale variation issues. Using capture networks with different fields of view enables detection and tracking of targets at various scales, accommodating the diverse scale changes of targets.
2) Tracking stage requirements: Pose detection and tracking generally encompass two stages: detecting the UAV and determining its pose, followed by continuous motion tracking. The detection stage necessitates a larger field of view to capture the entire target, while a smaller field of view can be used in the tracking stage to improve computational efficiency and accuracy. Thus, employing capture networks with different fields of view better caters to the distinct requirements of these two stages.
3) Computational efficiency and speed: Networks with smaller fields of view typically possess fewer parameters and computations, allowing faster prediction generation during the tracking stage, thereby achieving real-time performance. In the detection stage, larger-field networks enhance detection accuracy by locating the target within a broader field of view.

To meet the high-speed requirements for anti-low-altitude UAV pose detection, this paper proposes the GhostNeck lightweight bottleneck structure. To counter the impact of atmospheric turbulence on detection accuracy, a data augmentation method simulating low-altitude UAV capture images under atmospheric turbulence is introduced. Furthermore, a combination of multi-scale spatial attention mechanisms and a decoupled multi-branch detection head is employed, integrating the SimCC-based approach [15] for predicting keypoints, treating keypoint localization as a classification task. In comparison to heatmap-based algorithms [16], [17], [18], [19], the SimCC-based approach achieves competitive accuracy with lower computational effort.

In conclusion, this paper deploys UAV-Pose on different inference frameworks (PyTorch, ONNX Runtime, TensorRT) and hardware (NVIDIA Jetson NX, as illustrated in the physical image in Fig.2(b)) to assess its efficiency. Experimental results demonstrate that, utilizing TensorRT with fp16 quantization, UAV-Pose achieves an impressive 300 frames per second (fps) on the NVIDIA Jetson NX, with a PCK (Percentage of Correct Keypoints) metric reaching 99.3%. This accomplishment satisfies the demanding requirements of both speed and accuracy. In real-world field experiments, UAV-Pose successfully achieves accurate detection and continuous tracking of key features on UAVs. It is noteworthy that this paper introduces, for the first time, a method for tracking specific features on UAVs, overcoming the limitation of previous approaches that solely focused on tracking the entire rectangular frame of the UAV. This innovation not only enhances the precision of UAV attitude detection but also provides robust support for the execution of counter-UAV operations.

II. RELATED WORK

Relevant research work can be classified into three main categories: those based on vision, radar, and multi-sensor data fusion:

A. VISION-BASED APPROACHES

The application of computer vision techniques in UAV tracking has been extensively studied. Isaac-Medina et al. [20] collect and integrate the aforementioned three UAV datasets [21], [22], [23], and present a benchmark performance study using state of the art four object detection [24] and tracking methods (SORT [25], DeepSORT [18]). Zhao et al. [26] recently proposed a new dataset DUT Anti-UAV containing detection and tracking subsets. Based on this dataset, the authors evaluated various detection and tracking algorithms [27], [28], and proposed a strategy to fuse detection and tracking to further improve tracking performance.

B. RADAR-BASED APPROACHES

Radar provides stable distance and velocity information, making it suitable for UAV tracking tasks. Dogru and Marques [29] proposed an active UAV detection system using millimeter wave radar, which can detect, track and pursue target UAVs. Junior and Guo [30] designed a UAV localization and interception system based on MIMO radar, which can accurately track UAVs invading the safety zone. Dogru and Marques [29] developed a dual-axis rotary tracking platform, which combines visual image processing to automatically lock and track UAVs, and can measure the flight altitude of UAVs and calculate their coordinates.

C. MULTI-SENSOR DATA FUSION APPROACHES

Integrating different types of sensor data, such as vision and radar, can leverage the strengths of each sensor to enhance the stability and robustness of UAV tracking. Shi et al. [31] proposed an ADS-ZJU system integrating multiple surveillance technologies. This system incorporates

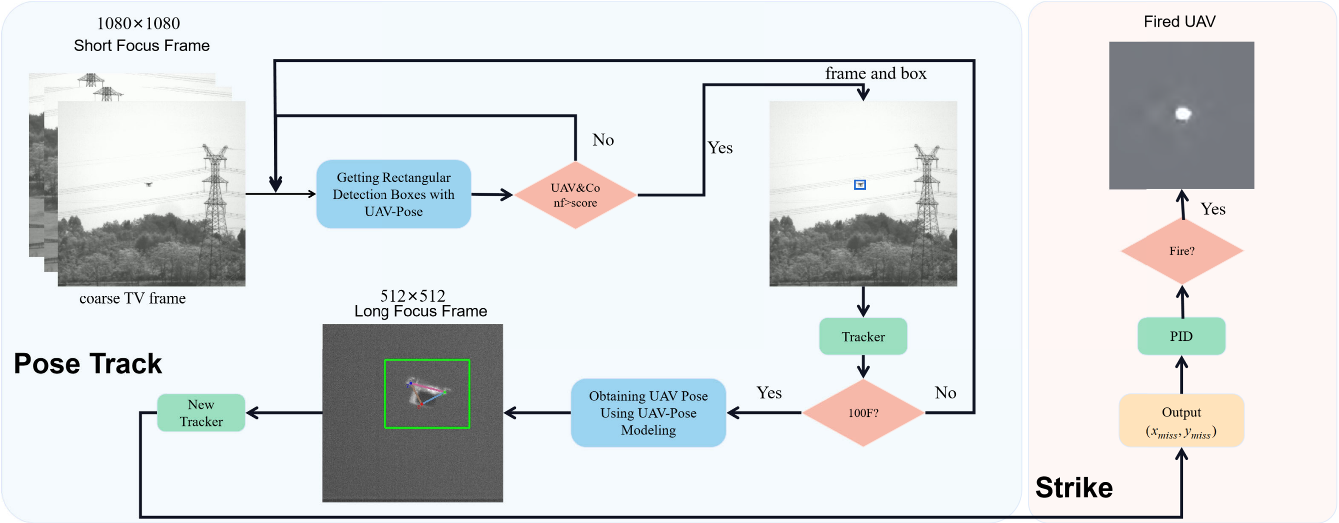


FIGURE 1. The Anti-UAV system workflow diagram.

three types of sensors: audio, video, and radio frequency. It detects UAVs by extracting audio, image, and radio frequency features and using support vector machines. The system can also conduct radio frequency interference on the detected UAVs. This solution effectively improves the detection accuracy but it has a large coverage and high cost due to the use of multiple scattered units.

III. PROPOSED METHOD

Firstly, we provided a description of the tasks within the UAV countermeasure system, outlining the components and strategies of the tracking algorithm. Subsequently, we elaborated on the model architecture and loss function design in this paper.

A. TASK DESCRIPTION

For the tracking algorithm of the UAV countermeasure system, it needs to include the capture, tracking, and strike point extraction stages. The overall framework of the tracking strategy is illustrated in Fig.1. When a UAV target enters the field of view, the following steps are executed:

1) COARSE TRACKING PHASE

The capture algorithm initially identifies the target and provides an accurate target bounding box, which is used to initialize the tracking module with the current frame;Once the tracking module is initialized, it continuously tracks the target in each frame, yielding the output target bounding box.

2) PRECISE TRACKING PHASE

To achieve target interception, strike point extraction is required. This involves the precise tracking stage: Utilizing the target bounding box from the tracking module, the corresponding Region of Interest (ROI) area in the precise camera is employed to extract the UAV's pose, facilitating strike point calculation;To establish a closed-loop control in the tracking system, the calculated strike point from the current frame, along with the designated tracking point, is used to compute the current tracking deviation. This deviation is then sent to the servo control system to enact closed-loop tracking.

