Real-Time Detection for Small UAVs: Combining YOLO and Multi-frame Motion Analysis

Juanqin Liu, Leonardo Plotegher, Eloy Roura, Cristino de Souza Junior, Shaoming He*

***Abstract*—Unmanned Aerial Vehicle (UAV) detection technology plays a critical role in mitigating security risks and safeguarding privacy in both military and civilian applications. However, traditional detection methods face significant challenges in identifying UAV targets with extremely small pixels at long distances. To address this issue, we propose the Global-Local YOLO-Motion (GL-YOMO) detection algorithm, which combines You Only Look Once (YOLO) object detection with multi-frame motion detection techniques, markedly enhancing the accuracy and stability of small UAV target detection. The YOLO detection algorithm is optimized through multi-scale feature fusion and attention mechanisms, while the integration of the Ghost module further improves efficiency. Additionally, a motion detection approach based on template matching is being developed to augment detection capabilities for minute UAV targets. The system utilizes a global-local collaborative detection strategy to achieve high precision and efficiency. Experimental results on a self-constructed fixed-wing UAV dataset demonstrate that the GL-YOMO algorithm significantly enhances detection accuracy and stability, underscoring its potential in UAV detection applications.**

***Index Terms*—Small UAV detection, YOLO, Moving object detection, Global-Local collaborative detection**

I. Introduction

Since UAV technology emerged, its widespread application across various domains has raised significant safety risks and privacy concerns [1], [2], [3]. In response, the development of long-range drone detection technology has become critical, enabling the prompt identification, localization, and intervention of drones to safeguard public safety and personal privacy. However, existing detection technologies face considerable challenges when addressing small-pixel UAV targets at long distances [4], [5], [6]. Drones typically occupy less than 0.1% of an image, leading to insufficient feature information. When combined with complex backgrounds, this results in decreased detection accuracy [7], [8]. Fig. 1 highlights some of the common challenges associated with detecting drone targets.

While popular object detection methods like Fast-RCNN, YOLO, and DETR are highly effective for larger targets, they suffer from high false positive and false negative rates i+n long-range, small-object drone detection [9], [10], [11]. Recent researchers have developed specialized detection methods optimized for drone characteristics [12], [13], [14]. For example, by integrating appearance and motion features, using techniques like frame differencing or optical flow to extract moving objects, and applying classification methods to distinguish drone targets from other interfering objects. However, due to the diminutive size of drone targets, the amount of useful feature information available for classification is limited, and downsampling often leads to the loss of critical information. Additionally, the presence of diverse noise between drones and complex backgrounds makes it challenging to construct a precisely annotated dataset that can reliably distinguish targets from their surroundings.

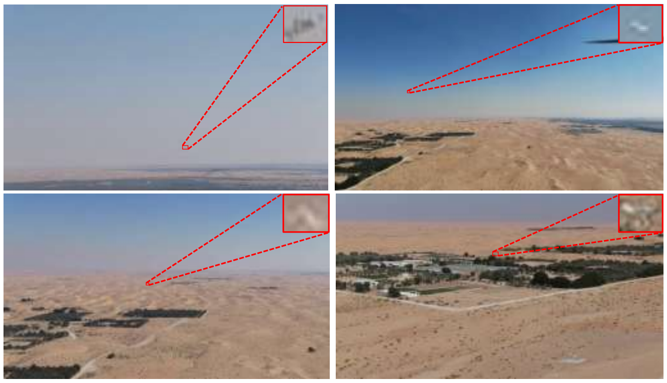


Fig. 1. Challenging conditions examples in UAV detections: 1) Minuscule targets with limited distinctive features; 2) Intricate backgrounds complicating target identification.

In this paper, we propose the GL-YOMO detection algorithm, which effectively integrates appearance and motion features. By enhancing the YOLO model, we aim to improve detection accuracy while reducing computational complexity. The algorithm incorporates multi-frame motion detection for secondary validation, ensuring precise detection of small objects without requiring a manually constructed classification dataset.

The main contributions of this paper are as follows:

1) **Development of the GL-YOMO Detection Algorithm**: This algorithm combines YOLO object detection with multi-frame motion detection, leveraging YOLO's efficient detection capabilities while incorporating motion feature capture to significantly enhance detection accuracy and stability.

2) **Improvement of the YOLO Model**: The enhancements increase detection accuracy and substantially reduce computational complexity and parameter count, resulting in a more efficient and lightweight model.

3) **Design of a Template Matching-Based Motion Detec-**

This work was supported by the Technology Innovation Institute under Contract No. TII/ARRC/2154/2023.

Juanqin Liu and Shaoming He are with the School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China.

Leonardo Plotegher, Eloy Roura and Cristino de Souza Junior are with the Autonomous Robotics Research Centre, Technology Innovation Institute, P.O.Box: 9639, Masdar City, Abu Dhabi, United Arab Emirates.

tion Algorithm: By analyzing pixel changes and displacement variations across three consecutive frames, this algorithm effectively detects extremely small objects, further improving small-object detection accuracy.

4) **Construction of the Fixed-Wings Dataset**: This dataset includes 13 video sequences and 24,951 frames, encompassing numerous UAV targets with an average image proportion of 0.01%, providing a robust resource for evaluating UAV detection algorithms.

The paper is structured as follows: Section II reviews small object detection and related methods for UAV detection. Section III details our proposed method. Section IV presents experimental results and analysis. Section V concludes the paper and discusses future research directions.

II. Related Work

A. Small Object Detection Methods

Since the introduction of the YOLO algorithm in 2016, its subsequent versions [15], [16], [17] have continually evolved, driving substantial advancements in object detection. To tackle the challenges associated with small object detection, many different useful strategies were proposed in recent works. Among these, multi-scale feature fusion emerged as a critical approach, effectively integrating semantic information across various levels to improve small object detection [18], [19], [20], [21], [22]. For example, Gold-YOLO [20] enhanced feature fusion by incorporating a Gather-and-Distribute mechanism, achieving a 39.9% AP on the COCO val2017 dataset, which is a 2.4% enhancement over the prior state-of-the-art model, YOLOv6., while ASF-YOLO [21] significantly improved small object detection and segmentation through the Scale Sequence Feature Fusion (SSFF) module and the Triple Feature Encoder (TPE) module.

Feature enhancement strategies were also pivotal, with many studies introducing attention mechanisms to amplify target features while suppressing background noise, thereby boosting detection accuracy. Notably, CEAM-YOLOv7 [23] incorporated global attention mechanisms in both the Backbone and Head, improving the mAP by 20.26% than the original YOLOv7 model. Gong et al. [24] introduced a normalization-based attention module that incrementally improved detection by focusing on channels and spatial dimensions through coefficient penalization, achieving a 7.1% improvement on the DOTA dataset. The AIE-YOLO [25] employed a context enhancement module that fuses multi-scale receptive fields with attention mechanisms to optimize feature representation. Additionally, super-resolution techniques led to considerable improvements [26], [27]. For instance, Yucong et al. [26] combined shallow high-resolution geometric details with deep super-resolution semantic features, augmented by channel attention, to significantly elevate detection precision. Collectively, these innovations represented a substantial leap forward in the field of small object detection.

B. UAV Detection Methods

Ensuring real-time performance in UAV detection while addressing the challenges posed by small objects and multi-scale variations is crucial. The YOLO series proved to be one of the most effective solutions for these challenges[28], [29], [30]. For example, the work in [10] integrated the Spatial Pyramid Pooling module into the YOLOv4 model's prediction head, combining it with a spatial attention mechanism to achieve real-time drone detection at 14.9 fps. Similarly, the authors in [31] leveraged the lightweight MobileNet as the backbone, incorporating depthwise separable convolution techniques, and reported a processing speed of 82 fps along with an accuracy of 93.52% mAP on a custom UAV dataset. Although these methods excelled at extracting prominent features from single frames and performed well in simpler scenarios, they encountered difficulties in more complex environments.

