# A Classwise Vulnerable Part Detection Method for Military Targets

Hanyu Wang, Qiang Shen, Juan Li, Zihao Chen, Yiran Guo, and Shouyi Zhang

***Abstract*—Accurate vulnerable part detection based on full target detection results shows great importance in improving the damage effectiveness of the military drone. However, traditional object detection methods have difficulty in handling inaccurate full target bounding boxes and fail to model the semantic relationships between various class full targets and their key parts, resulting in low localization accuracy. The proposed approach includes a classwise feature recalibration module, which effectively models the dependencies between the prior knowledge obtained from the full target detector and the location of the key part. Additionally, an optimized spatial transformation module is designed to preprocess the input image and eliminate interfering objects. Furthermore, a carefully constructed loss function is employed, linking the classification branch with the regression branch, thereby emphasizing the importance of localization accuracy. Our proposed model surpasses the performance of existing state-of-the-art models, demonstrating a significant advantage with maximum improvements of +24.9%, +30.2%, and +28.3% in mean average precision on the standard test set, generalized test set, and real-world dataset, respectively. The effectiveness and robustness are also confirmed through extensive ablation studies.**

***Index Terms*—Deep learning, key parts, military targets, prior knowledge.**

## I. Introduction

WITH the advance of unmanned aerial vehicles (UAVs) equipped with computer vision systems, object detection based on deep learning plays an important role in various military tasks, such as precision strikes, reconnaissance, and situational awareness [1], [2], [3]. However, taking antitank as an example, the number of tanks damaged as a result of hitting the turret was nearly nine times higher than that hit by other parts during the 2014–2015 armed conflict in Ukraine [4]. Therefore, it is of utmost importance and significance to delve deeper into the detection method for specific key parts based on UAV images, leveraging the detection results obtained from the full target detector.

Currently, object detection tasks in military scenarios heavily rely on human-in-the-loop approaches. With the advancements in deep learning, research efforts toward intelligent recognition have gained notable momentum. Qin et al. [5] developed a multilayer feature extraction network specifically tailored for military ship detection from satellite images, achieving an average precision of 97.0%. Kong et al. [6] proposed a lightweight network for detecting military targets in complex backgrounds, reaching an average accuracy of 89.3% in a real scene dataset.

Nevertheless, the identification of critical parts of military targets still poses significant challenges even after achieving full target detection. First, each target exhibits unique key parts. For instance, in the case of an infantry fighting vehicle being detected by the full target detector, the key part detector should accurately locate its tracks. Similarly, when an airplane is identified, the engine should be specifically highlighted. Our model needs to possess the ability to capture these diverse relationship pairs, which necessitates a high level of adaptive perceptual capability. The classification information, which decides the location of the key part, is already obtained from the full target detector but is ignored totally in the traditional detection networks, thus decreasing the localization accuracy. Additionally, the background of the image during the attack is often complex and full of disruptive objects, which will divert the attention of the model during training, especially for the cropped images that contain only a limited number of pixels. Therefore, the extraction accuracy of key parts highly depends on the quality of the detected bounding box of the full target, where small positioning errors are inevitable. The classic methods struggle to adaptively process these cropped images, resulting in a model that is highly susceptible to the influence of interfering objects and exhibits low detection efficiency. Third, as previously mentioned, the effective damage to the target relies on accurately hitting its key parts. As such, irrespective of the classification accuracy, a low localization accuracy would render the attack invalid. Consequently, our military target critical part detection task places greater emphasis on achieving high localization accuracy. However, the cross-entropy loss for classification is independent of the localization task and leads the model to learn all the positive anchors with high classification scores regardless of regression accuracy, which hurts the localization accuracy seriously [7]. Hence, the special optimization of the loss function is necessary for the key part detector. Finally, the stringent constraints on computational and storage resources

Manuscript received 15 January 2024; revised 10 March 2024; accepted 3 April 2024. Date of publication 16 April 2024; date of current version 29 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 61973033 and in part by the Natural Science Foundation of Chongqing, China under Grant cstc2021jcyj-msxmX0737. *(Corresponding author: Qiang Shen.)*

Hanyu Wang and Qiang Shen are with the School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China, and also with the Chongqing Innovation Center, Beijing Institute of Technology, Chongqing 401120, China (e-mail: 3120215124@bit.edu.cn; bit82shen@bit.edu.cn).

Juan Li, Zihao Chen, Yiran Guo, and Shouyi Zhang are with the School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China (e-mail: juanli@bit.edu.cn; 3220210102@bit.edu.cn; 3120220188@bit.edu.cn; 3220220170@bit.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2024.3389072

in military scenarios exacerbate the aforementioned three challenges.

To address the multiple challenges mentioned above, we propose a classwise key part detection model based on Faster RCNN [8], namely CWKPD-Faster RCNN. Our main contribution can be summarized as follows.

1) We proposed a new module to model the semantic relationship between the full target and the key part, thus improving the detection accuracy of the key part. This simple yet efficient approach offers the following advantages:
   a) versatility in handling diverse “full target-key parts” pairs;
   b) improved discriminative power;
   c) efficient computational overhead.

As far as we know, we are the first to consider the relationship between the full target and the key part.

2) We developed a novel spatial transformation module, which not only neatly reduces the localization error brought by the full target detector but also resolves the limitations of the fixed image size imposed by traditional spatial transformation methods. Additionally, a weighted classification loss function that enhances the correlation between classification and localization tasks is also built to move the model to more focus on the localization branch.
3) We have constructed a new simulated dataset comprising military targets observed by UAVs, expanding the diversity of existing UAV-looked military target datasets. Through a comprehensive evaluation of the simulated and real-world datasets, the superiority of the proposed CWKPD-Faster RCNN model, not only in precision but also in generalization ability, has been demonstrated.

## II. Related Works

### A. Key Part Identification

Nowadays, there are three main solutions for detecting key parts: template matching [9], semantic segmentation [10], and key parts localization method based on the deep object detection frameworks [11]. Template matching, as the traditional method, has the advantage of rapid detection, but relies on the precise expression of preinstalled templates, which shows poor accuracy and robustness when facing diverse target shapes and shooting angles, or partially occluded targets. Meanwhile, the semantic network understands all the pixels in the whole image, instead of concentrating only useful parts, resulting in poor real-time performance. Therefore, researchers preferred the third key part localization method based on the deep object detection frameworks. Nadeem et al. [12] proposed a top-down body part detection framework considering the intrinsic correlation among body parts. Hao et al. [13] proposed an improved YOLOv4 network for parts detection in insulator images during UAV inspection. Liu et al. [14] designed a novel pipeline based on YOLO and OpenPose, enabling real-time detection of hands from human images captured by drones. Choi et al. [15] constructed a modified VGG16 model to detect structural cracks in buildings from UAV images. Similarly, MUENet [16] describes a crack detection algorithm from road images acquired via UAVs, which fully utilizes the morphology and color characteristics of cracks. However, these methods solely focus on the detection of key parts for individual class targets, without explicitly modeling the diverse semantic relationships that exist between different class targets and their corresponding key parts. The research for identifying different specific parts of multiclass targets has rarely been studied directly. Furthermore, while certain studies have made efforts to incorporate the dependencies between the full target and the key part into the key part detector, these attempts heavily rely on the accurate representations of shapes. Consequently, this approach not only compromises the robustness of complex scenes but also suffers from limitations, such as feature dimensionality and computational complexity. The applicability of these models is limited to specific part detection problems, so the direct generalization to others is impractical. Therefore, the primary objective of this article is to explore a universal and efficient method for modeling the semantic relationship between full targets and their associated key parts.

Meanwhile, the key part detection task in military scenarios also faces typical challenges encountered in object detection in UAV images, such as domain shift, small objects, and limited real-time performance. Many highly inspiring studies [17], [18], [19] are proposed to alleviate these problems, which hold immense significance in fostering the progress of drone image object detection in various fields. For instance, Biswas et al. [20] substantially reduced the number of learnable parameters of YOLOv5 through compressed convolutional techniques, transfer learning, and backbone shrinkage. In contrast, this article will concentrate on the specific challenge of identifying key parts of military targets and aim to explore the potential benefits of incorporating the prior classification knowledge in enhancing the effectiveness of UAV object detection.

### B. Informed Machine Learning

One way of recovering from the first problem is to regard the classification results as prior knowledge and then integrate it into learning systems. Formally, the knowledge can be represented as logical rules, simulation results, graphs, etc., and can be integrated into the networks via distinct ways, such as additional loss terms, serving as a consistency check, and forming adjacency matrices in gated graph neural networks [21]. Chen et al. [22] converted the prediagnosis results based on the structural analysis into correlation graphs to improve the diagnosis accuracy. Zhao et al. [23] proposed an interpretable, weakly supervised deep learning framework incorporating prior knowledge to decode pathological images. However, the emphasis of our model is to utilize the strong correlation between knowledge and predictions to drive the model to adoptively focus on vulnerable areas, which is not suitable for the representation approaches of logical rules and simulation results. Graph neural networks can implicitly model semantic relationships, but are computationally intensive and require a large number of model parameters. How to solve this problem in a lightweight way to improve the model prediction accuracy and robustness is a major interest of this article.