3) REACQUISITION PHASE

In case of target loss during the tracking process, the detection module is reactivated to re-identify the target; To expedite the detection process, an initial detection attempt is made using the fine-capture network within a 512 × 512 area centered around the last known target position. If the target is not detected, the coarse-capture network is subsequently employed for comprehensive area detection.

During the coarse tracking phase, this paper utilizes UAV-Pose to exclusively obtain the UAV target's detection bounding box. In the precise tracking phase, along with obtaining the detection bounding box, this paper also extracts the UAV's keypoints. For quadcopter UAVs, this paper selects the two outermost rotors and the onboard camera to establish three keypoints.

B. MODEL ARCHITECTURE

The UAV pose detection and tracking algorithm has a stringent requirement for high speed performance, and a bulky model is inadequate for effectively accomplishing UAV target pose detection and tracking tasks. The overall network architecture is depicted in Fig.2(a). In this study, a novel lightweight convolutional technique called GSConv is introduced to reduce the model's parameter count and computational load while maintaining accuracy. Furthermore, a configurable Feature Pyramid Network (FPN) structure is devised by incorporating GSConv, GSBottleNeck, and the Multi-Scale Spatial Attention (MSSA) mechanism.

The model structure of UAV-Pose comprises three main components: the backbone feature extraction network, the

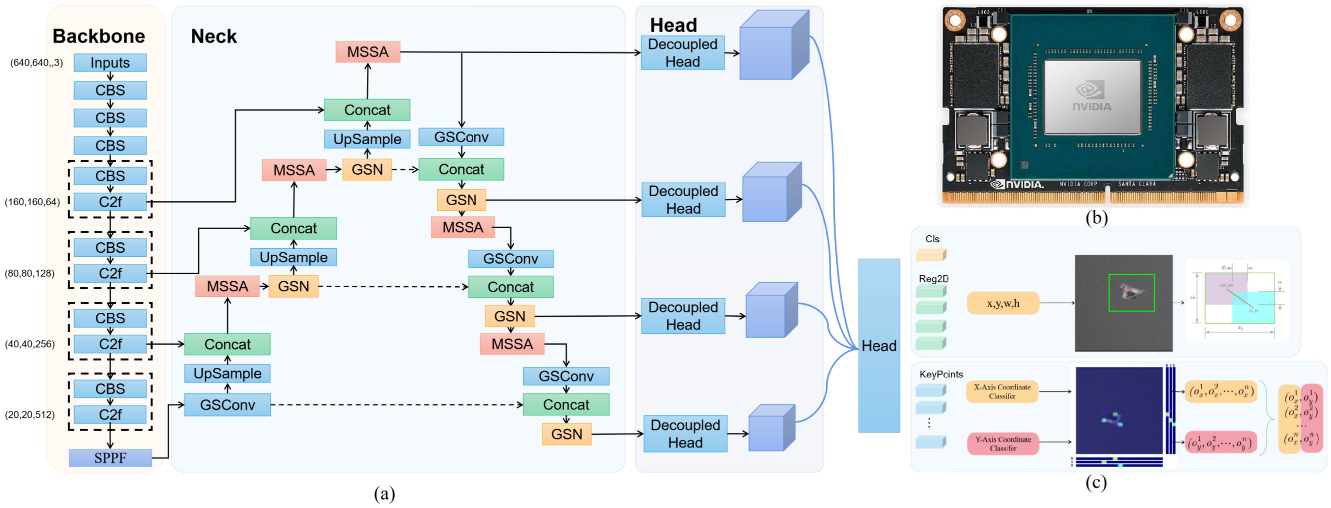


FIGURE 2. Overview of the system components. (a) Network architecture, (b) Physical image of NVIDIA Jetson NX, and (c) Thumbnail structure of the multi-branch decoupled detection head.

multi-scale feature pyramid, and the multi-branch decoupled detection heads. During the coarse tracking phase of UAV targets, a 4-layer FPN structure and Multi-Scale Spatial Attention (MSSA) mechanism are devised to enhance feature saliency and detection accuracy for targets of various scales. In the fine tracking phase of UAV targets, a 2-layer FPN structure is designed to accelerate the model's detection speed.

1) MULTI-SCALE FEATURE PYRAMIDS

The main network is primarily responsible for feature extraction from images, as illustrated in the diagram below. It mainly consists of CBS convolutional layers, C2f convolutional modules, and an SPPF module. The CBS convolutional layer includes convolutional layers, Batch Normalization (BN) layers, and SiLU activation functions, aimed at extracting features of different scales from the image. The C2f convolutional module is an efficient aggregation network that alters computational blocks based on Conv while maintaining the original transition layer structure. It enhances the network's learning capabilities using arithmetic techniques without disrupting the existing gradient pathways, and guides different feature groups to learn more diversified features. The SPPF module comprises spatial pyramid pooling layers with four different scale sizes of maximum pooling, adapting to different target resolutions to differentiate between various target sizes. In this paper, the existing backbone feature network is enhanced by utilizing larger scales, specifically $P_2 \in \mathbb{R}^{160\times160\times64}$, for detecting small UAV targets in a wider field of view.

The MobileNets [32], [33], [34], series of lightweight convolutional neural networks are designed for embedded and mobile devices. Experiments have indicated that the feature maps generated by these networks exhibit a certain degree of redundancy. Additionally, the depthwise separable convolutions they use contain a significant amount of convolution in the point convolution part, leading to computational complexity that can still be optimized. GhostNet [35] introduces GSConv to replace the point convolution in depthwise separable convolutions, reducing computational complexity while maintaining recognition performance.

Given an input feature map $G \in \mathbb{R}^{C\times H\times W}$, after convolution with multiple kernels $K \in \mathbb{R}^{C\times K\times K}$, an intermediate feature layer $G' \in \mathbb{R}^{C/s\times H'\times W'}$ is obtained. Here, "s" is a hyperparameter that determines the extent of channel compression for this convolution. Subsequently, deep depthwise separable convolutions are applied to G', and the result is concatenated with G' itself to yield the output feature map $G'' \in \mathbb{R}^{C'\times H'\times W'}$. The FLOPs calculation formula for GSConv can be expressed as:

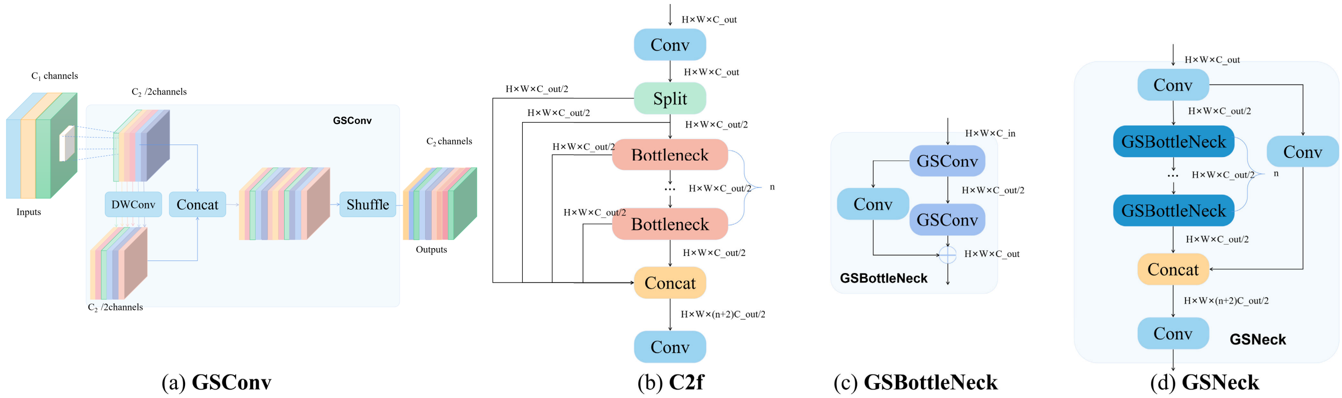
$$\begin{aligned} FLOPs_{GSConv} &= FLOPs_{cv1} + FLOPs_{cv2} \\ FLOPs_{cv1} &= 2 * H_{out} * W_{out} * C_{in} * (C_{out}/2) * K^2/S \\ FLOPs_{cv2} &= 2 * H_{out} * W_{out} * (C_{out}/2) * (C_{out}/2) * K^2/S \end{aligned} \tag{1}$$