To better handle challenges such as background interference and target occlusion in more intricate environments, researchers explored motion-based detection methods for drone targets. For instance, a motion detector that employed background subtraction was proposed in [32], leveraging post-processing to identify moving objects and utilizing the MobileNetV2 classifier for UAV classification, achieving an accuracy metric of 70.1% on the Drone-vs-Bird dataset [33]. However, this approach was limited in scenarios involving moving cameras. To address this limitation, the work in [34] introduced an optical flow-based motion information extractor that replaced the upsampling component in a standard Feature Pyramid Network, thereby significantly improving UAV detection accuracy, resulting in an impressive average precision(AP) of 67.8 on the Drone-vs-Bird dataset. Additionally, the studies in [35], [36] exploited optical flow estimation to generate candidate points relative to background motion, estimated background dynamics using a global motion model and applied background subtraction in conjunction with a hybrid detector for UAV classification. Low-rank methods, as presented in [37], [38], contributed to the field by generating small motion regions that facilitated drone target classification. Moreover, integrating visual and motion features was shown to enhance UAV detection. For example, the authors in [4] combined YOLO detectors with a motion target classifier through a global-local detection approach, 92% accuracy and 23.6FPS performance were achieved on the custom ARD-MAV dataset. The approach in [39] achieved precise detection using multi-channel temporal frames and spatiotemporal semantic segmentation with convolutional neural networks, alongside a ResNet classifier, which resulted in an average F1-score of 0.92 across three test videos of the drone-vs-bird dataset. Collectively, these approaches represented significant advancements in UAV target detection within complex environments.

III. Methods

We propose a high-precision and robust detection method called GL-YOMO, as depicted in Fig. 2. This method utilizes a global-local collaborative detection strategy. The process begins with object detection and localization across the entire 1920×1080 global frame. Once an object is consistently detected across consecutive frames, the system narrows its focus to a 300×300 local region for more detailed detection. The detection is carried out by our customized YOMO detector, which combines the YOLO detector with multi-frame motion

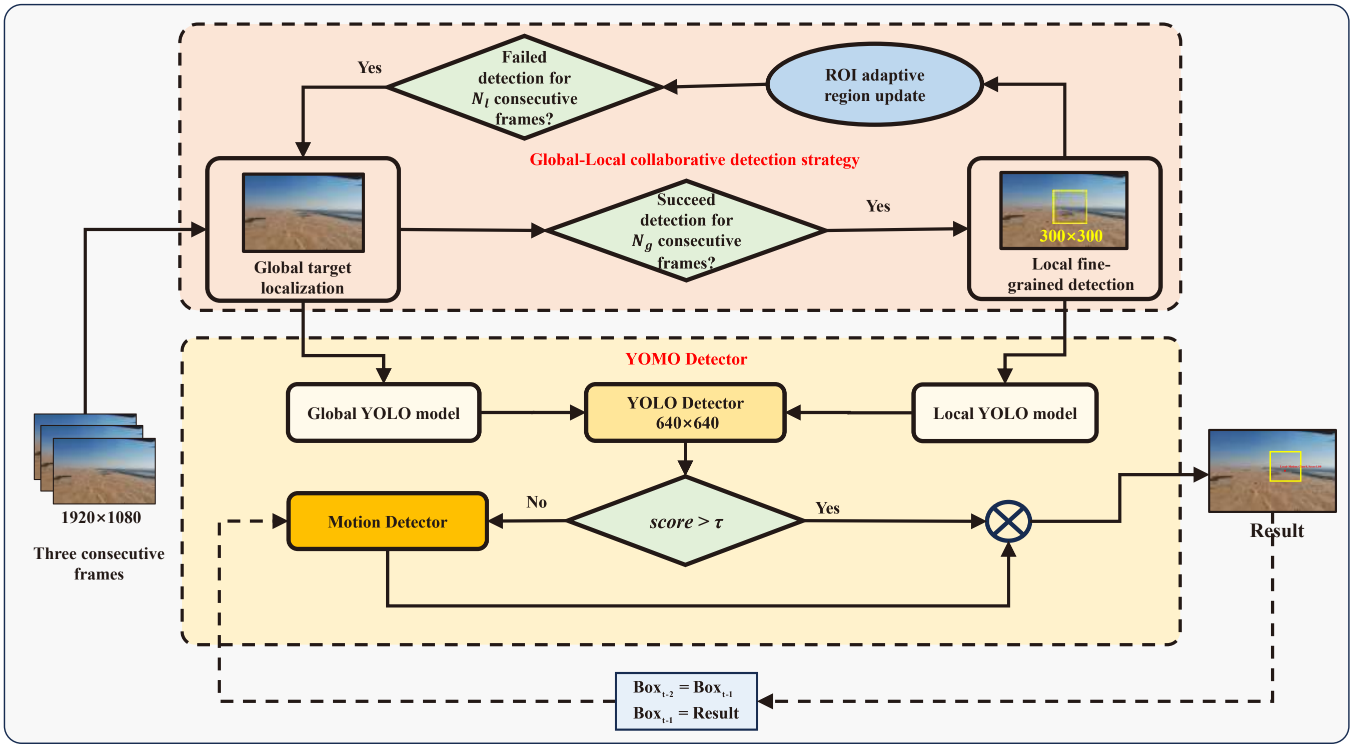

Global-Local collaborative detection strategy
Yes
Failed detection for N_l consecutive frames?
ROI adaptive region update
Global target localization
Succeed detection for N_g consecutive frames?
Yes
300×300
Local fine-grained detection
YOMO Detector
Global YOLO model
YOLO Detector 640×640
Local YOLO model
1920×1080
Three consecutive frames
Motion Detector
No
$score > \tau$
Yes
Result
$\mathrm{Box}_{t-2} = \mathrm{Box}_{t-1}$
$\mathrm{Box}_{t-1} = \mathrm{Result}$

Fig. 2. The GL-YOMO architecture: The YOMO Detector combines YOLO Detector with a multi-frame Motion Detector. A global-local strategy and ROI adaptive update enable continuous detection of small UAV targets.

detection technology to enable efficient object capture and tracking. To ensure stable and continuous detection of small UAV targets, an Region of Interest (ROI) adaptive update mechanism is integrated, dynamically adjusting the ROI to continuously refresh the detection scope.

A. Global-Local Collaborative Detection Strategy

While global detection offers broad coverage, it is susceptible to background noise and environmental interference, increasing the likelihood of false negatives and false positives. In contrast, local detection, with its narrower focus, is better suited for accurately capturing target locations. To enhance detection accuracy, this study adopts a global-local collaborative strategy, incorporating two key components: dynamic switching between global and local detection modes and adaptive updating of the ROI.

1) Dynamic Switching: The dynamic switching strategy between global and local detection modes is based on an analysis of target detection frame rates. Initially, the system performs global target localization. Once a target is consistently detected in N_g consecutive frames, the system automatically transitions to local mode for more refined detection. If no targets are detected within N_l consecutive frames in local mode, the system reverts to global mode for re-localization. By establishing appropriate frame thresholds (N_g and N_l), this approach enables seamless switching between global and local detection modes, thereby enhancing the system's stability and reliability for UAV target detection in complex environments.

2) ROI Adaptive Region Update: The ROI adaptive updating is crucial for defining the local detection range and directly impacts target detection effectiveness. When transitioning from global to local detection modes, the system crops an ROI, focusing the detection range on an initial 300×300 pixel area for more refined local detection. To accommodate UAV target movement, a dynamic strategy is employed based on the target's proximity to the ROI edge. If the target remains within R_s of the ROI's radius, the ROI remains unchanged; if the target moves beyond this range, the ROI is automatically updated to center on the target. This approach improves fault tolerance and reduces errors caused by frequent ROI updates. Additionally, to mitigate potential missed detections, the system expands the ROI based on missed frames, ensuring effective detection while preserving the advantages of local detection and minimizing the loss of global detection accuracy, as outlined in (1).

$$R_{size} = 300 + k_1 \cdot F_{lost} \tag{1}$$

where ROI is defined by an area of $R_{size} \times R_{size}$, k_1 is a proportional coefficient used to control the rate of ROI expansion, and F_{lost} represents the number of consecutive frames lost.