### C. Spatial Transformation Models

Various adaptive transformation models based on deep learning have been studied to overcome the second challenge. Xiong et al. [24] built the generative adversarial network to acquire images from different viewpoints. Ding et al. [25] proposed the affine variational autoencoder to learn the transforms explicitly in the latent space. Jaderberg et al. [26] proposed the spatial transformer network (STN) model, which learns the affine transformation parameters in an unsupervised way and then automatically adjusts the image. Compared to the first two methods, the STN model demonstrates more simplicity and efficiency, which shows good performance in human pose estimation, hyperspectral image classification, and pedestrian heading estimation [27], [28]. However, the STN model requires a fixed input image scale, which contradicts target detection tasks. The direct integration of the STN model into the detection model is anticipated to result in a decrease in accuracy.

### D. Accurate Localization

Improving localization accuracy continues to be a significant focus in the field of object detection. Increasing the IoU threshold for positive objects is a well-established and powerful trick, which also introduces a higher risk of false detections [29]. Moreover, the threshold needs to be adjusted manually for different tasks and datasets, limiting its flexibility. In addition, incorporating contextual information has shown the potential to enhance localization accuracy, which introduces additional computational overhead and renders the model vulnerable to interference from complex backgrounds, unfortunately [30]. In contrast, establishing the correlation between classification and localization tasks through a loss function is a more concise and efficient approach. Wu et al. [31] appended an IoU prediction branch, and Wu et al. [32] designed an IoU balance loss consisting of an IoU balance classification loss function and an IoU balance localization loss function. Li et al. [33] derived a power $\alpha$-Gaussian loss to avoid boundary discontinuity in rotated bounding box regression. However, rare methods balance between the localization accuracy improvement and increased model complexity, especially based on two-stage object detectors.

## III. Methods

A classwise detection model started with Faster RCNN is proposed, whose framework is shown in Fig. 1. First, after being cropped based on the regression results of the full target detector, an image patch is passed into the spatial transformer module optimized by spatial pyramid pooling (SPP) [34], named as STMO-SPP, to be preprocessed. Then, the feature map is generated through the backbone network. Third, a classwise feature recalibrated (CWFR) module is designed to model the classification results from the full target detector as channel descriptors and learn the strong correlation between prior knowledge and the detection results. The feature map is reorganized. Next, the recalibrated feature maps are passed into the region proposals network (RPN), the classification, and regression branches sequentially. Additionally, the proposed weighted classification loss function and localization loss function are used to guide the training process.

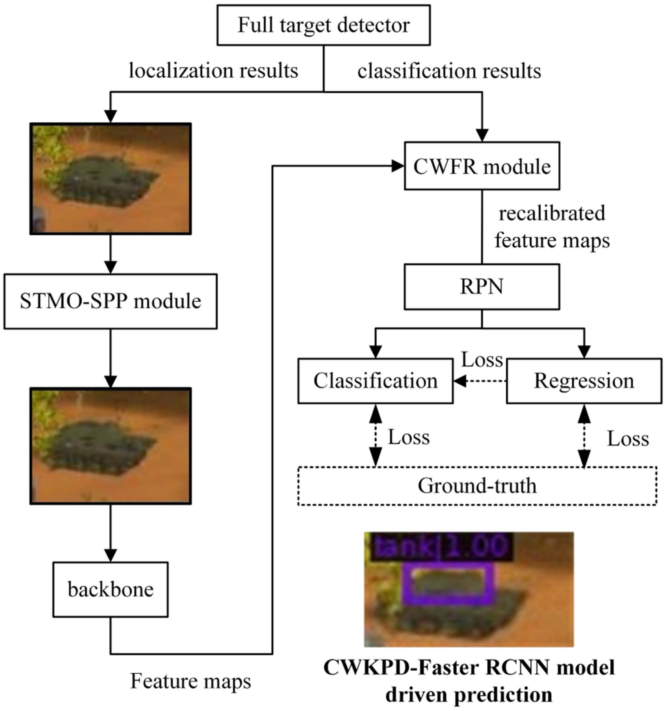


Fig. 1. Network structure of the proposed method.

### A. Spatial Transformer Module Optimized by SPP Layer

As shown in Fig. 2, the structure of the STMO-SPP module consists of three parts. The first part is a localization net optimized by SPP, whose input is the cropped image based on the results of the full target detectors. This part is aimed at acquiring the transformation matrix $A_\theta$ whose shape is determined by the expected transformation types. 2-D affine transformation is chosen in our model, which could represent transformations such as translation, rotation, scaling, and flips.

Table I presents the architecture of the built localization net optimized by SPP. Taking images with the size of 96 × 128, and 128 × 128 as examples, the transformation process of input images and the flexible adjustment process of parameters in the pooling layers are demonstrated. A convolutional layer with a kernel of 7 × 7 is used in the first place, to adjust the number of feature channels and extract salient features. Then, the feature map is down-sampled by a maximum pooling layer, where the sliding window size is 2 and stride size is 2, reducing the mean shift error caused by the convolutional layer and preserving more texture information. Subsequently, the ReLu activation function is adapted to perform a nonlinear transformation on the extracted features. These three steps are repeated to acquire a small-scale

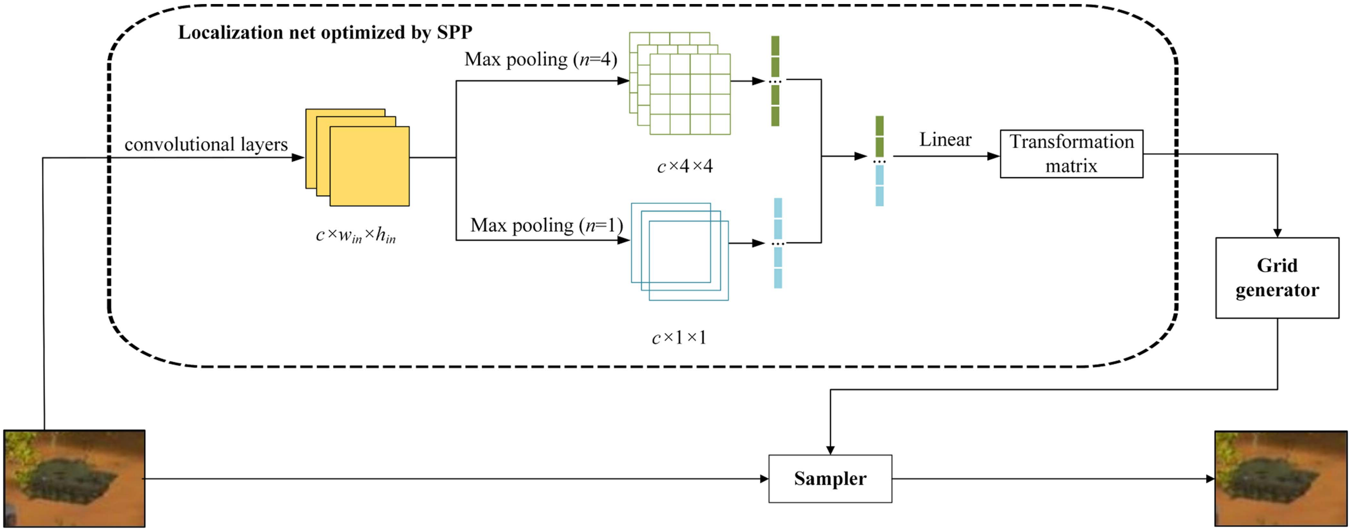


Fig. 2. Architecture of the proposed STMO-SPP block.

TABLE I
STRUCTURE OF THE LOCALIZATION NET

| Number | Type | Filter 1 | Stride 1 | Output 1 | Filter 2 | Stride 2 | Output 2 |
|---|---|---|---|---|---|---|---|
| 0 | Input | - | - | 3×96×128 | - | - | 3×128×128 |
| 1 | Convolutional | 7×7 | 1 | 24×90×122 | 7×7 | 1 | 24×122×122 |
| 2 | Max pooling | 2 | 2 | 24×45×61 | 2 | 2 | 24×61×61 |
| 3 | Convolutional | 5×5 | 1 | 30×41×57 | 5×5 | 1 | 30×57×57 |
| 4 | Max pooling | 2 | 2 | 30×20×28 | 2 | 2 | 30×28×28 |
| 5 | Max pooling1 (n=1) | 20×28 | 20×28 | 30×1×1 | 28×28 | 28×28 | 30×1×1 |
| 5 | Max pooling2 (n=4) | 5×7 | 5×7 | 30×4×4 | 7×7 | 7×7 | 30×4×4 |
| 5 | Concatenating | - | - | 510 | - | - | 510 |
| 6 | Linear | - | - | 32 | - | - | 32 |
| 7 | Linear | - | - | 6 | - | - | 6 |

feature map retaining crucial information. At this time, the size of the output feature map is not fixed and cannot be directly passed into the fully connected layer.