Here, $FLOPs_{cv1}$ and $FLOPs_{cv2}$ represent the FLOPs for the first and second convolutional layers, respectively. H_{out} and W_{out} are the height and width of the output feature map, C_{in} and C_{out} are the input and output channel numbers, K is the kernel size, and S is the stride. For the cv1 and cv2 layers in GSConv, we can substitute their parameters into the above formula to obtain:

Then, substituting $FLOPs_{cv1}$ and $FLOPs_{cv2}$ into the formula for $FLOPs_{GSConv}$, we can derive the FLOPs for GSConv. For a general Conv block, its FLOPs calculation formula is:

$$FLOPs_{Conv} = 2 * H_{out} * W_{out} * C_{in} * C_{out} * K^2/S \tag{2}$$

Comparing GSConv with a standard Conv block, it can be observed that the FLOPs calculation formula for GSConv is structurally similar to that of a standard Conv block.



(a) **GSConv** (b) **C2f** (c) **GSBottleNeck** (d) **GSNeck**

FIGURE 3. Diagrams illustrating various network structures in the current paper: (a) GSConv Structure, (b) C2f network architecture, (c) GSBottleNeck network structure, and (d) GSNeck network structure.

However, since GSConv contains two convolutional layers, and the input and output channel numbers for the second convolutional layer are both half of the original channel number, the FLOPs for GSConv will be smaller than that of a standard Conv block.

Subsequently, the combination of GSConv and deep depthwise separable convolutions forms the residual network GSBottleNeck, which mitigates the issue of gradient vanishing in deep networks.

The original C2f structure is depicted in the left diagram, where C2f employs the Split method to introduce more residual connections, thus leading to a richer gradient flow. However, this also results in an increase in the number of parameters and computational complexity. This paper introduces GSConv and designs the GSBottleNeck structure, which serves as the basis for GSNeck, replacing the original C2f structure. This change retains the adaptability of GSBottleNeck. To accelerate the calculation of predictions, the input images in the CNN must undergo a similar transformation process within the Backbone: gradually transmitting spatial information to the channels. Moreover, each spatial compression (width and height) and channel expansion of the feature map can lead to partial loss of semantic information. Dense convolutional computation maximally preserves hidden connections between each channel, while sparse convolution completely severs these connections. GSNeck aims to retain these connections as much as possible.

For GSBottleneck, its FLOPs can be calculated using the following formula:

$$\begin{aligned}&FLOPs_{GSB}\\&\quad = 2*(2*H_{out}*W_{out}*C_{in}*(C_{out}/2)*K^2/S\\&\qquad + 2*H_{out}*W_{out}*(C_{out}/2)*(C_{out}/2)*K^2/S) \qquad (3)\end{aligned}$$

Finally, substituting $FLOPs_{cv1}$, $FLOPs_{cv2}$, and $FLOPs_{GSB}$ into the formula for $FLOPs_{GSNeck}$, we obtain:

$$\begin{aligned}&FLOPs_{GSNeck}\\&\quad = 2*H_{out}*W_{out}*C_{in}*(C_{out}/2)*K^2/S\\&\qquad + 2*H_{out}*W_{out}*(C_{out}/2)*(C_{out}/2)*K^2/S\\&\qquad + n*(2*(2*H_{out}*W_{out}*C_{in}*(C_{out}/2)*K^2/S\\&\qquad + 2*H_{out}*W_{out}*(C_{out}/2)*(C_{out}/2)*K^2/S)) \qquad (4)\end{aligned}$$

On the other hand, the FLOPs calculation formula for the C2f module is:

$$\begin{aligned}&FLOPs_{C2f}\\&\quad = 2*H_{out}*W_{out}*C_{in}*(2*C_{out}*e)*K^2/S\\&\qquad + 2*H_{out}*W_{out}*((2+n)*C_{out}*e)*C_{out}*K^2/S\\&\qquad + n*(2*H_{out}*W_{out}*C_{in}*(C_{out}*e)*K^2/S\\&\qquad + 2*H_{out}*W_{out}*(C_{out}*e)*C_{out}*K^2/S) \qquad (5)\end{aligned}$$

From this analysis, it is evident that using GSNeck in the network's FPN structure can significantly reduce computational complexity.

2) MULTI-SCALE SPATIAL ATTENTION MECHANISM

FPN (Feature Pyramid Network) is a feature pyramid network used for multi-scale object detection. It constructs multi-scale feature maps by adding extra lateral connections in the Backbone network. These lateral connections extract features from different levels of the Backbone feature maps and merge them into the feature maps of the previous layer. This way, FPN can obtain rich semantic information at different scales and provide feature maps with different resolutions.

Attention mechanisms are widely used to improve the performance of deep learning models by selectively focusing on relevant information and suppressing irrelevant or noisy information [36], [37], [38]. However, these attention mechanisms overlook the scale information of feature maps. In the original FPN structure, when fusing features from different scales, the features are often adjusted to the same scale through upsampling or downsampling and directly concatenated along the channel dimension. This paper proposes a Multi-Scale Spatial Attention mechanism (MSSA), which adaptively adjusts the weights of features from P2 to P5. It increases the weights of scale features that are more beneficial for the recognition task while suppressing the weights of other scale features. Spatially, the model can focus on image textures and contextual information that are

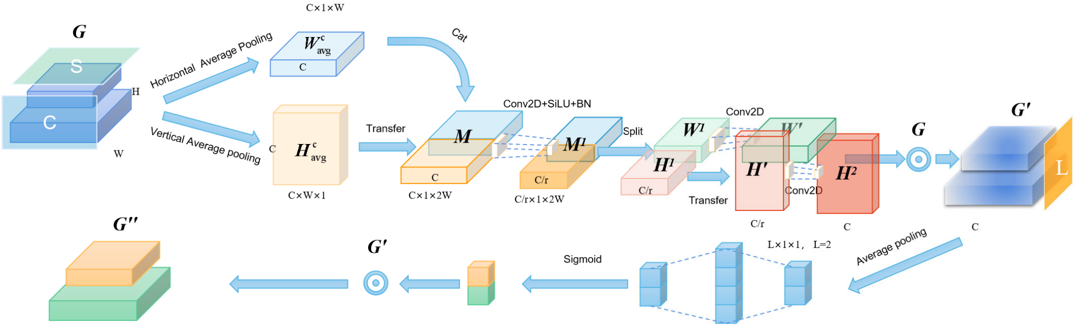


FIGURE 4. The MSSA network architecture.

advantageous for the recognition task. The structure of MSSA is shown in the Fig.4.