B. YOMO Detector

The YOMO Detector, a central method for detecting small UAV targets, comprises two key components: the YOLO Detector and the Motion Detector. The YOLO Detector is

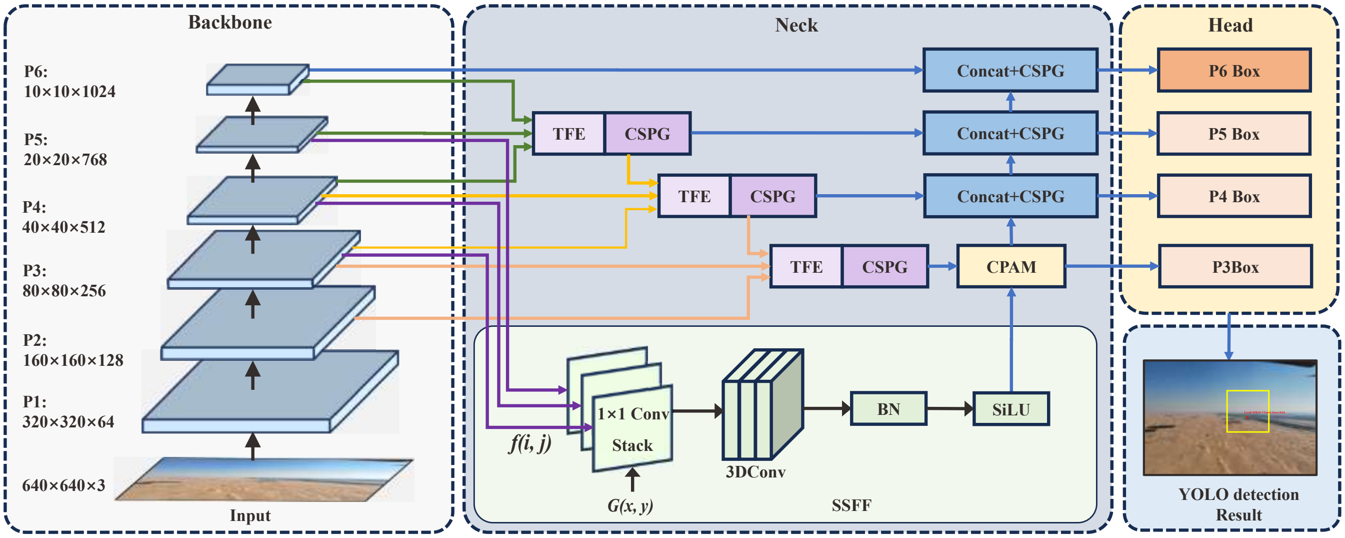


Fig. 3. YOLO network model architecture diagram: The model integrates SSFF, TFE, and CPAM modules from ASF-YOLO and replaces the C3 structure in the CSP module with C3Chost, forming the CSPG module. The model utilizes four detection heads to accommodate multi-scale object detection tasks.

known for its efficiency in target detection, while the Motion Detector captures the dynamic characteristics of targets. Together, these components work synergistically to significantly improve detection accuracy.

1) YOLO Detector: Although the YOLO algorithm is highly effective in a range of object detection tasks, it faces challenges when dealing with very small objects due to feature loss during downsampling. To overcome this limitation and inspired by ASF-YOLO [21], we introduce a new multi-scale feature fusion approach with its architecture illustrated in Fig. 3.

In the YOLO Detector, both the global detection and local detection phases involve resizing the input to 640×640 pixels, this adjustment was made to balance inspection accuracy with processing speed, ensuring high performance and reducing the computational burden, and multi-scale feature extraction is leveraged through successive downsampling in the Backbone. The backbone network is based on the YOLOv5 CSPDarknet53 architecture, utilizing stacked C3 modules and the Conv-BN-SiLU structure for feature extraction. This Backbone module generates four crucial feature layers: P3, P4, P5, and P6, each capturing features at different scales. These feature layers are then fed into the Neck for further multi-scale feature fusion. In the Neck, We have integrated the TFE module, as proposed by ASF-YOLO, which ingeniously fuses feature maps of large, medium, and small scales, concatenating them along the channel dimension. This approach preserves the fine-grained information crucial for detecting small objects. Additionally, the SSFF module employed Gaussian smoothing techniques, progressively increasing the standard deviation to process feature maps. It then utilizes 3D convolution technology for stacking and processing, enabling the model to better handle objects of varying sizes, orientations, and aspect ratios while capturing scale-space relationships. Subsequently, we combine the Channel and Position Attention Mechanism (CPAM) with the SSFF and TFE modules, CPAM captures cross-channel interactions through its channel attention mechanism without dimension reduction, while its position attention mechanism processes feature maps along horizontal and vertical axes separately. This allows the model to adaptively focus on small objects across different scales. The incorporation of multi-scale feature fusion and attention mechanisms allows the model to more accurately detect extremely small objects in drone imagery. The Head is a critical part of the YOLO module for object detection, and four detection heads are used, corresponding to the feature maps P3, P4, P5, and P6 at different scales. The Head section's multi-scale feature fusion allows the model to detect objects of various sizes, enhancing its performance in complex scenes and multi-scale target scenarios.

In terms of model efficiency, we have integrated the efficient Ghost module [40] into various layers of the YOLO model's Backbone and Neck. This includes replacing traditional convolution operations with GhostConv and substituting the original C3 module with the C3Ghost module. These modifications significantly reduce the model's computational load and parameter count without compromising performance.

Based on this enhanced YOLO model architecture, the dataset is used for training to produce the required YOLO models. It is important to emphasize that the training for the YOLO models in the global and local detection stages is distinct in focus. During the global detection stage, the model is trained on full images to create the Global YOLO Model, enabling comprehensive detection of targets across the entire image. Conversely, in the local detection stage, the model is trained on cropped images, resulting in the Local YOLO Model, which is specifically tailored for more precise target recognition within the ROI.

2) Motion Detector: When the confidence score of the YOLO detector falls below a predefined threshold, the sys-

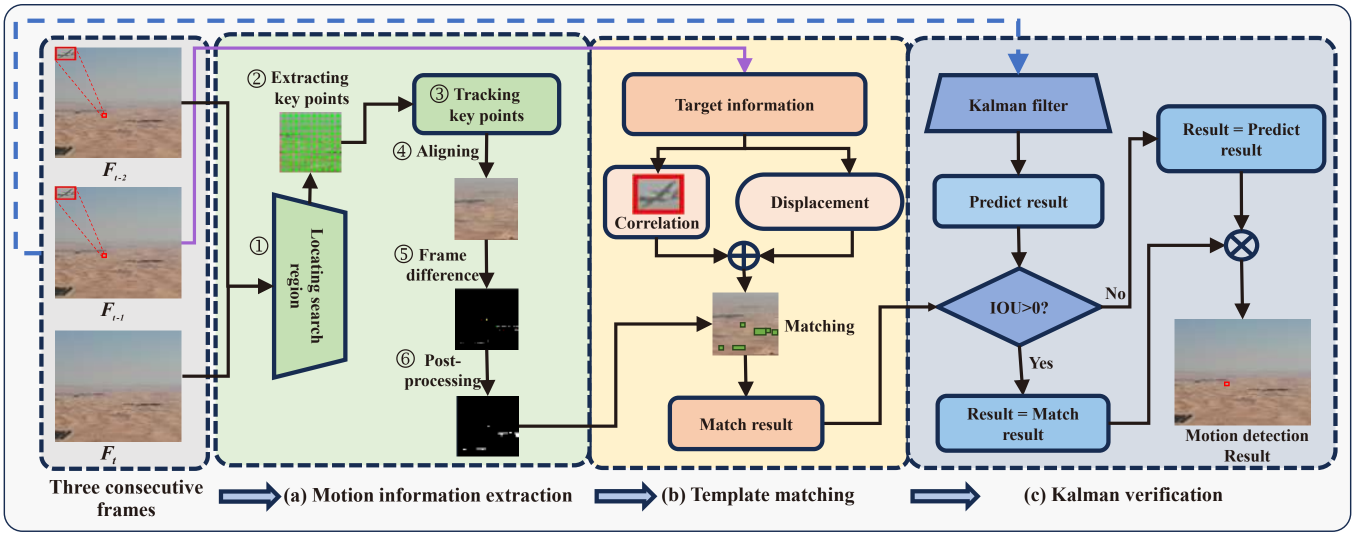


Fig. 4. Motion detection algorithm framework diagram: Three consecutive frames undergo three key stages: motion information extraction, template matching, and Kalman verification, to obtain the final motion detection result.

tem turns to the Motion Detector for further analysis. The reliability of the YOLO detection results is determined by the confidence threshold τ. In global detection, if the YOLO confidence score exceeds the threshold τ_g, the detection result is considered valid; otherwise, it is deemed invalid, prompting the system to activate the motion detector for supplementary analysis. Similarly, in local detection, if the confidence score exceeds the threshold τ_l, the detection result is considered valid; if not, the system will initiate the motion detector for further detection. The motion detector integrates optical flow methods, template matching techniques, and Kalman filtering algorithms to comprehensively analyze the motion information between frames. The implementation steps are detailed in Fig. 4, where F_t denotes the frame at time instant t.