Therefore, we introduced SPP to generate fixed-size feature vectors. The size of the feature map is represented as $(c, h_{\text{in}}, w_{\text{in}})$, where *c* is the number of channels, $h_{\text{in}}$ is height, and $w_{\text{in}}$ denotes width. As shown in Number 5 of the model structure in Table I, the parameters of the pooling layer, including the size of the sliding window, $k_h, k_w$, and the size of the stride, $s_h, s_w$, are adjusted adaptively to keep the output size as $(c, n, n)$. The subscripts *h* and *w* denote the height and width directions. The relationship between pooling layer parameters and input and output sizes is modeled as

$$\begin{aligned} k_h &= \lceil h_{in}/n \rceil \\ k_w &= \lceil w_{in}/n \rceil \\ s_h &= \lfloor h_{in}/n \rfloor \\ s_w &= \lfloor w_{in}/n \rfloor \end{aligned} \tag{1}$$

where *n* is selected as 1 and 4 in our framework, and $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$denote ceiling and floor operations.

Once the feature vector of a predefined size is obtained, it is subsequently passed through the fully connected layers to generate the affine transformation matrix, comprising six parameters.

The second part is the grid generator, which achieved the coordinates of each pixel in the transformed image, through parameterized grid sampling and affine transformation

$$\begin{bmatrix} u \\ v \end{bmatrix} = A_\theta \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2}$$

where $(x, y)$ is the coordinate in the input image and $(u, v)$ is that of the output correspondingly.

The sampler, as the last part of the proposed STMO-SPP module, corrects the coordinates of the points generated by the gird generator based on the Bilinear interpolation method, ensuring the pixels in the output image are always integer values.

### B. Classwise Feature Recalibration Module

The proposed CWFR module is a computational unit, to improve the quality of representations produced by the network, by explicitly modeling the interdependencies between the prior class results of the full target $x_{\text{class}}$ and the channels of the feature map. It can be built upon a transformation $G_{\text{tr}}$, mapping an input $X \in R^{H\times W\times C}$ to $\tilde{X} \in R^{H\times W\times C}$

$$\tilde{X} = G_{\text{tr}}(g, X)$$

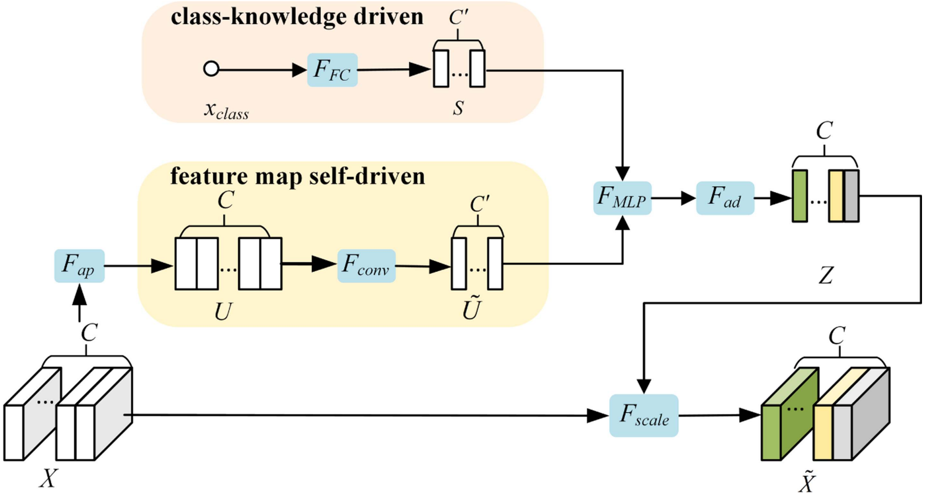


Fig. 3. Structure of the proposed classwise feature recalibration module.

$$g = \Gamma(X, x_{\text{class}}). \tag{3}$$

As illustrated in Fig. 3, we first use global average pooling $F_{\text{ap}}$ to represent global spatial information into a channel descriptor $U \in R^{1\times 1\times C}$. The input $X$ and output $U$ are denoted in channel dimension as

$$\begin{aligned} X &= \begin{bmatrix} x_1 & x_2 & \cdots & x_c \end{bmatrix} \\ U &= \begin{bmatrix} u_1 & u_2 & \cdots & u_c \end{bmatrix}. \end{aligned} \tag{4}$$

The conversion is derived as

$$\begin{aligned} u_n &= F_{\text{ap}}(x_n) \\ &= \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i,j), n = 1, 2, \ldots, C. \end{aligned} \tag{5}$$

Subsequently, the high-dimensional feature representation $\tilde{U} \in R^{1\times 1\times C'}$ is acquired by convolving all channels through $F_{\text{conv}}$.

In the other parallel branch, we opt to employ a simple gating mechanism with a sigmoid activation to process the input prior knowledge $x_{\text{class}}$, which is converted to a channel descriptor $s \in R^{1\times 1\times C'}$

$$\begin{aligned} s &= F_{\text{FC}}(x_{\text{class}}, K) = \sigma(g(x_{\text{class}}, K)) \\ &= \sigma_2\left(K_2 \sigma_1\left(K_1 x_{\text{class}}\right)\right) \end{aligned} \tag{6}$$

where the gating mechanism is designed as two fully connected layers, to limit model complexity and aid generalization. In (6), $\sigma_1$ refers to the ReLu activation function and $\sigma_2$ is the sigmoid activation function. Meanwhile, $K_1$ and $K_2$ are adjustable parameters for two activation functions, to increase the flexibility and expressive power of the model.$K_1 \in R^{r\times 1}$ and $K_2 \in R^{1\times r}$, and $r$denotes the expansion ratio.

Then, we follow it with a multilayer perceptron (MLP) operation, $F_{\text{MLP}}$, which consists of two linear transformation layers and two nonlinear activation functions, intending to perfectly learn the channelwise features as well as the interdependences between the class feature $s$ and the localization results

$$\begin{aligned} Z_0 &= F_{\text{MLP}}(s, \tilde{U}) = \sigma_2 \\ &\left(W_3 \sigma_1\left(W_1 s + W_2 \tilde{U} + b_1\right) + W_4 \tilde{U} + b_2\right) \end{aligned} \tag{7}$$

where $W_1, W_2, W_3,$ and $W_4$ represent weight matrices, whereas $b_1$ and $b_2$ denote bias vectors.

Afterward, the dimension-addition layer $F_{\text{ad}}$ is performed to rescale the output to the same dimension as input X, obtaining the feature descriptor related to class knowledge Z

$$Z = F_{\text{ad}}(Z_0), Z \in R^{1\times 1\times C}. \tag{8}$$

Formally, $Z$ is also denoted as

$$Z = \begin{bmatrix} z_1 & z_2 & \cdots & z_c \end{bmatrix}. \tag{9}$$

Finally, the output is reorganized by $Z$

$$\tilde{x}_n = F_{\text{scale}}(x_n, z_n) = z_n x_n, n = 1, 2, \ldots, C. \tag{10}$$

It is noted that in the above operations, the reason that the mapping is in the channel instead of the spatial or the triple, including spatial and channel, is explained and verified in Section VI.

### C. Loss Function

In the Faster RCNN model, the smoothed L1 loss function and the cross-entropy loss function are used to supervise the regression and classification branches, respectively, which are isolated from each other. To enhance the focus of the model on localization tasks, we proposed the weighted classification loss function, correlating with the localization accuracy

$$\begin{aligned} & w_i\left(\text{GIoU}_i\right) = 1 - \text{GIoU} \\ & \text{CE}\left(p_i, \hat{p}_i\right) = -\log\left[p_i \hat{p}_i + (1 - \hat{p}_i)(1 - p_i)\right] \\ & L_{\text{wcls}} = \frac{\sum_{i\in\text{Pos}}^{N} w_i\left(\text{GIoU}_i\right) * \text{CE}\left(p_i, \hat{p}_i\right) + \sum_{i\in\text{Neg}}^{M} \text{CE}\left(p_i, \hat{p}_i\right)}{N_{\text{cls}}} \end{aligned} \tag{11}$$

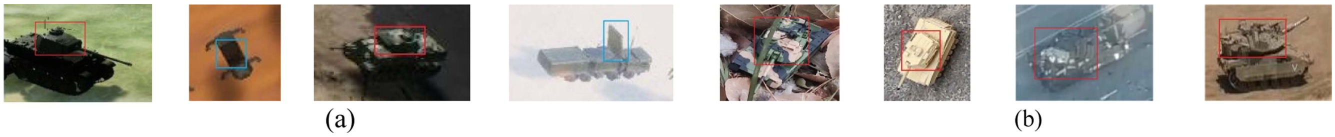


Fig. 4. Examples of the dataset. (a) ULMT-Key dataset. (b) Real-world dataset.

TABLE II
DATASETS SETTINGS

| Dataset | Type | Terrain | Number of Unique Appearances | | Numbers |
|---|---|---|---|---|---|
| | | | Tank | RV | |
| ULMT-Key dataset | Training set | grass, desert, military camps | 4 | 1 | 6019 |
| | Validation set | grass, desert, military camps | 4 | 1 | 752 |
| | Standard test set | grass, desert, military camps | 4 | 1 | 752 |
| | Generalized test set | beach | 7 | 2 | 700 |
| Real-world dataset | Equivalent scaling test | grass, snowfield, Urban Road | | | |
| | Flight test | desert, military camps | / | / | 300 |
| | Satellite images | military camps | | | |

where Pos and Neg represent the sets of positive and negative training samples, respectively, $\mathrm{CE}(p_i, \hat{p}_i)$ denotes the cross-entropy loss function, $p_i$ indicates the predicted probability of each anchor box being a positive sample while $\hat{p}_i$ is the ground truth, and $w_i(\mathrm{GIoU}_i)$ is the results of the GIoU loss function for each positive sample, as the weight assigned to the corresponding results of the cross-entropy loss function. Equation (11) demonstrates that samples with GIoU values are assigned higher weights, resulting in larger gradients during training. This weighted function also facilitates the model in effectively learning the interdependencies between classification and localization tasks.