(1) First, concatenate the compacted features. Since the dimensions of features $W^{\mathrm{c}}_{\mathrm{avg}}$ and $H^{\mathrm{c}}_{\mathrm{avg}}$ do not match, the width and height dimensions of feature $H^{\mathrm{c}}_{\mathrm{avg}}$ need to be transposed before concatenating it with $H^{\mathrm{c}}_{\mathrm{avg}}$ to obtain the feature map M.

(2) Set a hyperparameter r such that M is passed through a 2D convolution to obtain the feature map M^1, where the number of channels changes from c to $\frac{c}{r}$ In this paper, r is set, and the number of channels in M^1 should not be less than 8. Then, insert a BN layer and a SiLU activation function to obtain the feature map M^2. At this stage, M^2 incorporates both the feature compaction of the input feature G along the x-axis and y-axis, allowing spatial information of the input feature G to interact.

The mixed spatial information in M^2 is then divided, transposed, and passed through another 2D convolution to restore the channel number to c, resulting in W' and H'. These two feature maps represent the spatial weights. Finally, element-wise multiplication is performed between W', H', and the corresponding elements of matrix G to obtain G'. This process combines the spatial weights in the input feature map, where the spatial weights beneficial for the recognition task are increased. In the diagram, $\odot$ represents element-wise multiplication between matrices. Thus, the spatial information S and channel information C are adaptively adjusted. This process can be represented as:

$$\begin{aligned} G' &= \sigma(f \otimes ([AvgPoolW(\mathbf{G}); AvgPoolH(\mathbf{G})])) \\ &= \sigma\left(f \otimes \left(\left[W^c_{\mathbf{avg}}; H^c_{avg}\right]\right)\right) \end{aligned} \tag{6}$$

where σ represents the sigmoid function, $\otimes$ represents the convolution process, and f is a convolution kernel.

Next, the feature map G' is adaptively average-pooled at scale L to obtain a compacted feature of size 2×1×w. Then, an MLP is used to adjust the compacted feature, and a sigmoid function is applied to obtain the activated weights. Finally, the weights are added to the original feature map G' to obtain the feature map G''. This process adapts the weights at scale L. It can be represented as:

$$\begin{aligned} G'' &= \sigma(f \otimes ([AvgPoolL(\mathbf{G}')])) \\ &= \sigma(f \otimes ([W^l_{\mathrm{avg}}])) \end{aligned} \tag{7}$$

3) MULTI-BRANCH DECOUPLING DETECTION HEAD

In object detection, the conflict between classification and regression tasks is a common issue. Therefore, the decoupling of classification and localization heads has been widely applied in both one-stage and two-stage detection approaches. However, with the evolution of the YOLO series backbone and feature pyramid, the detection head remains coupled. In this paper, a multi-branch decoupled detection head is introduced in the detection stage, as shown in Fig.5. The model's output includes regression of 2D bounding boxes, object classification, and classification of the x and y coordinates of keypoints.

SimCC [15] introduces an innovative approach that represents keypoints as classifications within subpixel boxes for both horizontal and vertical coordinates. This approach offers several benefits. Firstly, SimCC is not reliant on high-resolution heatmaps, eliminating the need for a bulky architecture or expensive upsampling layers. Secondly, SimCC conducts classification on the flattened final feature map, avoiding global pooling and preserving spatial information. These attributes render SimCC a compelling choice for constructing lightweight pose estimation models. In this study, we further harness the coordinate classification scheme to optimize model architecture and training strategies.

C. LOSS FUNCTION

For the prediction of two-dimensional bounding boxes, this paper employs the Weighted IoU (WIoU) loss, where IoU represents the traditional Intersection over Union. For an anchor box $\vec{B} = [x, y, w, h]$ and the ground truth box $\vec{B}_{gt} = [x_{gt}, y_{gt}, w_{gt}, h_{gt}]$, W and H denote the width and height of the overlapping region between the boundaries. The specific formula is as follows:

$$\mathcal{L}_{WIou} = \exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{(W_g^2 + H_g^2)^*}\right)\mathcal{L}_{IoU} \tag{8}$$

In the initial SimCC, keypoint localization is treated as a classification problem. The core idea is to divide the horizontal and vertical axes into equi-width bins and discretize continuous coordinates into integral bin labels. Then, the model is trained to predict the bin in which the keypoint resides. By using a large number of bins, quantization errors can be simplified to subpixel levels. Due to this novel

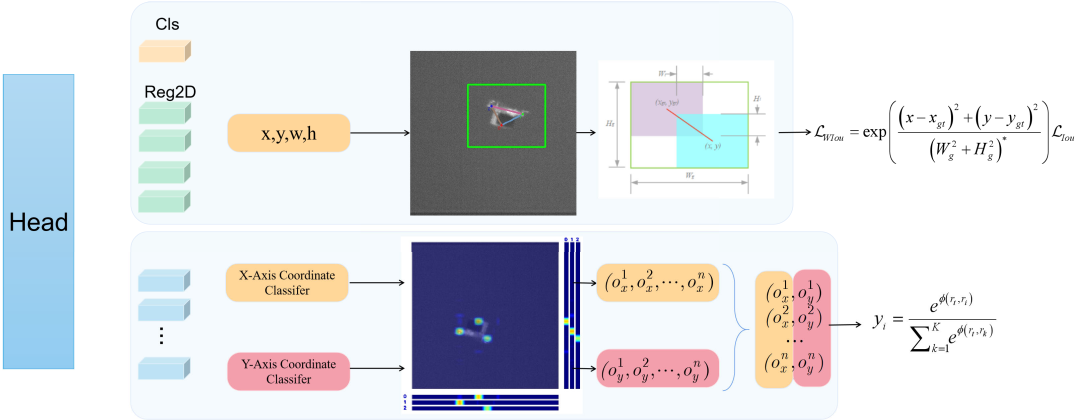

Head
Cls
Reg2D
x,y,w,h
$\mathcal{L}_{WIoU} = \exp\left(\frac{(x-x_{gt})^2+(y-y_{gt})^2}{(W_g^2+H_g^2)^*}\right)\mathcal{L}_{IoU}$
X-Axis Coordinate Classifer
Y-Axis Coordinate Classifer
$(o_x^1, o_x^2, \cdots, o_x^n)$
$(o_y^1, o_y^2, \cdots, o_y^n)$
(o_x^1, o_y^1)
(o_x^2, o_y^2)
(o_x^n, o_y^n)
$y_i = \frac{e^{\phi(r_t, r_i)}}{\sum_{k=1}^{K} e^{\phi(r_t, r_k)}}$

FIGURE 5. The multi-branch decoupled head structure.

formulation, SimCC has a very straightforward structure, utilizing a 1 × 1 convolution layer for transformation.

Regarding the loss function, we treat coordinate classification as an ordinal regression task and follow the soft label encoding proposed in SORD:

$$y_i = \frac{e^{\phi(r_t, r_i)}}{\sum_{k=1}^{K} e^{\phi(r_t, r_k)}} \tag{9}$$

Here, $\phi(r_t, r_i)$ is the chosen metric loss function, used to penalize the distance between the true metric value r_t and the ranking $r_i \in Y$. In this work, we adopt the unnormalized Gaussian distribution as the inter-class distance metric:

$$y_i = \frac{e^{\phi(r_t, r_i)/\tau}}{\sum_{k=1}^{K} e^{\phi(r_t, r_l)/\tau}} \tag{10}$$

IV. EXPERIMENTS

In this section, the paper elaborates on the details of model training, the effectiveness of data augmentations, conducts ablation experiments, and analyzes the experimental results. Additionally, in the final subsection, UAV-Pose is quantitatively compared with some state-of-the-art pose detection algorithms. A NVIDIA GeForce RTX 3080 Ti GPU is employed, with a batch size of 4 and a total of 300 epochs for model training. The SGD optimizer is used with betas set to (0.9, 0.009), an initial learning rate of 0.01, and weight decay of 0.0005.