(a) Motion Information Extraction. To accurately extract motion information, it is crucial to isolate moving objects from dynamic backgrounds, forming the foundation for subsequent template-matching algorithms. We employ a frame interval-based strategy, utilizing the interval between F_{t-2} and F_t to capture motion information. This approach reveals motion changes more distinctly than using consecutive frames, particularly for small UAV targets. According to Fig. 4(a), the process of extracting motion information is as follows:

To retain small-sized UAV targets while minimizing noise, motion information is extracted from a localized region around the target position rather than the entire image. Using the target coordinates from F_{t-2} as the center, crop a $\Delta p \times \Delta p$ pixel region (e.g., $\Delta p = 50$) for optical flow extraction. This region size, determined through extensive experimentation, adequately encompasses the target's motion range over three frames, ensuring accurate and effective motion capture while avoiding unnecessary noise and enhancing processing efficiency. Partial results of motion feature extraction are shown in Fig. 5.

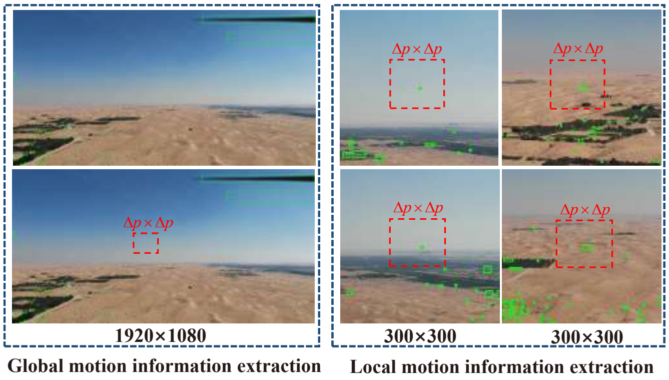

$\Delta p \times \Delta p$
1920×1080
300×300
300×300
Global motion information extraction
Local motion information extraction

Fig. 5. Examples of motion information extraction at global and local scales: The green boxes highlight regions containing moving targets or noise, where distinguishing between small UAV targets and noise is challenging. The red boxes denote the motion information extraction areas defined by our method, which effectively minimizes interference from extraneous noise.

The extraction method leverages optical flow to track motion. Key points within the extraction region are first identified using a grid-based approach, which divides the area into a grid and selects key points at the intersections to capture critical motion features. The pyramidal Lucas-Kanade method is then employed for multi-scale analysis of optical flow, allowing precise tracking of key point trajectories and subtle motion changes [41]. To account for camera movement or perspective shifts, a 2D perspective transformation is applied to the previous frame F_{t-2}, which maps points from one plane to another, aligning the content of F_{t-2} geometrically with F_t. This alignment compensates for motion unrelated to scene dynamics, ensuring that static elements in the scene are closely matched. After alignment, frame difference is performed to isolate genuine scene changes, following a process similar to that outlined in [4]. Significant differences between frames are identified by computing pixel-wise differences between the aligned frame and the current frame. To further enhance the results, corrections for lighting variations and background motion are applied, followed by binarization to emphasize regions of motion. The frame difference identifies pixels corresponding to moving objects, but the output often contains noise and lacks smoothness. therefore post-processing techniques are necessary for noise suppression and shape optimization. First, morphological operations such as erosion and dilation are used to refine the contours of detected objects, improving their structure. Next, median filtering and Gaussian blurring are applied to further reduce noise and smooth the object boundaries. After noise reduction, a thresholding operation is performed to create a clean binary image. Finally, connected component analysis is used to segment individual objects, enabling the extraction of relevant attributes including their location and size.

(b) Template Matching. Given the high consistency of the target across consecutive frames, we adopt a template-matching approach to solve the target tracking problem. The target detected in the previous frame is used as a template and matched with the motion region extracted from optical flow analysis in the current frame. To enhance matching accuracy, we introduce a weighted matching mechanism that combines correlation metrics with displacement information from three consecutive frames, providing a more comprehensive assessment of the target's matching probability.

Specifically, the normalized cross-correlation coefficient is employed to measure the visual similarity between the target region and the template as

$$NCC_{(x,y)} = \frac{\sum\limits_{u,v} (T_{(u,v)} - \bar{T})(I_{(x+u,y+v)} - \overline{I_{(x,y)}})}{\sqrt{\sum\limits_{u,v} (T_{(u,v)} - \bar{T})^2 \sum\limits_{u,v} (I_{(x+u,y+v)} - \overline{I_{(x,y)}})^2}} \tag{2}$$

which is then normalized to the range $[0, 1]$, as

$$N_c = \frac{NCC_{(x,y)} + 1}{2} \tag{3}$$

where $T_{(u,v)}$ denotes the pixel value of the template image at coordinate (u, v), $\bar{T}$ represents the mean value of the template image, $I_{(x+u,y+v)}$ denotes the pixel value of the original image at location $(x+u, y+v)$, and $\overline{I_{(x,y)}}$ represents the mean pixel value of the region in the original image with (x, y) as the top-left corner and the same size as the template.

To account for potential variations in target size, a multi-scale visual similarity evaluation is performed for each candidate region. The candidate region is scaled up or down according to three different scale factors, and the visual similarity between the target and the previous frame is computed at each of these scales. The final similarity C_c is defined as the maximum normalized cross-correlation N_c across the different scales as

$$C_c = \max\left\{N_c^{\text{scale}=0.7}, N_c^{\text{scale}=1.0}, N_c^{\text{scale}=1.3}\right\} \tag{4}$$

where N_c^{scale} represent the normalized cross-correlation value at each respective scale.

Define C_d as the similarity of the displacement between the current target and the target in the previous frame, and C_d is determined by

$$C_d = 1 - (\Delta d_{norm} + \Delta\theta_{norm})/2 \tag{5}$$

where Δd_{norm} and $\Delta\theta_{norm}$, respectively, represent the normalized distance difference and direction difference, which are determined by

$$\Delta d_{norm} = \frac{|d_{F_t} - d_{F_{t-1}}|}{\sqrt{w_t^2 + h_t^2}} \tag{6}$$

$$\Delta\theta_{norm} = \frac{|\theta_{F_t} - \theta_{F_{t-1}}|}{\pi} \tag{7}$$

where d_{F_t} is the pixel Euclidean distance between the current frame and its preceding frame, i.e., the pixel distance between two bounding box center points. We use the image dimensions w_t and h_t of the current frame to normalize Δd_{norm}. Likewise, the normalized direction difference $\Delta\theta_{norm}$ is derived by normalizing the directional change between two consecutive frames. Define $(x_{c,t}, y_{c,t})$ as the pixel position of the center of the bounding box at time instant t, the Euclidean distance d_{F_t} and directional change θ_{F_t} are given by

$$d_{F_t} = \sqrt{(x_{c,t} - x_{c,t-1})^2 + (y_{c,t} - y_{c,t-1})^2} \tag{8}$$

$$\tan\theta_{F_t} = \frac{y_{c,t} - y_{c,t-1}}{x_{c,t} - x_{c,t-1}} \tag{9}$$

The final weighted matching cost, denoted by C_w, can then be given by

$$C_w = k_2 \cdot C_c + k_3 \cdot C_d \tag{10}$$

where the constants k_2 and k_3 are weighting coefficients used to balance the impact of C_c and C_d in the final matching result.

(c) Kalman Filter Verification. To enhance detection accuracy and robustness, we introduce the Kalman filter as a verification mechanism for state estimation. Throughout the target detection process, we use the 8-state Kalman filter to predict the target state in the next frame. The state vector $\mathbf{x}_t = [x, y, w, h, v_x, v_y, v_w, v_h]^T$ includes the target position, size, and rate of change. Here, (x, y) represent the top-left coordinates of the target bounding box, (w, h) its width and height, (v_x, v_y) the velocities, and (v_w, v_h) the rates of change in width and height. We leverage the output of the YOLO detection as the observation vector $\mathbf{z}_t$, i.e., $\mathbf{z}_t = [x, y, w, h]^T$. With this in mind, the motion of the bounding box can modeled by a linear dynamics as

$$\begin{cases} \mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} \\ \mathbf{z}_t = \mathbf{H}\mathbf{x}_t \end{cases} \tag{11}$$

where $\mathbf{F}$ is the state transition matrix and $\mathbf{H}$ denotes the obervation matrix.