The localization branch is penalized by GIoU loss functions, and the total loss function is the sum of those

$$L_{\mathrm{loc}} = \frac{1}{N_{\mathrm{loc}}} \sum_i (1 - \mathrm{GIoU}) \tag{12}$$

$$L_{\mathrm{total}} = L_{\mathrm{wcls}} + \lambda L_{\mathrm{loc}}. \tag{13}$$

## IV. Experimental Setup

### A. Dataset

*1) Simulated Dataset:* Considering that establishing a large-scale military target dataset in real-world scenarios is unrealistic, we referred to the dataset-making methods in fields such as autonomous driving and drone SLAM [35], [36] and constructed a UAV-looked military targets (ULMT) dataset through Unreal Engine. The ULMT dataset contains four types of terrain: grass, desert, military camps, and beach, each of which consists of two classes of targets with 2–5 kinds of appearances: tanks and radar vehicles (RV).

The dataset employed for validating key part detection, named the ULMT-Key dataset, is obtained through cropping images in the ULMT dataset, based on the prediction results of the full target detector, which is chosen as the Faster RCNN. The size of the cropped images varies from (41 43) to (658 566). The vulnerable part of a tank target is defined as the top turret part, while that of an RV target is defined as the antenna part. Examples of the ULMT-Key dataset and their labels are illustrated in Fig. 4(a).

To assess the generalization capability of our model, the test set of the ULMT-Key dataset is categorized into two types: the standard test set, which maintains consistency in terrains and target appearances with the training set, and the generalized test set, which introduces variations. The detailed settings are shown in Table II. In addition, the training, validation, and standard test sets are divided through random sampling [37], allocating 80% of the data to the training set and 10% each to the validation and standard test sets. Meanwhile, the ratio of numbers of the two classes of objects is 3:2 in all sets.

*2) Real-World Dataset:* To assess the robustness of the proposed classwise detection model in real-world scenarios, we created a real-world dataset, which includes three parts: drone testing based on equivalent scaling models, recorded videos [38], [39], [40], [41] during the military exercises, and satellite images [42]. Examples and their labels are shown in Fig. 4(b). Evidently, the real-world samples exhibit substantial differences compared to the simulated ones, stemming from variations in textures, appearances, lighting conditions, and scene layouts, among other factors. Furthermore, the presence of a jitter in drone footage occasionally results in blurred imagery.

### B. Experiment Settings

In this article, all experiments were performed with the Windows 10 operating system, implemented on the MMDetection platform [43] and the Pytorch 1.7 framework. CUDA 11.2 was used to speed up the calculation, and the processor on the computer was Intel (R) Xeon (R) W-2255 CPU @3.70GHz, with 64 GB memory and an RTX A4000 graphics card. The batch size is set to 48. Each model is trained using the AdamW optimizer for 12 epochs with 0.05 weight decay, momentums $\beta_1 = 0.9$ and $\beta_2 = 0.95$. The initial learning rate was set to 5e-5. The original images are cropped into 112 × 112 pixels. Moreover, input images are augmented by random flips. The localization loss weight $\lambda$ is set to 1.0. Batch normalization is employed, whose

TABLE III
COMPARISON RESULTS WITH STATE-OF-THE-ART METHODS

| Types | Model | mAP$_{0.5:0.95}$ on ULMT-Key dataset | | mAP$_{0.5:0.95}$ on real-world dataset | Model size (MB) | Computational complexity (GFLOPS) | FPS |
|---|---|---|---|---|---|---|---|
| | | Standard | Generalized | | | | |
| Transformer | DINO [44] | 55.7% | 11.8% | 29.4% | 47.54 | 365 | 11.4 |
| | DDQ [45] | 45.8% | 10.0% | 27.6% | 48.31 | 199 | 9.7 |
| | RT-DETR [46] | 51.4% | 18.5% | 17.1% | 125.17 | 108.0 | 108 |
| One-stage | Yolov7-L [47] | 58.0% | 12.1% | 11.2% | 37.2 | 104 | 55 |
| | Yolov8-L [48] | 62.3% | 35.5% | 24.7% | 42.5 | 165 | 71 |
| RCNN | Faster | 62.9% | 27.7% | 27.0% | 41.13 | 118.23 | 22.4 |
| | Cascade [49] | 67.9% | 37.7% | 32.8% | 68.93 | 120.26 | 18.0 |
| | Grid [50] | 63.9% | 30.7% | 25.0% | 64.24 | 231.7 | 15.7 |
| | Libra [51] | 66.8% | 37.7% | 29.6% | 41.4 | 118.87 | 21.1 |
| | Double-Head [52] | 66.8% | 25.9% | 20.1% | 46.72 | 392.42 | 12.4 |
| | Dynamic [53] | 66.8% | 33.2% | 31.8% | 41.13 | 118.23 | 22.0 |
| | Sparse [54] | 63.8% | 35.8% | 13.6% | 105.95 | 94.19 | 18.4 |
| | Boosting [55] | 57.2% | 15.7% | 18.1% | 83.22 | 248.76 | 13.5 |
| | Decoupled [56] | 50.4% | 12.4% | 25.0% | 47.68 | 168.37 | 14.5 |
| | **Ours** | **70.7%** | **40.2%** | **39.5%** | 41.14 | 119.89 | 16.6 |

The bold values indicate the highest-performing results within their respective columns.

parameters will also be updated during training. To ensure fairness in subsequent comparative experiments, the widely adopted ResNet50 has opted as the backbone, which is pretrained on ImageNet. In addition, our backbone consists of four stages, with the first stage being frozen, thereby rendering the pretrained weights from this stage as non-trainable parameters.

### C. Evaluation Metrics

We reported the five indicators of mean average precision (mAP), production accuracy (PA), user accuracy (UA), overall accuracy (OA), and F1-score to evaluate the effectiveness of the model, which are calculated using the following equations:

$$\begin{aligned}
\mathrm{PA} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \\
\mathrm{UA} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \\
\mathrm{OA} &= \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \\
F_1 &= \frac{2\times P\times R}{P+R} \\
\mathrm{AP} &= \int_0^1 P(R)dR \\
\mathrm{mAP} &= \frac{1}{N}\sum_{i=1}^{N}\mathrm{AP}(i)
\end{aligned} \tag{14}$$

where TP represents the number of correctly detected positive instances, TN represents the number of correctly identified negative instances, FP represents the number of instances where the model mistakenly detects positive instances that do not exist, and FN represents the number of instances where the model misses the detection of positive instances. Additionally, *P* refers to precision and *R* refers to recall.

## V. RESULTS

To assess the effectiveness of the proposed CWKPD-Faster RCNN model on key part detection for UAV imagery, comparative experiments are conducted with state-of-the-art algorithms on our ULMT dataset and real-world dataset first, where all models are fine-tuned on the real-world dataset. Visualization studies are then conducted to more intuitively confirm the validity of our method.

### A. Comparison Results

The task of detecting key parts for multiclass targets based on the outputs of complete target detectors has not been specifically investigated to date. Furthermore, the existing part detectors made use of domain-specific expert knowledge, rendering them unsuitable for evaluation on our dataset. Therefore, we choose the state-of-the-art object detection methods in recent years as baselines, including RCNN-based models, one-stage models, and transformer-based models. The comparative results are presented in Tables III and IV, with "RCNN" being omitted when referring to RCNN-based models. It can be seen that our model always performs optimally in mAP, accuracy metrics, and F1-score, on both simulated and real-world datasets.