Furthermore, all algorithms are deployed on the NVIDIA Jetson Xavier NX platform for experimentation. The CPU is NVIDIA Carmel ARM® 8.2 64-bit, memory is 8 GB, and GPU is NVIDIA Volta™GPU. The runtime libraries include CUDA 10.1, cuDNN 8.3.1, and TensorRT 7.0.0.

A. DATASET

This paper independently collected and constructed a dataset named UAV-ADT (UAV Attitude Detection and Tracking) for training and evaluating target detection algorithms. The dataset consists of two sub-datasets: CTD (Coarse Tracking Dataset) and PTD (Precise Tracking Dataset).

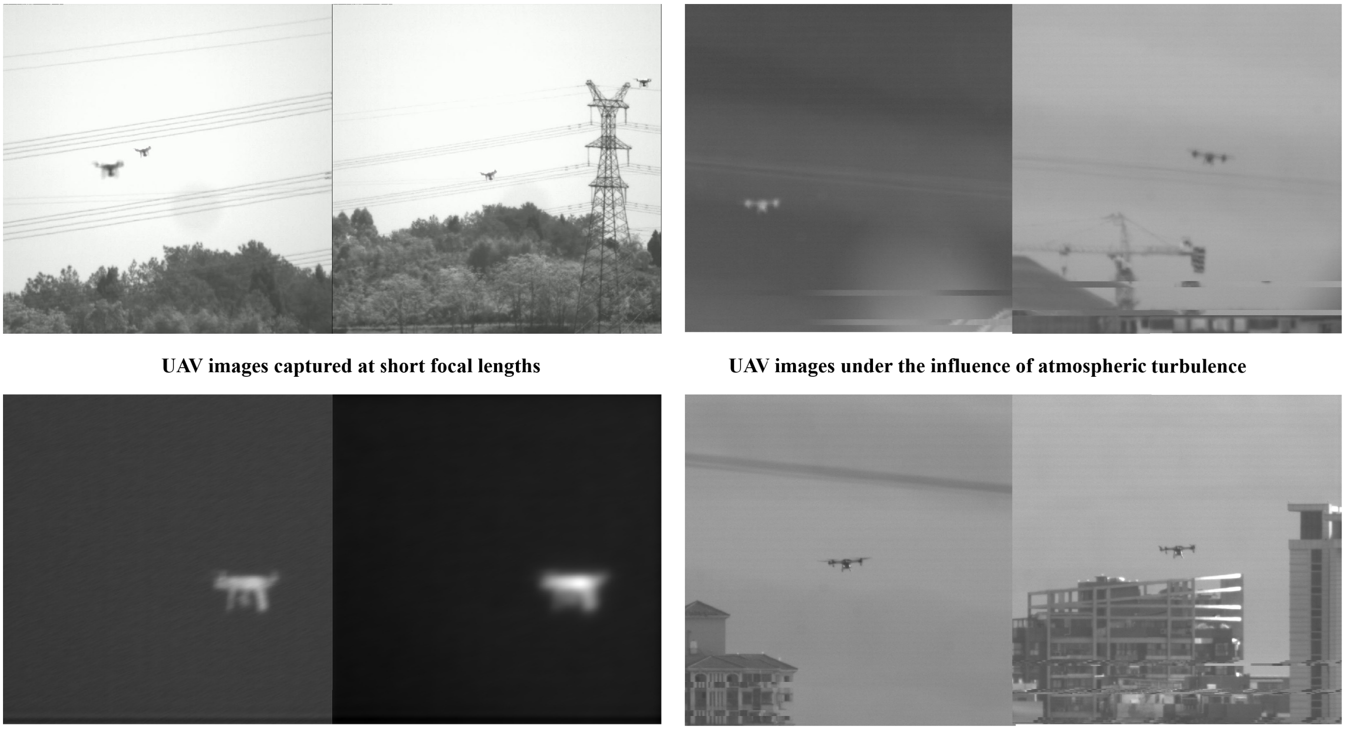
The CTD dataset, utilized for UAV target detection, comprises various models of rotary-wing unmanned aerial vehicles, including DJI Phantom 3, DJI Phantom 4, DJI Mavic Air, and fixed-wing drones. It encompasses diverse backgrounds such as open sky, forests, and urban settings, featuring UAV poses like level flight, maneuvers, and rapid ascent/descent. The dataset consists of 22,608 original images. Manual annotation was performed using the labelme image annotation tool following the VOC dataset format. To enhance the detection accuracy of the research methodology, data augmentation techniques such as horizontal mirroring, slight rotation, blur, and randomly adding birds were applied to the original UAV images.

The PTD corresponds to the dataset captured by a corresponding telephoto lens, annotated with key points, and used for training and evaluating attitude detection algorithms. The algorithm simulated atmospheric turbulence to augment the UAV images. This dataset comprises 4,743 images. The final dataset is divided into training and validation sets with an 80:20 ratio. Sample images and visualizations of data augmentation effects are presented in Fig.6.

B. REAL-TIME DATA AUGMENTATION

In 2D object detection, data augmentation is a commonly used technique to increase the diversity and richness of training data, thereby enhancing the model's robustness and generalization capability. When capturing images of low-altitude UAVs using a camera, they are susceptible to the effects of atmospheric turbulence. Atmospheric turbulence refers to irregular and random airflow movements in the atmosphere, resulting in momentary changes in air velocity, direction, and density. These turbulent phenomena can occur at various scales, ranging from microscopic molecular scales to macroscopic atmospheric scales. The presence of turbulence in the air significantly impacts applications such as UAV photography and imaging, as it can lead to image blurring, shaking, and distortion, thereby affecting image quality.

To mitigate the impact of atmospheric turbulence on imaging, this paper devised a real-time data augmentation technique to simulate UAV imaging under atmospheric turbulence conditions. Additionally, other augmentation techniques were



UAV images captured at short focal lengths

UAV images under the influence of atmospheric turbulence

Modeling the effects of atmospheric turbulence using data enhancement

UAV images without the influence of atmospheric turbulence

FIGURE 6. The dataset samples and data augmentation effects.