The Kalman filter operates in two key stages: prediction and update. The prediction stage can be expressed mathematically as

$$\begin{cases} \hat{\mathbf{x}}_{t|t-1} = \mathbf{F}\hat{\mathbf{x}}_{t-1|t-1} \\ \mathbf{P}_{t|t-1} = \mathbf{F}\mathbf{P}_{t-1|t-1}\mathbf{F}^T + \mathbf{Q} \end{cases} \tag{12}$$

where $\hat{\mathbf{x}}_{t|t-1}$ represents the predicted state at time t, $\mathbf{P}_{t|t-1}$ is the covariance matrix of the predicted state, and $\mathbf{Q}$ is the process noise covariance matrix.

In the update phase, the detection results provided by the YOLO detector are used to correct the predicted state as

$$\begin{cases} \mathbf{y}_t = \mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t-1} \\ \mathbf{S}_t = \mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^T + \mathbf{R} \\ \mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}^T\mathbf{S}_t^{-1} \\ \hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t\mathbf{y}_t \\ \mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_{t|t-1} \end{cases} \tag{13}$$

where $\mathbf{y}_t$ is the residual vector, $\mathbf{S}_t$ is the covariance matrix of the residual, $\mathbf{K}_t$ is the Kalman gain, $\mathbf{R}$ is the measurement noise covariance matrix, and $\hat{\mathbf{x}}_{t|t}$ and $\mathbf{P}_{t|t}$ are the updated state and state covariance matrix, respectively.

During motion detection, we combine the Kalman filter's predicted output with the template matching results, verifying accuracy through the Intersection over Union (IOU) ratio. Since the target UAV occupies a minimal area in the image, the template matching is confirmed to be accurate if the IOU is positive. In contrast, zero or negative IOU indicates a potential mismatch and the detection output updates the target position based on the Kalman filter prediction.

C. Summary

Algorithm 1 provides a detailed description of the core logic and procedural flow of the proposed GL-YOMO method. The detection outputs are characterized by bounding box *boxes*, detection scores *scores* and target ID *class_ids*.

IV. Experiments

A. Dataset

In evaluating the performance of our proposed GL-YOMO algorithm, we selected two challenging video datasets to ensure a comprehensive and accurate assessment.

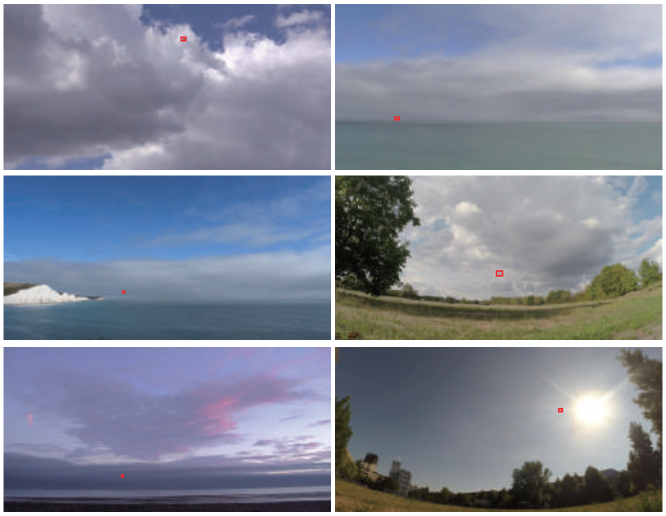


Fig. 6. Sample Images from the Drone-vs-Bird dataset. The red boxes highlight actual drone targets.

Algorithm 1 GL-YOMO Algorithm

```
Require: Previous two frames and current frame, i.e., $F_{t-2}, F_{t-1}, F_t$
    $Y_{\text{det}}$: YOLO Detector
    $M_{\text{det}}$: Motion Detector
Ensure: Detection results, i.e., $boxes, scores, class\_ids$
Initialize $N_g \leftarrow 0$, $N_l \leftarrow 0$, detector $\leftarrow$ 'global', $ROI \leftarrow$ None
for each frame $F_t$ do
    if detector = 'global' then
        $boxes, scores, class\_ids \leftarrow Y_{\text{det}}(F_t)$
    else
        $boxes, scores, class\_ids \leftarrow Y_{\text{det}}(ROI(F_t))$
    end if
    if $boxes = \emptyset$ then
        $boxes, scores, class\_ids \leftarrow M_{\text{det}}(F_{t-2}, F_{t-1}, F_t)$
        ▷ Switch to motion detection
    end if
    if $boxes \neq \emptyset$ then
        Update $ROI$ based on $boxes$
        if detector = 'global' then
            $N_g \leftarrow N_g + 1$, $N_l \leftarrow 0$
            if $N_g \geq 30$ then
                detector $\leftarrow$ 'local'
            end if
        else
            $N_l \leftarrow 0$
        end if
    else
        if detector = 'local' then
            $N_l \leftarrow N_l + 1$
            Enlarge $ROI$
            if $N_l \geq 60$ then
                detector $\leftarrow$ 'global'
                $N_g \leftarrow 0$
                Reset $ROI$ to full frame
            end if
        end if
    end if
    Update frame history: $F_{t-2} \leftarrow F_{t-1}$, $F_{t-1} \leftarrow F_t$
end for
```

1) Drone-vs-Bird Dataset: The Drone-vs-Bird dataset consists of 77 videos, encompassing a total of 104,760 frames. Many of the videos feature small drones filmed from considerable distances, frequently accompanied by birds and insects, adding an extra layer of complexity to the surveillance tasks. Captured with both static and dynamic cameras, these videos effectively simulate a wide range of outdoor scenarios. The average object size is 34×23 pixels, constituting 0.1% of the image, as shown in Fig. 6. To ensure reliable evaluation results, we used 60 videos for training and validation, while the remaining 15 were reserved for testing.

2) Fixed-Wings Dataset: We have developed a challenging video dataset specifically for fixed-wing UAV targets. Comprising 13 video sequences with a total of 24,951 frames, all recorded at 30 FPS and 1920×1080 resolution, the dataset features numerous tiny targets against complex backgrounds, as shown in Fig. 7. Many targets closely resemble background features, making them difficult to distinguish visually. To validate our algorithm, we used a test set comprising 4,673 frames with tiny targets ranging from 1×1 to 146×95 pixels, as illustrated in Fig. 8, the average size of the test set occupies only 0.01% of the image area. The remaining 12 sequences were randomly split at 8:2 for training and validation. To our knowledge, this is one of the smallest fixed-wings UAV target datasets available.

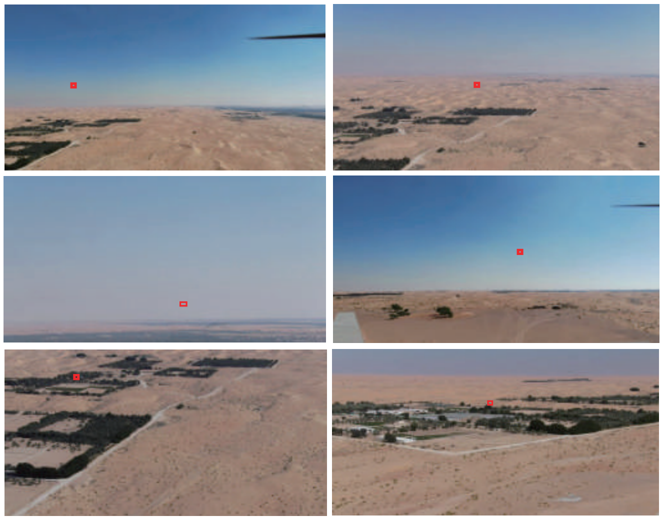


Fig. 7. Sample Images from the Fixed-Wings dataset. The red boxes highlight actual drone targets.

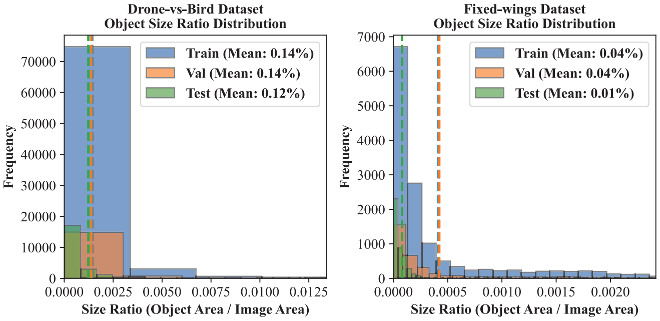


Fig. 8. Object size ratio distribution statistics of Drone-vs-Bird dataset and Fixed-wings dataset.