In comparison to all RCNN-based models, our model is almost the smallest, differing from the first place by only 0.02% in the number of model parameters. In terms of computational complexity, compared with Sparse RCNN, only an addition of 27.28% of model complexity can be traded for a 25.9% increase in accuracy on the real-world dataset, an average 5.65% increase on the simulated dataset as well as a 61.17% decrease on model size. As for the computational time, our model exhibits a slight decrease in computational speed by 25.89% in contrast with Faster RCNN. However, it demonstrates substantial improvements in mAP and overall accuracy, with an average

TABLE IV
ACCURACY RESULTS WHEN IOU = 0.75

| Types | Model | ULMT-Key dataset (standard) | | | | ULMT-Key dataset (Generalized) | | | | real-world dataset | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | PA | UA | OA | F1 | PA | UA | OA | F1 | PA | UA | OA | F1 |
| Transformer | DINO | 0.78 | 0.58 | 0.48 | 0.67 | 0.16 | 0.12 | 0.06 | 0.14 | 0.3 | 0.1 | 0.27 | 0.15 |
| | DDQ | 0.72 | 0.72 | 0.54 | 0.72 | 0.02 | 0.53 | 0.05 | 0.04 | 0.2 | 0.05 | 0.23 | 0.08 |
| | RT-DETR | 0.59 | 0.84 | 0.49 | 0.69 | 0.3 | 0.34 | 0.12 | 0.32 | 0.37 | 0.14 | 0.25 | 0.20 |
| One-stage | Yolov7-L | 0.63 | 0.78 | 0.53 | 0.70 | 0.27 | 0.28 | 0.06 | 0.27 | 0.18 | 0.14 | 0.13 | 0.16 |
| | Yolov8-L | 0.74 | 0.86 | 0.70 | 0.79 | 0.37 | 0.35 | 0.25 | 0.36 | 0.36 | 0.16 | 0.29 | 0.22 |
| RCNN | Faster | 0.89 | 0.41 | 0.55 | 0.56 | 0.23 | 0.09 | 0.09 | 0.13 | 0.32 | 0.11 | 0.08 | 0.16 |
| | Cascade | 0.9 | 0.44 | 0.75 | 0.59 | 0.38 | 0.2 | 0.23 | 0.26 | 0.32 | 0.15 | 0.17 | 0.20 |
| | Grid | 0.86 | 0.55 | 0.68 | 0.67 | 0.35 | 0.23 | 0.25 | 0.28 | 0.2 | 0.05 | 0.09 | 0.08 |
| | Libra | 0.89 | 0.44 | 0.58 | 0.59 | 0.4 | 0.18 | 0.22 | 0.25 | 0.32 | 0.09 | 0.11 | 0.14 |
| | Double-Head | 0.89 | 0.42 | 0.66 | 0.57 | 0.22 | 0.16 | 0.12 | 0.19 | 0.28 | 0.09 | 0.1 | 0.14 |
| | Dynamic | 0.85 | 0.42 | 0.72 | 0.56 | 0.36 | 0.17 | 0.25 | 0.23 | 0.26 | 0.1 | 0.15 | 0.15 |
| | Sparse | 0.85 | 0.2 | 0.53 | 0.32 | 0.56 | 0.01 | 0.25 | 0.02 | 0.07 | 0.01 | 0.1 | 0.02 |
| | Boosting | 0.8 | 0.35 | 0.51 | 0.49 | 0.16 | 0.03 | 0.03 | 0.05 | 0.16 | 0.1 | 0.04 | 0.12 |
| | Decoupled | 0.58 | 0.82 | 0.57 | 0.68 | 0.37 | 0.02 | 0.04 | 0.04 | 0.21 | 0.06 | 0.05 | 0.09 |
| | **Ours** | **0.9** | **0.89** | **0.79** | **0.89** | **0.41** | **0.36** | **0.27** | **0.38** | **0.37** | **0.31** | **0.33** | **0.34** |

The bold values indicate the highest-performing results within their respective columns.

enhancement of 10.15% and 1.21 times on the simulated dataset, and a remarkable improvement of 12.5% and 3.13 times on the real-world scene dataset.

In comparison to YOLOv8, which shows superior overall performance among all transformer-based models and one-stage models, our model experiences a decrease of 76.6% in computing speed. However, it compensates for this by delivering significant savings of 27.3% in terms of computing resources and 3.2% in storage resources. Importantly, our model exhibits notable enhancements in average precision by 6.55% and overall accuracy by 10.4% on the simulated datasets, while demonstrating remarkable improvements in average precision by 14.8% and overall accuracy by 13.8% on the real-world dataset. Overall, considering the constraints imposed by the frame rate of cameras mounted on UAVs and the demanding nature of storage, computing resources, and accuracy in real-world applications, the slight additional computational cost incurred by our enhanced model is both valuable and significant.

### B. *Visualization Results*

Furthermore, Fig. 5 presents four examples illustrating the detection results of the five most competitive models on both the simulated dataset and real-world datasets. The first two columns correspond to the flight test sets and equivalent scaling test sets in the real-world dataset, while the last two columns represent the generalized test set in the ULMT-Key dataset. In Fig. 5, the green and purple bounding boxes correspond to the ground truth and predictions of the tank, respectively. Similarly, the orange and red boxes indicate the ground truth and predicted locations of the RV. It is evident that our model consistently outperforms the others, which is not only reflected in precise positioning but also higher confidence scores. In conclusion, our proposed network demonstrates remarkable advantages in remote sensing vulnerable part detection tasks.

## VI. DISCUSSIONS

In this section, a comprehensive set of ablation experiments is conducted to validate the innovativeness and superiority of the proposed modules. Subsequently, the impact of the chosen backbones on the performance of our model is further assessed. Additionally, we analyze and discuss the instances where our model fails.

### A. *Ablation Study*

The ablation study is conducted to verify the effectiveness of each improvement strategy. The experimental results are shown in Table V. It can be found that introducing each module separately yields performance improvements while incorporating all modules collectively leads to the most significant enhancement. A particularly illustrative comparison is among the improvement strategies 1–4 on the standard test set, where the STMO-SPP module plays a more significant role in the advancement of detection accuracy. This improvement stems from the dual characteristics of the proposed STMO-SPP module: precise transformation to mitigate the negative impact of localization in full object detectors, and the ability to handle unrestricted input image sizes where the intrinsic features of the target are preserved. For the generalized dataset, the CWFR module plays a stronger role under the same comparisons, which demonstrates that the incorporation of semantic relationship pairs between targets and their key parts in the model can significantly enhance both the detection accuracy and generalization capability of key part detectors.

Furthermore, in the case of strategy 4, where only our designed weighted classification loss function is introduced, the model also exhibits a considerable improvement in localization accuracy. Particularly in scenarios where high localization accuracy is demanded, such as when the IoU is set to 0.95 for the standard test set and 0.85 for the generalized test set, this module

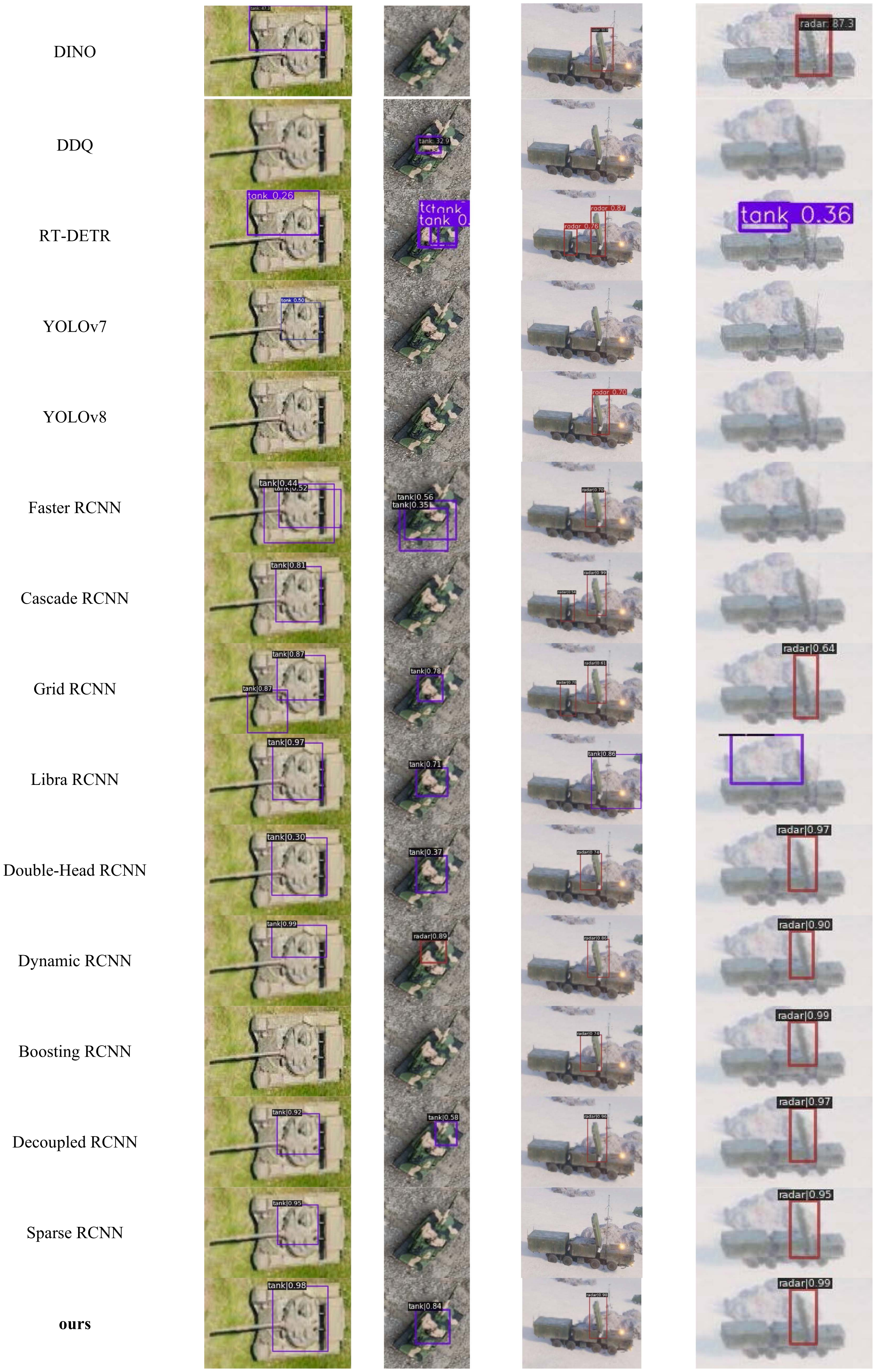


Fig. 5. Comparison of visualization results.