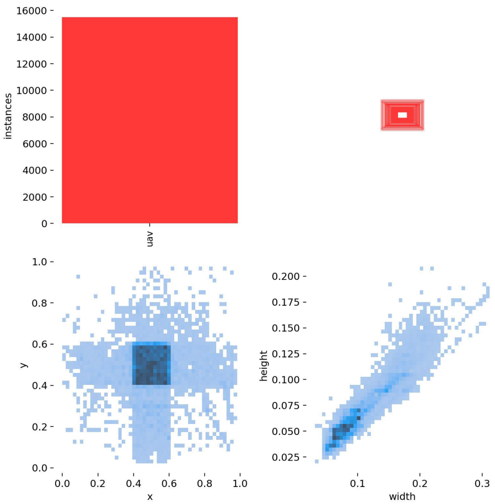


FIGURE 7. The statistical analysis of the category distribution in the test set is as follows.

applied to the original UAV images, including horizontal mirroring, slight rotations, blurring, and the random addition of birds. The pseudo-code implementation for simulating atmospheric turbulence is provided below:

Algorithm 1 Algorithm for Simulating Images

```
Require: $N, M, d, \lambda, Ii$
Initialize parameters
Create coordinate grids $u, v, fu, fv, x, y$
Generate pupil function $pupil$
Create simulation object $calc$
for $i = 1$ to $num$ do
    Load image $A$
    Resize $A$ to $N \times N$
    Create extended target $Ii$
    Generate turbulence phase screen $phz$
    Calculate aberrated PSF $ima\_err$ and phase error $\Phi$
    Calculate aberrated image $image\_err$
    Save $\Phi$ and images
end for
```

C. EXPERIMENTS AND ANALYSIS

PCK (Percentage of Correct Keypoints) is a commonly used evaluation metric in pose estimation, aimed at assessing the accuracy of keypoint (joint) localization. PCK measures the distance between the estimated keypoint positions and the ground truth keypoint positions, determining whether the keypoints are accurately localized within a certain threshold.

The computation of PCK involves comparing the Euclidean distance between the estimated keypoint position and the ground truth position with a predefined threshold. If the Euclidean distance is less than the threshold, the keypoint is considered correctly located; otherwise, it is considered inaccurately located. Ultimately, PCK represents the percentage of correctly located keypoints. In this study, PCK is primarily utilized to quantitatively compare the performance of keypoint detection.

TABLE 1. The comparison of the experimental and ablation study.

Model	Datasets	Task	map50	map75	PCK	Params(M)	GFlops
YOLOXm	*Coarse Tracking Dataset*	*Detection*	*86.2*	*57.8*	-	*25.3*	*73.8*
YOLOv8m			*86.7*	*58.2*	-	*25.9*	*78.9*
Ours+GSN			*80.4*	*45.3*	-	*11.73*	*28.2*
Ours+4Head			*84.8*	*54.2*	-	*11.74*	*53.9*
Ours+MSSA			*85.8*	*54.7*	-	*11.76*	*54.1*
YOLOXs-Pose	*Precise Tracking Dataset*	*Pose*	-	*0.951*	*0.964*	*9.0*	*26.8*
YOLOv8s-Pose			-	*0.955*	*0.969*	*11.42*	*29.6*
RTMDet+SKPS			-	*0.988*	*0.991*	*28.65*	*40.88*
Ours+GSN			-	*0.964*	*0.977*	*11.73*	*28.2*
Ours+MSSA			-	*0.983*	*0.981*	*11.74*	*28.4*
Ours+2Head			-	*0.978*	*0.974*	*11.13*	*22.5*
Ours+SimCC			-	*0.978*	*0.993*	*11.13*	*22.5*

The specific formula for calculating PCK is as follows:

$$PCK = \frac{\text{Number of Correctly Located Keypoints}}{\text{Total Number of Keypoints}} \times 100\% \tag{11}$$

In the experiments, this study compares the performance of object detection with YOLOX, YOLOv7, and YOLOv8 on the coarse tracking dataset. Similarly, for the fine tracking dataset, keypoint detection performance is compared with YOLOx-Pose, YOLOv8-Pose, and SKPS. The models are denoted as follows: Ours +GSN: YOLOv8-based model with GhostConv and GhostNeck bottleneck structures; Ours +4Head: Improvement upon the previous model with four detection heads; Ours +MSSA: Incorporation of multi-scale spatial attention mechanism in the previous model; Ours+2Head: Model modified back to two detection heads; Ours+SimCC: Usage of decoupled detection heads and introduction of SimCC loss function for UAV pose detection.

These models are evaluated and compared based on their respective performance metrics using PCK in the context of keypoint detection for UAV pose estimation.

The analysis of the experimental results reveals that, in the case of object detection on the coarse tracking dataset, the Ours+GSN model significantly reduces both the parameter count and computational workload by nearly half. This reduction is accompanied by a decrease in map50 and map75 values. However, through modifications involving four detection heads and the introduction of the multi-scale spatial attention (MSSA) mechanism, the model's parameter count remains almost unchanged, and the computational workload increases to 54.1. A comprehensive comparison of map50 and map75 results indicates that the proposed Ours+MSSA model achieves a remarkable 54% reduction in parameter count and 27% reduction in computational workload while only experiencing a marginal 0.9% drop in accuracy. This clearly demonstrates that the integration of the multi-scale spatial attention mechanism (MSSA) and the use of four detection heads significantly enhance the accuracy of object detection. When it comes to keypoint pose estimation on the fine tracking dataset, the introduction of GSNeck and GSConv, along with the employment of a dual-detection-head approach, notably reduces the model's parameter count and computational workload. However, with subsequent incorporation of the multi-scale spatial attention mechanism (MSSA) and SimCC loss function, there is a substantial improvement in PCK values. In comparison to YOLOXs-Pose and YOLOv8s-Pose, the final model manages to enhance map75 by 2.7 percentage points and PCK by 2.4 percentage points while reducing computational workload by 24%. This underscores the positive effects of both the multi-scale spatial attention mechanism (MSSA) and the SimCC loss function in enhancing keypoint localization accuracy.

TABLE 2. The performance comparison of different detection algorithms.

Algorithm	capture accuracy	miss distance	tracking speed
	(%)	(pixel)	(fps)
YOLOv8+CSRT	93.2	1.956	75
YOLOv8+DeepSort	93.2	1.872	72
YOLOv8-Pose+DeepSort	97.8	0.783	68
YOLOv8-Pose+OcSort	97.8	0.767	72
UAV-Pose+OcSort	98.6	0.590	82