B. Evaluation Metrics and Implementation Details

In this study, we employed a standard evaluation metric system to quantify the performance of the detection algorithm, including Precision, Recall, AP, and two variants of mean average precision: mAP50 and mAP50-95. The experiments were conducted on a high-performance computing system equipped with two NVIDIA GeForce RTX 3090 GPUs. The image size was adjusted to 640×640 during the training phases of the model, with SGD as the optimizer, a momentum of 0.937, an initial learning rate of 0.01, 200 epochs, a batch size of 64, and an IOU threshold of 0.1 for evaluation. Table I outlines the threshold parameters in this paper, which have been validated to ensure precision and reliability.

TABLE I
THRESHOLD PARAMETER CONFIGURATION

Notation	Value	Notation	Value	Notation	Value
N_g	30	τ_g	0.3	k_1	1
N_l	60	τ_l	0.1	k_2	0.6
R_s	4/5	Δp	50	k_3	0.4

C. Comparison with Existing Works

Our study quantitatively compares the performance of the proposed GL-YOMO algorithm with several state-of-the-art methods across various datasets, with results detailed in Table II and Table III. As shown in Table II, our approach excels across multiple metrics on the Drone-vs-Bird dataset, with a Recall 4.8% higher than the second-best performer, RT-DETR [42]. Other metrics also demonstrate superior performance compared to other methods. In Table III, the performance of our method is even more pronounced on the Fixed-Wing dataset, with a 23.7% improvement in F1 score and a 25.1% increase in AP over RT-DETR.

TABLE II
COMPARISON OF THE GL-YOMO WITH STATE-OF-THE-ART METHODS ON DRONE-VS-BIRD DATESET

Method	Precision	Recall	F1-score	AP
YOLOv5s	0.824	0.756	0.789	0.761
YOLOv8s	0.837	0.682	0.752	0.744
YOLOv10s	0.822	0.662	0.734	0.702
TPH-YOLOv5	0.821	0.671	0.739	0.718
RT-DETR	0.859	0.761	0.807	**0.773**
GL-YOMO	**0.886**	**0.809**	**0.846**	0.763

TABLE III
COMPARISON OF THE GL-YOMO WITH STATE-OF-THE-ART METHODS ON FIXES-WINGS DATESET

Method	Precision	Recall	F1-score	AP
YOLOv5s	0.640	0.405	0.496	0.382
YOLOv8s	0.663	0.177	0.280	0.207
YOLOv10s	0.706	0.175	0.280	0.196
TPH-YOLOv5	0.613	0.353	0.448	0.335
RT-DETR	0.826	0.577	0.680	0.571
GL-YOMO	**0.981**	**0.861**	**0.917**	**0.822**

Analysis indicates that other methods exhibit significant shortcomings in detecting small objects, leading to a marked decline in recall, F1-score, and AP. This issue primarily arises from information loss caused by downsampling techniques, which adversely affect detection accuracy. In contrast, the GL-YOMO algorithm markedly enhances small object detection capabilities by innovatively integrating appearance and motion features. Additionally, the global-local synergistic strategy of GL-YOMO effectively preserves critical visual information of small objects and significantly reduces interference from complex backgrounds, thereby substantially improving detection performance.

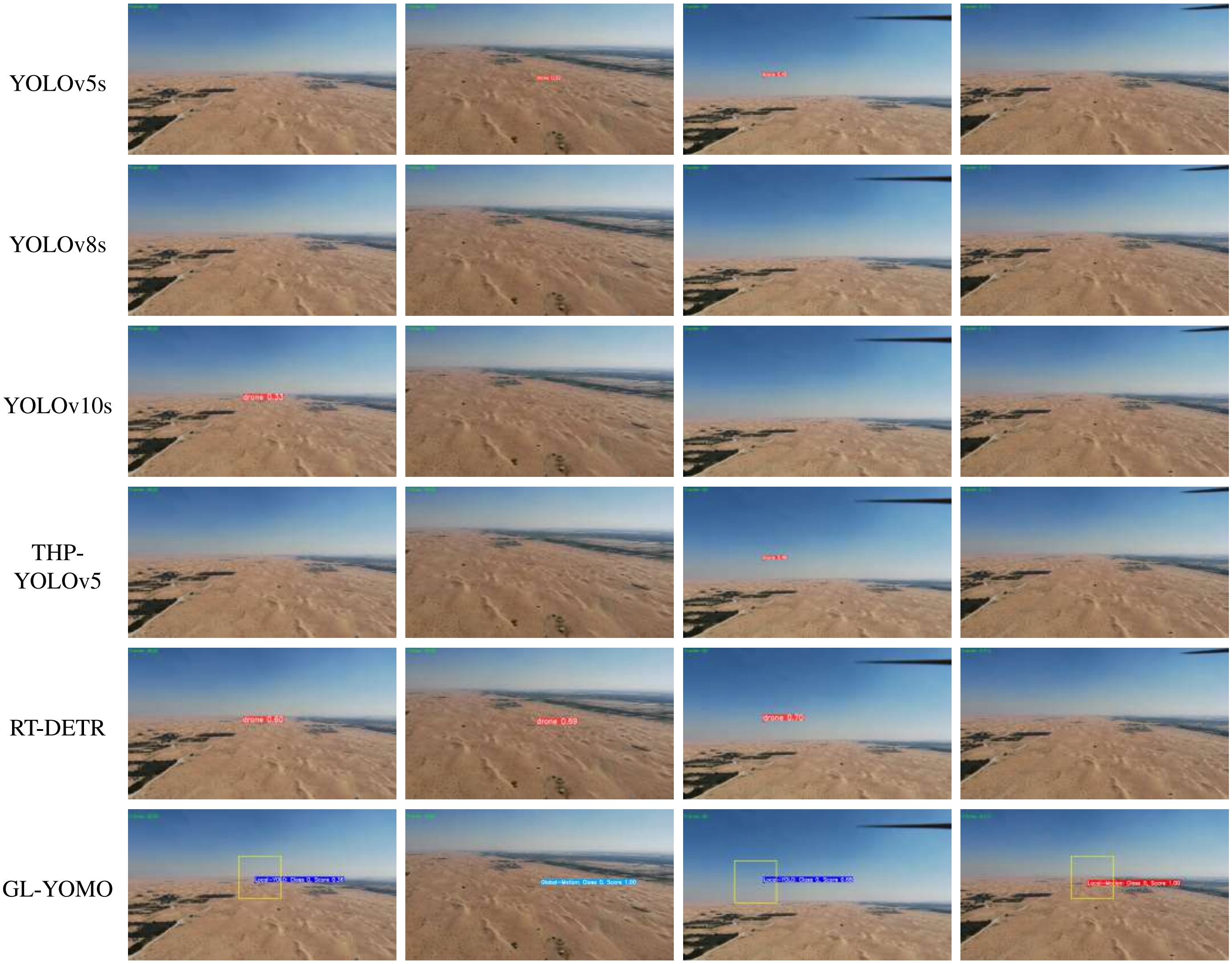


Fig. 9. Comparison of typical detection result samples achieved by various methods, with each row presenting the output of a different detection method.

As shown in Fig. 9, the GL-YOMO methods demonstrate exceptional performance in detecting extremely small targets, with the majority of targets being effectively identified through local detection strategies. Moreover, it proves highly effective in detecting particularly challenging targets. For instance, as shown in the fourth column of images, other methods fail to detect the target, while our approach successfully identifies the correct target with precision. Overall, the experimental results from both datasets underscore the effectiveness of the GL-YOMO algorithm in small object detection tasks.