TABLE V
ABLATION STUDY

| No. | Improvement strategy | Standard test set | | | | | | Generalized test set | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | AP$_{.95}$ | AP$_{.9}$ | AP$_{.85}$ | AP$_{.8}$ | AP$_{.75}$ | A$_{.5:.7}$ | AP$_{0.9}$ | AP$_{0.85}$ | AP$_{0.8}$ | AP$_{0.75}$ | A$_{.5:.7}$ |
| 0 | Baseline | 2.4% | 9.5% | 32.2% | 59.1% | 76.7% | 89.8% | 0.5% | 1.3% | 6.4% | 11.1% | 46.1% |
| 1 | + STMO-SPP | 7.3% | 23.4% | 45.8% | 68.2% | 86.9% | 90.3% | 1% | 7.3% | 8.7% | 20.2% | 55.4% |
| 2 | + CWFR | 7.1% | 21.6% | 44.5% | 67.8% | 82.3% | 90.5% | 0.8% | 2.8% | 12.6% | 19.9% | 57.7% |
| 3 | $+L_{los}$ | 5% | 22.5% | 42.9% | 66.3% | 82.2% | 90.3% | 1.2% | 5.4% | 9.3% | 14.8% | 50.1% |
| 4 | $+L_{wcls}$ | 4.7% | 14.7% | 38.6% | 56.9% | 81.1% | 90.3% | 0.8% | 4.7% | 8% | 13.8% | 52.4% |
| 5 | $+L_{total}$ | 7.5% | 23.5% | 44.8% | 70.2% | 84% | 90.4% | 2.3% | 5.8% | 9.5% | 14.9% | 54.6% |
| 6 | +all | 9.2% | 27.6% | 54.8% | 75% | 87.4% | 90.6% | 4.8% | 6.5% | 14.2% | 23% | 62.6% |

TABLE VI
ADVANTAGE VERIFICATION OF THE PROPOSED CWFR MODULE

| Model | Standard test set | | | | | | Generalized test set | | | | | Model size/MB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AP$_{.95}$ | AP$_{.9}$ | AP$_{.85}$ | AP$_{.8}$ | AP$_{.75}$ | A$_{.5:.7}$ | AP$_{.9}$ | AP$_{.85}$ | AP$_{.8}$ | AP$_{.75}$ | A$_{.5:.7}$ | |
| Baseline | 1.6% | 12.7% | 37.3% | 68% | 81.1% | 90.2% | 0.3% | 2.4% | 7.5% | 16.7% | 41.9% | 41.13 |
| + CWFR | 4.4% | 24.3% | 49.6% | 72.5% | 88.4% | 90.6% | 0.4% | 2.6% | 8.9% | 17.9% | 49.8% | 41.14 |
| + CWFR-Spatial | 4.2% | 23.3% | 49.2% | 70% | 86.9% | 90.5% | 0.3% | 3.6% | 8.3% | 18.3% | 49.1% | 41.24 |
| + CWFR-Triple | 4.7% | 20.4% | 47.4% | 70.2% | 87.6% | 90.6% | 0.4% | 4% | 8.7% | 22.5% | 49.5% | 63.85 |

astonishingly enhances the mAP by 0.95 times and 2.61 times, respectively. This can be attributed to the enhanced correlation between the classification and location tasks, which empowers the model to achieve more precise localization.

### B. Advantage Verification of CWFR Module

*1) Quantitative Evaluation:* Our proposed classwise feature recalibration module encodes the prior knowledge as the feature descriptor in the channel dimension, instead of the spatial or triple dimension, the advantages of which will be evaluated. Four networks are compared: the Faster RCNN, the Faster RCNN model with the classwise feature recalibration module in the spatial dimension, and the Faster RCNN model with the classwise feature recalibration module in the triple dimension of spatial and channel. It should be remarked that the input size of the image is random in spatial, resulting in difficulties for the improvement in the spatial and triple modules. To show the performance obviously, we just crop the input image at a fixed size for the four models in this subsection. The detection accuracy results under different IoU thresholds are shown in Table VI.

Compared to Faster RCNN, the introductions of the CWFR block in the channel, spatial, and triple dimensions results in an additional model size increase of 0.04%, 0.26%, and 55.24%, respectively. Concurrently, it leads to substantial enhancements in detection accuracy, such as an improvement of 0.91 times, 0.83 times, and 0.6 times on the standard test sets when the IoU threshold is 0.85, and an improvement of 0.18 times, 0.11 times, and 0.16 times on the generalized test sets when the IoU threshold is set to 0.8. It is concluded that the slight additional storage burden on the channel dimensions has a significant effect on the improvement of detection accuracy.

In summary, the proposed CWFR module exhibits a simple and efficient approach for modeling semantic relationships for all classes of targets, which contributes significantly to the improvement of both detection accuracy and robustness with little increase in storage resources.

*2) Qualitative Evaluation:* Fig. 6(a) shows the original feature map corresponding to four distinct scenes, and Fig. 6(b) shows those adjusted by the CWFR module. It is apparent that the feature maps that learned prior knowledge can pay direct attention to the vulnerable areas, rather than the background and other parts of the target.

### C. Advantage Verification of STMO-SPP Module

*1) Quantitative Evaluation:* To demonstrate the superiority of the proposed spatial transformation module optimized by SPP, we compare the performance of three networks with different structures under the test sets: Faster RCNN with random input size, namely the baseline; Faster RCNN optimized by STN, whose input only could be fixed size; the proposed Faster RCNN optimized by STMO-SPP with random input size. The experimental results are shown in Table VII. Our proposed network optimized by STMO-SPP outperforms all other models on each metric on both standard test sets and generalized test sets. When the localization accuracy requirements gradually increase, the improvement driven by the STMO-SPP significantly increases, especially when the IoU threshold is greater than 0.75.

The maximum increase on the standard test set is two times when IoU $= 0.95$, and that on the generalized test set is 4.6

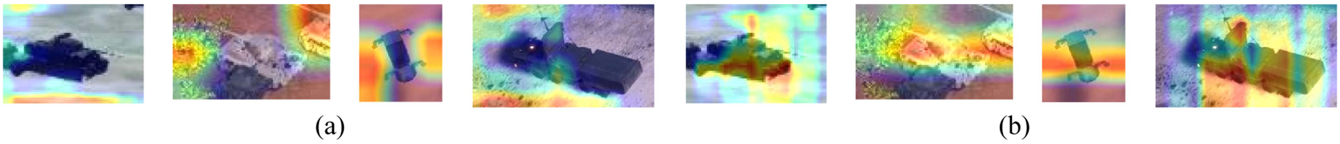
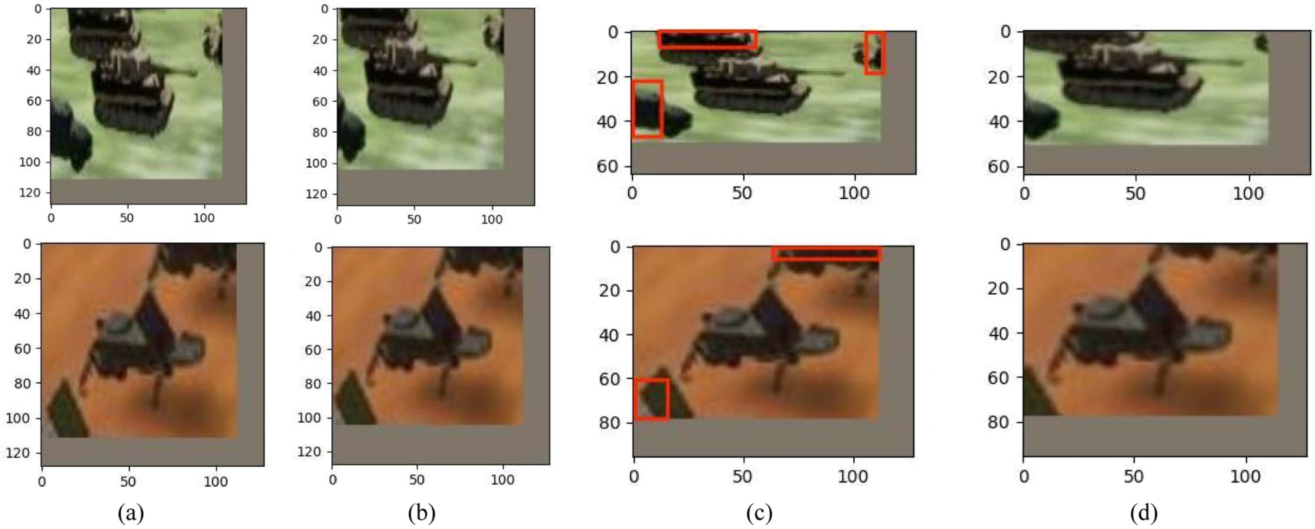

(a) (b)

Fig. 6. Visualization of CWFR module. (a) With CWFR module. (b) Without CWFR module.