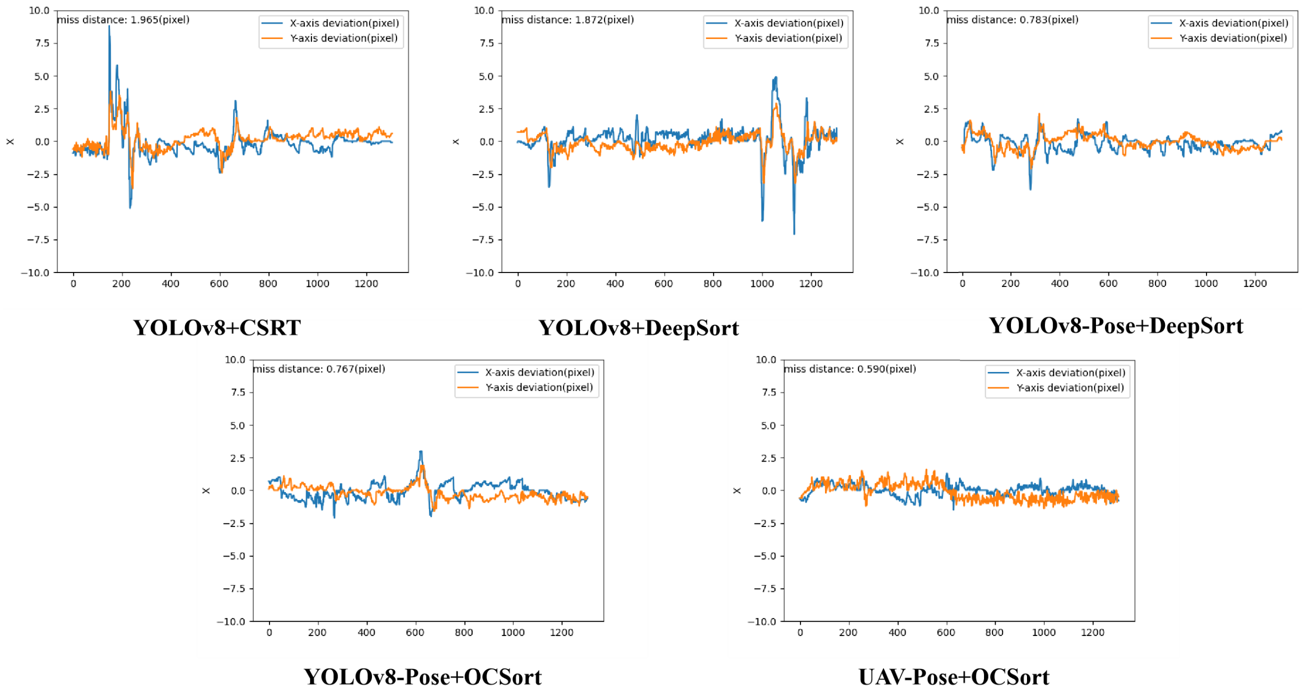
D. EXPERIMENT ON LOW-ALTITUDE UAV DETECTION AND TRACKING IN COMPLEX BACKGROUND

The entire set of object detection and tracking algorithms proposed in this article were applied in experiments within an UAV countermeasure system. In each test, the UAV maintained consistent altitude, speed, attitude, and direction of movement. The DJI Phantom 4 served as the test UAV, with experiments conducted at a distance of 1 km from the UAV. The camera resolution was set at 1024×1024, with a data acquisition frequency of 85 frames per second (fps). A continuous data stream of 1300 frames was collected, and the overall tracking performance was evaluated using the tracking miss distance metric, defined by the formula:

$$M = \sqrt{\frac{\sum_c (x_c - x_o)^2}{N}} \tag{12}$$

where: M represents the tracking miss distance; x_c is the current frame's estimated target position; x_o is the target's desired closed-loop position; N is the total number of tracking frames.

The experimental results are presented in the table2 and Fig.8. The test results demonstrate that the improved



YOLOv8+CSRT

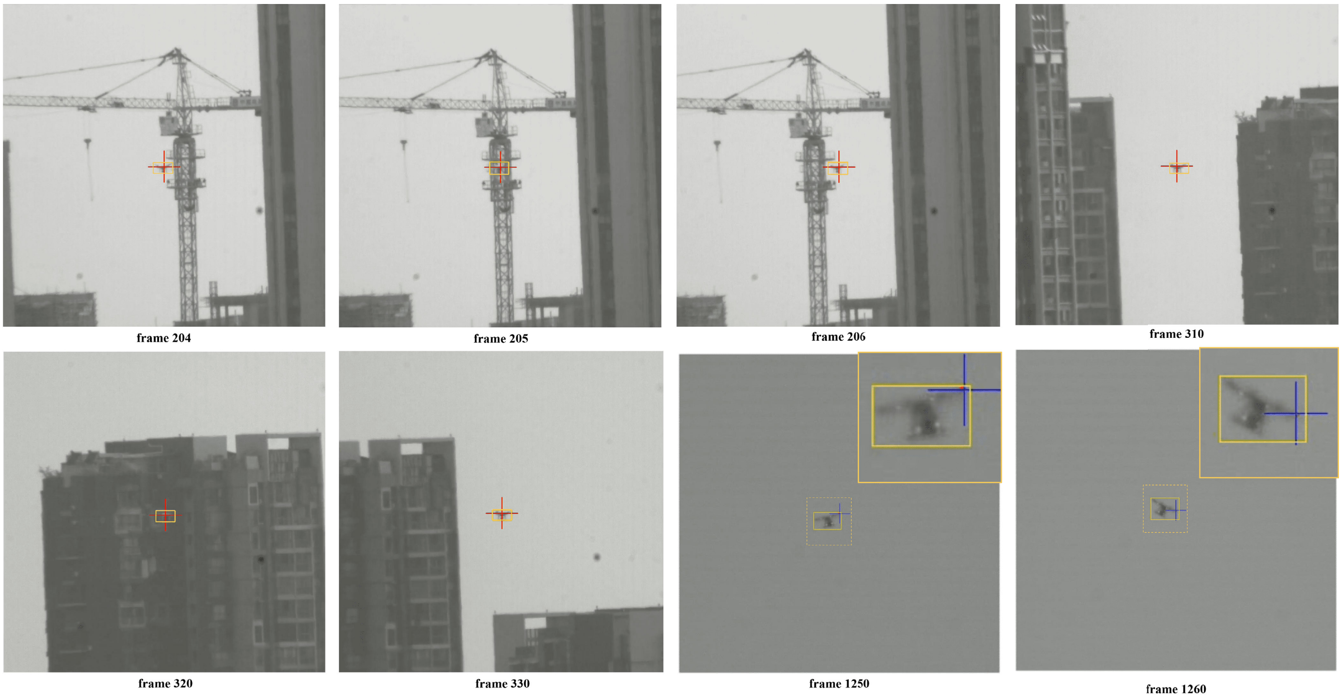
YOLOv8+DeepSort

YOLOv8-Pose+DeepSort

YOLOv8-Pose+OCSort

UAV-Pose+OCSort

FIGURE 8. Comparison of miss distance: The horizontal axis represents the video frame numbers, while the vertical axis depicts the offset of the frame's target position and the closed-loop target position on the x-axis and y-axis, respectively.



frame 204

frame 205

frame 206

frame 310

frame 320

frame 330

frame 1250

frame 1260

FIGURE 9. Visualization of algorithm tracking UAV feature points effect after optimization.

object detection and tracking algorithm presented in this paper exhibit high accuracy and stability. They also meet the real-time requirements and are capable of tracking low-altitude UAV targets against complex backgrounds. In field experiments, the algorithm not only accurately detected UAV targets at a 1 km distance but also maintained stable and continuous tracking. Ultimately, the tracking miss distance remained within the requirements of the countermeasure system, leading to the successful interception of the target UAV.

E. RESULTS ANALYSIS

The comprehensive UAV target detection and tracking algorithm proposed in this paper achieved a real-time tracking speed of 75 frames per second on the testing platform. The success rate of target capture using the proposed algorithm reached 98.4%, exhibiting a notable improvement of 10.6% compared to the optimized baseline. The algorithm effectively detected small targets at a distance of 1 km, mitigating the challenges of missed detections and false positives in complex backgrounds.

The practical experimentation confirmed that the algorithm's tracking miss distance remained within 1.2 pixels, meeting the engagement requirements of UAV countermeasure systems. The algorithm successfully demonstrated real-time detection and tracking of UAV targets in complex scenarios, and this achievement was visually verified through the visualization of tracking results for specific frames (204, 205, 206) as well as discrete frames (310, 320, 330) from a tracking video, as illustrated in the Fig.9. Additionally, frame (1250, 1260) demonstrates the closed-loop tracking performance of the algorithm presented in this paper when the right rotor of the unmanned aerial vehicle serves as the target point of engagement.

These results collectively demonstrate the algorithm's capability to accurately detect and track UAV targets in complex environments. The algorithm's ability to handle real-time challenges, such as accurate detection at long distances and precise tracking, validates its efficacy in UAV countermeasure applications.