D. Ablation Experiment

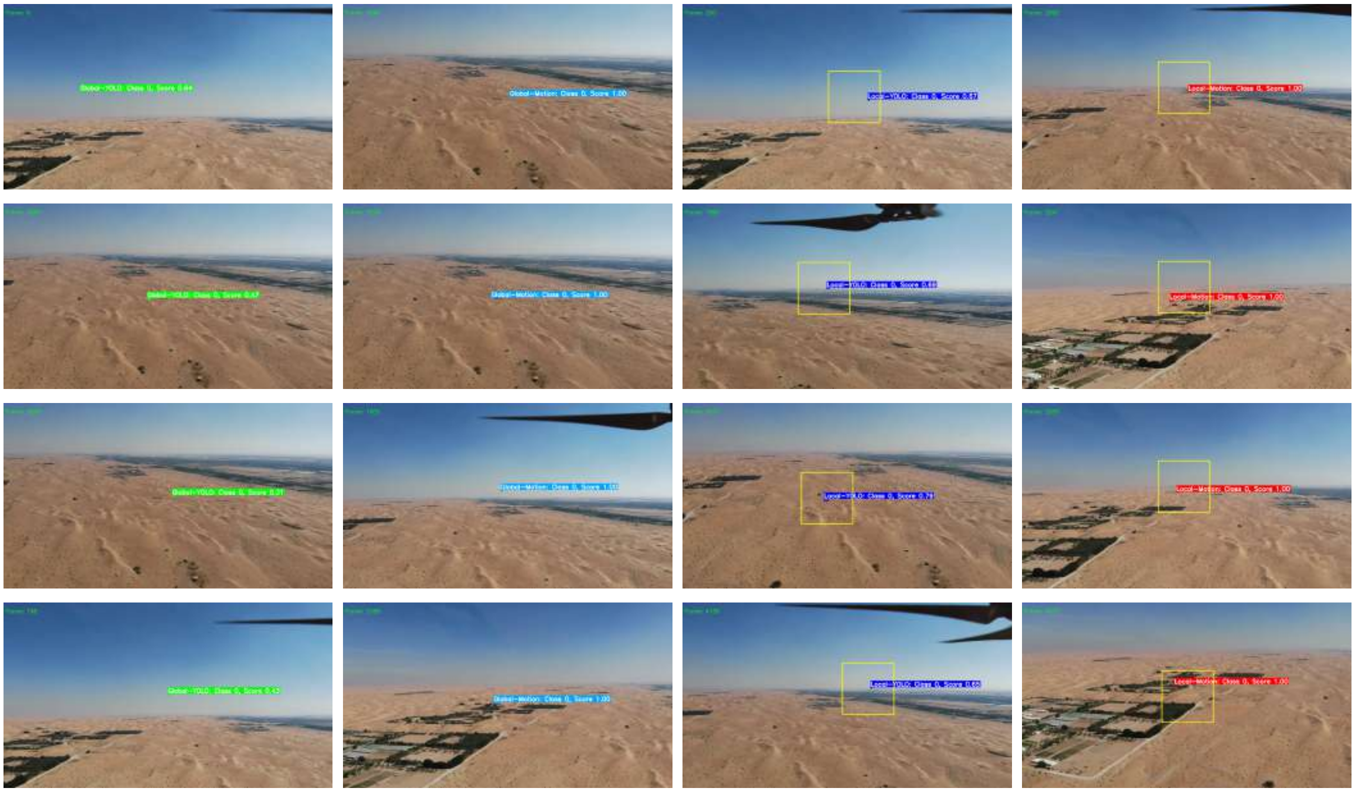
1) YOLO Improvement Experiment: To validate the performance of the improved YOLO Detector for small object detection, we conducted a comprehensive comparison with several state-of-the-art methods. TABLE IV presents the performance metrics of each model under the complete image, where ‘P’ denotes the addition of a detection head, ‘g’ indicates the use of the Ghost module for lightweight design, and ‘a’ represents the incorporation of multi-scale fusion features and attention mechanisms.

The optimized YOLO Detector significantly outperforms baseline models YOLOv5s, YOLOv8s, and YOLOv10s in recall, mAP50, and mAP50-95. The integration of a small object detection head effectively reduces information loss from downsampling, boosting recall by 4.4% and mAP by 2.6%. Attention mechanisms and multi-scale feature fusion further improve performance, with a 2% increase in mAP50-95. The Ghost module enhances efficiency, cutting model size to 14.3M while reducing parameters and GFLOPs, making it ideal for resource-limited environments. Overall, the improved YOLO Detector excels in small object detection with strong results across recall, mAP, and model size.

Furthermore, TABLE V presents a performance analysis of object detection on local image datasets, showcasing significant improvements with our YOLO Detector. Notably, it achieves the highest mAP50 of 0.880 and the best mAP50-95 score of 0.503, along with a 0.6% increase in Precision compared to YOLOv8s-pga. While there are slight differences in Recall between the YOLO Detector and YOLOv8s-pga, our detector consistently outperforms in other metrics. Compared to global image detection, local image detection exhibits

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT YOLO MODELS ON GLOBAL IMAGES

Model	p	g	a	Model Size	GFLOPs	Precision	Recall	mAP50	mAP95
YOLOv5s				14.4M	15.8	0.834	0.637	0.768	0.423
YOLOv8s				21.5M	28.4	0.84	0.377	0.448	0.241
YOLOv10s				16.5M	24.4	0.839	0.379	0.432	0.227
YOLOv5s-p	✓			25.0M	16.1	0.842	0.671	0.788	0.425
YOLOv5s-pa	✓		✓	26.0M	18.6	0.827	0.650	0.774	0.431
YOLOv8s-pg	✓	✓		10.6M	22.0	0.862	0.592	0.649	0.327
YOLOv10-pg	✓	✓		13.0M	25.1	0.835	0.569	0.628	0.317
YOLOv8s-pga	✓	✓	✓	**9.4M**	22.8	**0.863**	0.597	0.664	0.343
YOLO Detector	✓	✓	✓	14.3M	**10.6**	0.844	**0.681**	**0.794**	**0.433**


Global-YOLO: Class 0, Score 0.44
Global-Motion: Class 0, Score 1.00
Local-YOLO: Class 0, Score 0.67
Local-Motion: Class 0, Score 1.00
Global-YOLO: Class 0, Score 0.47
Global-Motion: Class 0, Score 1.00
Local-YOLO: Class 0, Score 0.69
Local-Motion: Class 0, Score 1.00
Global-YOLO: Class 0, Score 0.31
Global-Motion: Class 0, Score 1.00
Local-YOLO: Class 0, Score 0.79
Local-Motion: Class 0, Score 1.00
Global-YOLO: Class 0, Score 0.43
Global-Motion: Class 0, Score 1.00
Local-YOLO: Class 0, Score 0.65
Local-Motion: Class 0, Score 1.00

(a) Global-YOLO detection (b) Global-Motion detection (c) Local-YOLO detection (d) Local-Motion detection

Fig. 10. Some typical sample results from the Fixed-Wings dataset: Four different colors are used to distinguish the various detection outcomes: Global-YOLO results are displayed in green, Global-Motion results in light blue, Local-YOLO detections in dark blue, and Local-Motion results in red.

TABLE V
PERFORMANCE COMPARISON OF DIFFERENT METHODS ON LOCAL IMAGES

Method	Precision	Recall	mAP50	mAP50-95
YOLOv5s	**0.912**	0.804	0.873	0.491
YOLOv8s	0.898	0.804	0.826	0.442
YOLOv8s-pga	0.894	**0.816**	0.836	0.450
YOLO Detector	0.910	0.811	**0.880**	**0.503**

clear advantages, underscoring its potential for small object detection tasks and validating our approach's effectiveness in handling images with reduced background complexity.

2) Motion Detection Experiment: Table VI shows the effectiveness of the Motion Detector. Where G-YO refers to YOLO detection performed globally; G-YOMO refers to global YOLO combined with motion detection; GL-YO denotes YOLO detection at both global and local levels; and GL-YOMO represents global and local YOLO with motion detection. The results indicate that motion detection augments the sensitivity at the cost of some extra false positives, leading to a slight reduction in precision. However, this trade-off is compensated by significant improvements in other metrics. In global detection, adding a motion detector improved recall, F1-score, and AP by 5.7%, 3.9%, and 1.2%, respectively. In local detection, the addition of a motion detector also brought notable gains, with increases of 3.2% in recall and 1.5% in F1-score. These results highlight the importance and effectiveness of motion detectors in object detection tasks, significantly enhancing performance in both global and global-local collaborative detection modes.

TABLE VI
IMPACT OF MOTION DETECTOR INTEGRATION ON DETECTION PERFORMANCE

Method	Precision	Recall	F1-Score	AP
G-YO	0.968	0.560	0.709	0.553
G-YOMO	0.948	0.617	0.748	0.565
GL-YO	**0.989**	0.829	0.902	**0.828**
GL-YOMO	0.981	**0.861**	**0.917**	0.822

E. Inference Time

To comprehensively evaluate the deployment performance of our method on edge computing devices, we selected the NVIDIA Jetson Xavier NX as our testing platform. By leveraging TensorRT for model optimization and acceleration, we achieved an average frame rate of 21.6 FPS using a 640×640 resolution model for inference on the test video, meeting the requirements for real-time applications. Fig. 10 illustrates some typical detection results of our method on the Fixed-Wings dataset.