TABLE VII
ADVANTAGE VERIFICATION OF THE PROPOSED STMO-SPP MODULE

| Model | Standard test set | | | | | | Generalized test set | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\text{AP}_{.95}$ | $\text{AP}_{.9}$ | $\text{AP}_{.85}$ | $\text{AP}_{.8}$ | $\text{AP}_{.75}$ | $\text{A}_{.5:.7}$ | $\text{AP}_{.9}$ | $\text{AP}_{.85}$ | $\text{AP}_{.8}$ | $\text{AP}_{.75}$ | $\text{A}_{.5:.7}$ |
| Baseline | 2.4% | 9.5% | 32.2% | 59.1% | 76.7% | 89.8% | 0.5% | 1.3% | 6.4% | 11.1% | 46.1% |
| +STN | 2.1% | 11.7% | 34.8% | 62.4% | 77.8% | 90.2% | 0.1% | 3.8% | 7.7% | 16.6% | 50.6% |
| +STMO-SPP | 7.3% | 23.4% | 45.8% | 68.2% | 86.9% | 90.4% | 1% | 7.3% | 8.7% | 20.2% | 55.4% |



(a) (b) (c) (d)

Fig. 7. Visualization results of STN and STMO-SPP. (a) Input of STN. (b) Output of STN. (c) Input of STN STMO-SPP. (d) Output of STMO-SPP.

(a) (b)

Fig. 8. Failure cases. (a) Miss of the full target detector. (b) Failure case owing to domain shift.

times when IoU = 0.85, at which point the baseline is almost completely invalidated. This demonstrates the proposed STMO-SPP is especially beneficial for accurate localization. Whereas, when only the STN module is introduced, the degree of improvement of the detection accuracy is substantially reduced, even lower than the baseline at IoU = 0.95 on the standard test set and at IoU = 0.9 on the generalized test set, whose reason can be deduced from the following visualization results.

*2) Qualitative Evaluation:* In Fig. 7, we visualize the input and output images of the networks optimized by the STN module and STMO-SPP module respectively to display explicitly the

TABLE VIII
COMPARISON RESULTS WITH VARIOUS BACKBONES

| Backbone | Models | ULMT-Key dataset (standard) | ULMT-Key dataset (Generalized) | real-world dataset |
|---|---|---|---|---|
| ResNeXt | Faster RCNN | 62.1 | 18.2 | 17.45 |
| | Cascade RCNN | 67.4 | 31.6 | 27.69 |
| | Dynamic RCNN | 61.4 | 21.4 | 28.27 |
| | Ours | 68.1 | 41.9 | 32.58 |
| Swin-Transformer | Faster RCNN | 61.2 | 28.9 | 11.4 |
| | Cascade RCNN | 67.2 | 29.1 | 13.60 |
| | Dynamic RCNN | 64.8 | 35.2 | 15.44 |
| | Ours | 69.0 | 37.5 | 20.17 |

learned affine transformation. Two images are taken as examples, where the full target is partially occluded or there are other interfering objects. The first column, i.e., Fig. 7(a), shows the input image of the STN module, whose size is always cropped to 112 × 112, and the second column is the corresponding output. Fig. 7(c) shows the input image of the STMO-SPP module without any preprocess and its output is demonstrated in Fig. 7(d). We find that the output images of the STN module are closer to the original images shown in Fig. 7(c), which means the STN module for the detection task tends to scale fixed-size images to appropriate proportions and sizes, yet rarely being cropped or rotated. Considering that the STN module cannot recover the input image perfectly, the target ratio always changes inevitably and reduces the detection accuracy to some extent. On the contrary, the proposed STMO-SPP module effectively addresses the limitations of the fixed input image size in STN and fully utilizes its advantages. Specifically, the STMO-SPP module preserves the original aspect ratio of the input image while eliminating interference objects through adaptive transformations, leading to a significant enhancement in the precision of key part detection. The interference areas autonomously cropped by the STMO-SPP module are highlighted in red in Fig. 7(c), aiming to enhance their prominence. It should be noted that very few interfering objects are completely removed, as this could also result in the accidental removal of essential components of the target.

### D. Evaluation of the Influence of the Chosen Backbone

Furthermore, we conducted additional validation of the performance of our model using different backbones. Alongside Faster R-CNN, two RCNN-based models are also included as comparative models: dynamic R-CNN and cascade R-CNN, which exhibited favorable overall performance according to Tables III and IV. ResNeXt [57] and Swin-Transformer [58] are employed as alternative backbone architectures of ResNet50. As shown in Table VIII, regardless of the selected backbone architecture, our model consistently achieves superior average accuracy, further demonstrating its competitive advantage.

### E. Failure Cases

Since the proposed method is based entirely on the results predicted by the full target detector, the designed key part detector cannot compensate for profoundly adverse mistakes made by the full target detector, such as misclassification or totally missing the regression box. In addition, our model shows reduced detection accuracy on visually distinct images compared to the training images owing to domain shift. Fig. 8 illustrates two failure cases.

## VII. Conclusion

This article builds a key part detector for UAV remote sensing images, aiming to locate the key part accurately based on the full target detection results, in which the spatial transformer module optimized by SPP and the classwise feature recalibration module are designed. The STMO-SPP block can adaptively crop out the interfering objects to encourage the model to focus more on the target, which is suitable for random sizes of input images. The CWFR module introduces the prior class information from the results of the full target detector, motivating the model to learn the dependencies between the prior knowledge and the location of the key part. Meanwhile, we proposed a classification loss function related to GIoU to alleviate the problem that the traditional cross-entropy loss function is irrelevant to the location task. The experimental results show that our proposed model exhibits better localization results and robustness both on the simulated dataset and real-world dataset.

In the future, we will enlarge our dataset by supplementing more classes of targets and backgrounds, as well as exploring various tricks and other localization loss functions to optimize our model, such as increasing the IOU threshold for positive objects, distance-IoU loss functions, and transfer our model onto transformer-based models or one-stage models. Moreover, we will be committed to building an integration framework combining the full target detector and key part detector closely and leveraging domain adaptive models in object detection in military scenarios to enhance the practicality of our model.

## References

[1] J. Lv et al., “Recognition of deformation military targets in the complex scenes via MiniSAR submeter images with FASAR-Net,” *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5209219, doi: 10.1109/TGRS.2023.3280946.

[2] X. Hu et al., “SPNet: Spectral patching end-to-end classification network for UAV-borne hyperspectral imagery with high spatial and spectral resolutions,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5503417, doi: 10.1109/TGRS.2021.3049292.

[3] Z. Qin et al., “Task selection and scheduling in UAV-enabled MEC for reconnaissance with time-varying priorities,” *IEEE Internet Things J*, vol. 8, no. 24, pp. 17290–17307, Dec. 2021, doi: 10.1109/JIOT.2021.3078746.

[4] L. Lomazzi et al., “Vulnerability assessment to projectiles: Approach definition and application to helicopter platforms,” *Def. Technol*, vol. 18, no. 9, pp. 1523–1537, Sep. 2022, doi: 10.1016/j.dt.2021.09.001.

[5] P. Qin et al., “Multilayer feature extraction network for military ship detection from high-resolution optical remote sensing images,” *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 11058–11069, 2021, doi: 10.1109/JSTARS.2021.3123080.

[6] L. Kong et al., “YOLO-G: A lightweight network model for improving the performance of military targets detection,” *IEEE Access*, vol. 10, pp. 55546–55564, 2022, doi: 10.1109/ACCESS.2022.3177628.

[7] K. Wang et al., “Reconcile prediction consistency for balanced object detection,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3611–3620, doi: 10.1109/ICCV48922.2021.00361.

[8] S. Ren et al., “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[9] S. Wang et al., “Automatic laser profile recognition and fast tracking for structured light measurement using deep learning and template matching,” *Measurement*, vol. 169, Feb. 2021, Art. no. 108362, doi: 10.1016/j.measurement.2020.108362.

[10] M. Liu et al., “Exploit visual dependency relations for semantic segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9721–9730, doi: 10.1109/CVPR46437.2021.00960.

[11] B. He et al., “Part-regularized near-duplicate vehicle re-identification,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3992–4000, doi: 10.1109/CVPR.2019.00412.

[12] A. Nadeem et al., “Automatic human posture estimation for sport activity recognition with robust body parts detection and entropy Markov model,” *Multimedia Tools Appl.*, vol. 80, no. 14, pp. 21465–21498, Jun. 2021, doi: 10.1007/s11042-021-10687-5.

[13] K. Hao et al., “An insulator defect detection model in aerial images based on multiscale feature pyramid network,” *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 3522412, doi: 10.1109/TIM.2022.3200861.

[14] C. Liu et al., “Real-time Human detection and gesture recognition for on-board UAV rescue,” *Sensors*, vol. 21, no. 6, Mar. 2021, Art. no. 2180, doi: 10.3390/s21062180.