V. CONCLUSION

In this study, we constructed a large-scale dataset named UAV-ADT (UAV Attitude Detection and Tracking), laying the foundation for the proposed low-altitude unmanned aerial vehicle (UAV) attitude detection and tracking method, UAV-Pose. This method was successfully applied to laser UAV countermeasure systems, achieving the groundbreaking capability of tracking UAV feature points, overcoming the previous limitation of tracking only the entire UAV bounding box. The deployment on different inference frameworks and hardware confirmed the efficiency of UAV-Pose, particularly on the NVIDIA Jetson NX, where we achieved an impressive detection speed of 300fps while maintaining PCK and map75 metrics at 99.3% and 97.8%, respectively, meeting the high demands for both speed and accuracy. This innovation significantly enhanced the precision of laser UAV countermeasure systems.

Our designed dual-capture network demonstrated multiple advantages across various stages and task requirements. Firstly, multi-scale processing allowed us to flexibly address scale variations in low-altitude UAVs, adapting to target detection and tracking at different distances and speeds. Secondly, the smaller field-of-view network in the tracking stage improved computational efficiency and provided precise attitude tracking when the target was in continuous motion. Additionally, our approach excelled in handling challenges such as target occlusion and background variations in complex scenarios.

In summary, the UAV-Pose method achieved remarkable success in the field of low-altitude UAV attitude detection and tracking, establishing a solid foundation for achieving laser precision strikes and providing crucial military intelligence support. However, future research directions become even more intriguing. We plan to expand this method by incorporating more UAV feature points in detection and tracking, extending its application to other types of UAVs, including fixed-wing and cruise missiles. This extension aims to bring broader applicability and practicality to the field of UAV countermeasures, offering comprehensive solutions for future military and security challenges.

REFERENCES

[1] M. Lort, A. Aguasca, C. López-Martínez, and T. M. Marín, "Initial evaluation of SAR capabilities in UAV multicopter platforms," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 1, pp. 127–140, Jan. 2018.

[2] V. Chamola, P. Kotesh, A. Agarwal, Naren, N. Gupta, and M. Guizani, "A comprehensive review of unmanned aerial vehicle attacks and neutralization techniques," *Ad Hoc Netw.*, vol. 111, Feb. 2021, Art. no. 102324.

[3] H.-W. Lee, "Research on multi-functional logistics intelligent unmanned aerial vehicle," *Eng. Appl. Artif. Intell.*, vol. 116, Nov. 2022, Art. no. 105341.

[4] T.-Z. Xiang, G.-S. Xia, and L. Zhang, "Mini-unmanned aerial vehicle-based remote sensing: Techniques, applications, and prospects," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 3, pp. 29–63, Sep. 2019.

[5] W. Liu, T. Zhang, S. Huang, and K. Li, "A hybrid optimization framework for UAV reconnaissance mission planning," *Comput. Ind. Eng.*, vol. 173, Nov. 2022, Art. no. 108653.

[6] F. Hoffmann, M. Ritchie, F. Fioranelli, A. Charlish, and H. Griffiths, "Micro-Doppler based detection and tracking of UAVs with multistatic radar," in *Proc. IEEE Radar Conf. (RadarConf)*, May 2016, pp. 1–6.

[7] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Jun. 2017, pp. 1137–1149.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[11] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

[12] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," Apr. 2018, *arXiv:1804.02767*.

[13] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Liao, "YOLOv4: Optimal speed and accuracy of object detection," Apr. 2020, *arXiv:2004.10934*.

[14] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "YOLOv6: A single-stage object detection framework for industrial applications," Sep. 2022, *arXiv:2209.02976*.

[15] Y. Li, S. Yang, P. Liu, S. Zhang, Y. Wang, Z. Wang, W. Yang, and S.-T. Xia, "SIMCC: A simple coordinate classification perspective for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, Jul. 2021, pp. 89–106.

[16] J. Huang, Z. Zhu, F. Guo, and G. Huang, "The devil is in the details: Delving into unbiased data processing for human pose estimation," in

Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 5699–5708.

[17] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 472–487.

[18] F. Zhang, X. Zhu, H. Dai, M. Ye, and C. Zhu, "Distribution-aware coordinate representation for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7091–7100.

[19] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "ViTPose: Simple vision transformer baselines for human pose estimation," in *Proc. Adv. Neural Inf. Process. Syst.*, Apr. 2022, pp. 38571–38584.

[20] B. K. S. Isaac-Medina, M. Poyser, D. Organisciak, C. G. Willcocks, T. P. Breckon, and H. P. H. Shum, "Unmanned aerial vehicle visual detection and tracking using deep neural networks: A performance benchmark," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 1223–1232.

[21] A. Coluccia, "Drone-vs-bird detection challenge at IEEE AVSS2019," in *Proc. 16th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Sep. 2019, pp. 1–7.

[22] A. Rodriguez-Ramos, J. Rodriguez-Vazquez, C. Sampedro, and P. Campoy, "Adaptive inattentional framework for video object detection with reward-conditional training," *IEEE Access*, vol. 8, pp. 124451–124466, 2020.

[23] N. Jiang, W. Kuiran, X. Peng, X. Yu, Q. Wang, J. Xing, G. Li, J. Zhao, G. Guo, and Z. Han, "Anti-UAV: A large multi-modal benchmark for UAV tracking," Jan. 2021, *arXiv:2101.08466*.

[24] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*, Nov. 2020, pp. 213–229.

[25] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3464–3468.

[26] J. Zhao, J. Zhang, D. Li, and D. Wang, "Vision-based anti-UAV detection and tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25323–25334, Dec. 2022.

[27] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOx: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.

[28] Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu, "SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines," in *Proc. Assoc. Adv. Artif. Intell.*, Apr. 2020, pp. 12549–12556.

[29] S. Dogru and L. Marques, "Pursuing drones with drones using millimeter wave radar," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4156–4163, Jul. 2020.

[30] M. M. Junior and B. Guo, "Sensing spectrum sharing based massive MIMO radar for drone tracking and interception," *PLoS ONE*, vol. 17, no. 5, May 2022, Art. no. e0268834.

[31] X. Shi, C. Yang, W. Xie, C. Liang, Z. Shi, and J. Chen, "Anti-drone system with multiple surveillance technologies: Architecture, implementation, and challenges," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 68–74, Apr. 2018.

[32] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[33] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[34] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.

[35] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1577–1586.

[36] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[37] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2018, pp. 3–19.

[38] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13708–13717.

JIANG YOU received the bachelor's degree from the University of Electronic Science and Technology of China, in 2019. He is currently pursuing the Ph.D. degree with the China Academy of Engineering Physics. His research interests include beam control and target detection.

ZIXUN YE received the bachelor's degree from the Southwest University of Science and Technology, in 2021, where he is currently pursuing the master's degree. He is working as an Algorithm Engineer with the Institute of Applied Electronics, China Academy of Engineering Physics. His research interests include encompass computer vision, image processing, object tracking, 2D and 3D object detection, and facial recognition.

JINGLIANG GU received the bachelor's degree in optoelectronic information engineering from Zhejiang University, in 2002, and the master's degree in optical engineering from the Graduate School, China Academy of Engineering Physics, in 2005. Since 2005, he has been an Assistant Researcher, an Associate Researcher, and a Senior Engineer with the Institute of Applied Electronics, China Academy of Engineering Physics. His main research interests include automatic control theory and algorithms, digital image processing, object detection and recognition, pattern recognition, and other related areas.

JUNTAO PU received the bachelor's degree from the Southwest University of Science and Technology, in 2021, where he is currently pursuing the master's degree. He is working as an Algorithm Engineer with the Institute of Applied Electronics, China Academy of Engineering Physics. His research interests include computer vision, image processing, object tracking, small target detection, and edge computing.