V. CONCLUSIONS

In this paper, we proposed GL-YOMO, an advanced real-time detection method designed to tackle the challenges of detecting small-pixel UAV targets at long distances. GL-YOMO combines the YOLO detection algorithm with multi-frame motion analysis to achieve high-precision UAV detection. Our approach utilizes a global and local collaborative detection strategy: initially performing broad target localization, followed by an adaptive mechanism that refines the detection range for more precise local analysis. This method not only improves detection accuracy but also allows seamless transitions back to global detection when local detection struggles, enhancing overall system performance. Future work will focus on optimizing GL-YOMO for multi-UAV detection to handle more complex environments.

REFERENCES

[1] A. N. Sayed, O. M. Ramahi, and G. Shaker, "Machine learning for uav classification employing mechanical control information," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 60, no. 1, pp. 68–81, 2023.

[2] S. Wagner, W. Johannes, D. Qosja, and S. Brüggenwirth, "Small target detection in a radar surveillance system using contractive autoencoders," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 60, no. 1, pp. 51–67, 2023.

[3] T. Sangam, I. R. Dave, W. Sultani, and M. Shah, "Transvisdrone: Spatio-temporal transformer for vision-based drone-to-drone detection in aerial videos," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 6006–6013.

[4] H. Guo, Y. Zheng, Y. Zhang, Z. Gao, and S. Zhao, "Global-local mav detection under challenging conditions based on appearance and motion," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[5] M. W. Ashraf, W. Sultani, and M. Shah, "Dogfight: Detecting drones from drones videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7067–7076.

[6] Y. Chen, Y. Cao, H. Hu, and L. Wang, "Memory enhanced global-local aggregation for video object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 337–10 346.

[7] J. Xie, J. Yu, J. Wu, Z. Shi, and J. Chen, "Adaptive switching spatial-temporal fusion detection for remote flying drones," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 6964–6976, 2020.

[8] A. Rozantsev, V. Lepetit, and P. Fua, "Detecting flying objects using a single moving camera," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 5, pp. 879–892, 2016.

[9] B. K. Isaac-Medina, M. Poyser, D. Organisciak, C. G. Willcocks, T. P. Breckon, and H. P. Shum, "Unmanned aerial vehicle visual detection and tracking using deep neural networks: A performance benchmark," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1223–1232.

[10] X. Zhou, G. Yang, Y. Chen, C. Gao, B. Zhao, L. Li, and B. M. Chen, "Admnet: Anti-drone real-time detection and monitoring," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 3009–3016.

[11] M. Zhu and E. Kong, "Multi-scale fusion uncrewed aerial vehicle detection based on rt-detr," *Electronics*, vol. 13, no. 8, p. 1489, 2024.

[12] W. Weihua, W. Peizao, and N. Zhaodong, "A real-time detection algorithm for unmanned aerial vehicle target in infrared search system," in *2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. IEEE, 2018, pp. 1–5.

[13] Y. Zheng, Z. Chen, D. Lv, Z. Li, Z. Lan, and S. Zhao, "Air-to-air visual detection of micro-uavs: An experimental evaluation of deep learning," *IEEE Robotics and automation letters*, vol. 6, no. 2, pp. 1020–1027, 2021.

[14] S. Srigrarom and K. H. Chew, "Hybrid motion-based object detection for detecting and tracking of small and fast moving drones," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 615–621.

[15] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "Yolov9: Learning what you want to learn using programmable gradient information," *arXiv preprint arXiv:2402.13616*, 2024.

[16] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "Yolov10: Real-time end-to-end object detection," *arXiv preprint arXiv:2405.14458*, 2024.

[17] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 2778–2788.

[18] K. Tan, S. Ding, S. Wu, K. Tian, and J. Ren, "A small object detection network based on multiple feature enhancement and feature fusion," *Scientific Programming*, vol. 2023, no. 1, p. 5500078, 2023.

[19] H. Liu, F. Sun, J. Gu, and L. Deng, "Sf-yolov5: A lightweight small object detection algorithm based on improved feature fusion mode," *Sensors*, vol. 22, no. 15, p. 5817, 2022.

[20] C. Wang, W. He, Y. Nie, J. Guo, C. Liu, Y. Wang, and K. Han, "Gold-yolo: Efficient object detector via gather-and-distribute mechanism," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[21] M. Kang, C.-M. Ting, F. F. Ting, and R. C.-W. Phan, "Asf-yolo: A novel yolo model with attentional scale sequence fusion for cell instance segmentation," *Image and Vision Computing*, vol. 147, p. 105057, 2024.

[22] J. Yang, S. Deng, F. Zhang, A. Pan, and Y. Yang, "Fatcnet: Feature adaptive transformer and cnn for infrared small target detection," *IEEE Transactions on Aerospace and Electronic Systems*, 2024.

[23] S. Liu, Y. Wang, Q. Yu, H. Liu, and Z. Peng, "Ceam-yolov7: Improved yolov7 based on channel expansion and attention mechanism for driver distraction behavior detection," *IEEE Access*, vol. 10, pp. 129 116–129 124, 2022.

[24] H. Gong, T. Mu, Q. Li, H. Dai, C. Li, Z. He, W. Wang, F. Han, A. Tuniyazi, H. Li *et al.*, "Swin-transformer-enabled yolov5 with attention mechanism for small object detection on satellite images," *Remote Sensing*, vol. 14, no. 12, p. 2861, 2022.

[25] B. Yan, J. Li, Z. Yang, X. Zhang, and X. Hao, "Aie-yolo: Auxiliary information enhanced yolo for small object detection," *Sensors*, vol. 22, no. 21, p. 8221, 2022.

[26] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.

[27] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1222–1230.

[28] Q. Wang, H. Xu, S. Lin, J. Zhang, W. Zhang, S. Xiang, and M. Gao, "A low-slow-small uav detection method based on fusion of range doppler map and satellite map," *IEEE Transactions on Aerospace and Electronic Systems*, 2024.

[29] B. Aydin and S. Singha, "Drone detection using yolov5," *Eng*, vol. 4, no. 1, pp. 416–433, 2023.

[30] S. Singha and B. Aydin, “Automated drone detection using yolov4,” *Drones*, vol. 5, no. 3, p. 95, 2021.

[31] H. Cai, Y. Xie, J. Xu, and Z. Xiong, “A lightweight and accurate uav detection method based on yolov4,” *Sensors*, vol. 22, no. 18, p. 6874, 2022.

[32] U. Seidaliyeva, D. Akhmetov, L. Ilipbayeva, and E. T. Matson, “Real-time and accurate drone detection in a video with a static background,” *Sensors*, vol. 20, no. 14, p. 3856, 2020.

[33] A. Coluccia, A. Fascista, L. Sommer, A. Schumann, A. Dimou, D. Zarpalas, and N. Sharma, “Drone-vs-bird detection grand challenge at icassp2023,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–2.

[34] H. Wang, X. Wang, C. Zhou, W. Meng, and Z. Shi, “Low in resolution, high in precision: Uav detection with super-resolution and motion information extraction,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[35] J. Li, D. H. Ye, M. Kolsch, J. P. Wachs, and C. A. Bouman, “Fast and robust uav to uav detection and tracking from video,” *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 3, pp. 1519–1531, 2021.

[36] J. Li, D. H. Ye, T. Chung, M. Kolsch, J. Wachs, and C. Bouman, “Multi-target detection and tracking from a single camera in unmanned aerial vehicles (uavs),” in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016, pp. 4992–4997.

[37] L. Du, C. Gao, Q. Feng, C. Wang, and J. Liu, “Small uav detection in videos from a single moving camera,” in *Computer Vision: Second CCF Chinese Conference, CCCV 2017, Tianjin, China, October 11–14, 2017, Proceedings, Part III*. Springer, 2017, pp. 187–197.

[38] C. Wang, T. Wang, E. Wang, E. Sun, and Z. Luo, “Flying small target detection for anti-uav based on a gaussian mixture model in a compressive sensing domain,” *Sensors*, vol. 19, no. 9, p. 2168, 2019.

[39] C. Craye and S. Ardjoune, “Spatio-temporal semantic segmentation for drone detection,” in *2019 16th IEEE International conference on advanced video and signal based surveillance (AVSS)*. IEEE, 2019, pp. 1–5.

[40] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, “Ghostnet: More features from cheap operations,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1580–1589.

[41] Q.-V. Tran, S.-F. Su, and V.-T. Nguyen, “Pyramidal lucas—kanade-based noncontact breath motion detection,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 7, pp. 2659–2670, 2018.

[42] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, “Detrs beat yolos on real-time object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16 965–16 974.