[15] D. Choi et al., “UAV-driven structural crack detection and location determination using convolutional neural networks,” *Sensors*, vol. 21, no. 8, Apr. 2021, Art. no. 2650, doi: 10.3390/s21082650.

[16] X. He et al., “UAV-based road crack object-detection algorithm,” *Autom. Construction*, vol. 154, Oct. 2023, Art. no. 105014, doi: 10.1016/j.autcon.2023.105014.

[17] D. Biswas et al., “Unsupervised domain adaptation with debiased contrastive learning and support-set guided pseudolabeling for remote sensing images,” *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 3197–3210, 2024, doi: 10.1109/JSTARS.2024.3349541.

[18] Y. Zhu et al., “RFA-Net: Reconstructed feature alignment network for domain adaptation object detection in remote sensing imagery,” *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 5689–5703, 2022, doi: 10.1109/JSTARS.2022.3190699.

[19] D. Biswas and J. Tesic, “Small object difficulty (SOD) modeling for objects detection in satellite images,” in *Proc. IEEE 14th Int. Conf. Comput. Intell. Commun. Netw.*, 2022, pp. 125–130, doi: 10.1109/CICN56167.2022.10008383.

[20] D. Biswas, M. M. M. Rahman, Z. Zong, and J. Tesic, “Improving the energy efficiency of real-time DNN object detection via compression, transfer learning, and scale prediction,” in *Proc. IEEE Int. Conf. Netw., Archit. Storage*, 2022, pp. 1–8, doi: 10.1109/NAS55553.2022.9925528.

[21] L. Von Rueden et al., “Informed machine learning - A taxonomy and survey of integrating prior knowledge into learning systems,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 614–633, Jan. 2021, doi: 10.1109/TKDE.2021.3079836.

[22] J. Chen et al., “Graph convolutional network-based method for fault diagnosis using a hybrid of measurement and prior knowledge,” *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9157–9169, Sep. 2022, doi: 10.1109/TCYB.2021.3059002.

[23] W. Zhao et al., “Deep learning for COVID-19 detection based on CT images,” *Sci. Rep.*, vol. 11, no. 1, Jul. 2021, Art. no. 14353, doi: 10.1038/s41598-021-93832-2.

[24] W. Xiong et al., “Fine-grained image-to-image transformation towards visual recognition,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5839–5848, doi: 10.1109/CVPR42600.2020.00588.

[25] Z. Ding et al., “Guided variational autoencoder for disentanglement learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7917–7926, doi: 10.1109/CVPR42600.2020.00794.

[26] M. Jaderberg, K. Simonyan, and A. Zisserman, “Spatial transformer networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, vol. 28, pp. 2017–2025.

[27] Z. Zhong et al., “Spectral–spatial transformer network for hyperspectral image classification: A factorized architecture search framework,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5514715, doi: 10.1109/TGRS.2021.3115699.

[28] L. Li, M. Pagnucco, and Y. Song, “Graph-based spatial transformer with memory replay for multi-future pedestrian trajectory prediction,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2221–2231, doi: 10.1109/CVPR52688.2022.00227.

[29] J. Yan et al., “IoU-adaptive deformable R-CNN: Make full use of IoU for multi-class object detection in remote sensing imagery,” *Remote Sens.*, vol. 11, no. 3, Feb. 2019, Art. no. 286, doi: 10.3390/rs11030286.

[30] J. Nie et al., “Efficient selective context network for accurate object detection,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 9, pp. 3456–3468, Sep. 2021, doi: 10.1109/TCSVT.2020.3038649.

[31] S. Wu et al., “IoU-balanced loss functions for single-stage object detection,” *Pattern Recognit. Lett.*, vol. 156, pp. 96–103, Apr. 2022, doi: 10.1016/j.patrec.2022.01.021.

[32] S. Wu et al., “IoU-aware single-stage object detector for accurate localization,” *Image Vis. Comput.*, vol. 97, May 2020, Art. no. 103911, doi: 10.1016/j.imavis.2020.103911.

[33] Y. Li et al., “Learning power Gaussian modeling loss for dense rotated object detection in remote sensing images,” *Chin. J. Aeronaut.*, vol. 36, no. 10, pp. 353–365, Oct. 2023, doi: 10.1016/j.cja.2023.04.022.

[34] K. He et al., “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015, doi: 10.1109/TPAMI.2015.2389824.

[35] Z. Lin et al., “Point cloud change detection with stereo V-SLAM: Dataset, metrics and baseline,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 12443–12450, Oct. 2022, doi: 10.1109/LRA.2022.3219018.

[36] M. Fonder and M. Van Droogenbroeck, “Mid-air: A multi-modal dataset for extremely low altitude drone flights,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 553–562, doi: 10.1109/CVPRW.2019.00081.

[37] M. S. Mahmud et al., “A survey of data partitioning and sampling methods to support Big Data analysis,” *Big Data Mining Anal.*, vol. 3, no. 2, pp. 85–101, Jun. 2020, doi: 10.26599/BDMA.2019.9020015.

[38] The Sun, Dramatic moment Russian drone strikes Ukrainian tanks in Zaporizhzhia [Video]. YouTube, Jun. 12, 2023. [Online]. Available: https://www.youtube.com/watch?v=g3xEnAKzomg

[39] News.com.au, Drone footage captures strikes on Russian tank in Mariupol, Ukraine [Video]. YouTube, Mar. 17, 2022. [Online]. Available: https://www.youtube.com/watch?v=gXoyWH5FMgU

[40] TEW22, Ukrainian forces drone eliminate Russian troops tanks & IFVs in Vuhledar [Video]. YouTube, Feb. 28, 2023. [Online]. Available: https://www.youtube.com/watch?v=QbE0a3naSWw

[41] The Sun, Russian BMPT ‘Terminator’ tank is hit and destroyed by Ukrainian drone [Video]. YouTube, Aug. 12, 2023. [Online]. Available: https://www.youtube.com/watch?v=a1v0dkh7dtc

[42] Roboflow, Inc., “Pure Tank dataset,” Aerial detection, distributed by Roboflow, Jan. 2023. [Online]. Available: https://universe.roboflow.com/aerial-detection/pure-tank

[43] K. Chen et al., “MMDetection: Open MMLab detection toolbox and benchmark,” Jun. 17, 2019. [Online]. Available: http://arxiv.org/abs/1906.07155, Accessed: Sep. 25, 2023.

[44] H. Zhang et al., “DINO: DETR with improved denoising anchor boxes for end-to-end object detection.” Jul. 11, 2022. [Online]. Available: http://arxiv.org/abs/2203.03605, Accessed: Mar. 8, 2024.

[45] S. Zhang et al., “Dense distinct query for end-to-end object detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 7329–7338, doi: 10.1109/CVPR52729.2023.00708.

[46] E. Guemas et al., “Automatic patient-level recognition of four plasmodium species on thin blood smear by a real-time detection transformer (RT-DETR) object detection algorithm: A proof-of-concept and evaluation,” *Microbiol. Spectr.*, vol. 12, no. 2, pp. e01440–e01423, Feb. 2024, doi: 10.1128/spectrum.01440-23.

[47] C.-Y. Wang et al., “YOLOv7: Trainable bag-of-freebies sets new State-of-the-art for real-time object detectors,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 7464–7475, doi: 10.1109/CVPR52729.2023.00721.

[48] F. M. Talaat et al., “An improved fire detection approach based on YOLO-v8 for smart cities,” *Neural Comput. Appl.*, vol. 35, no. 28, pp. 20939–20954, Oct. 2023, doi: 10.1007/s00521-023-08809-1.

[49] Z. Cai and N. Vasconcelos, “Cascade R-CNN: Delving into high quality object detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6154–6162, doi: 10.1109/CVPR.2018.00644.

[50] X. Lu et al., “Grid R-CNN,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7355–7364.

[51] J. Pang et al., “Libra R-CNN: Towards balanced learning for object detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 821–830, doi: 10.1109/CVPR.2019.00091.

[52] Y. Wu et al., “Rethinking classification and localization for object detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10183–10192, doi: 10.1109/CVPR42600.2020.01020.

[53] H. Zhang et al., “Dynamic RCNN: Towards high quality object detection via dynamic training,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 260–275.

[54] P. Sun et al., “Sparse R-CNN: End-to-end object detection with learnable proposals,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14449–14458, doi: 10.1109/CVPR46437.2021.01422.

[55] P. Song et al., “Boosting R-CNN: Reweighting R-CNN samples by RPN’s error for underwater object detection,” *Neurocomputing*, vol. 530, pp. 150–164, Apr. 2023, doi: 10.1016/j.neucom.2023.01.088.

[56] D. Wang et al., “Decoupled R-CNN: Sensitivity-specific detector for higher accurate localization,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 9, pp. 6324–6336, Sep. 2022, doi: 10.1109/TCSVT.2022.3167114.

[57] T. Zhou et al., “ResNeXt and Res2Net structures for speaker verification,” in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2021, pp. 301–307, doi: 10.1109/SLT48900.2021.9383531.

[58] Z. Liu et al., “Swin Transformer: Hierarchical vision transformer using shifted Windows,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9992–10002, doi: 10.1109/ICCV48922.2021.00986.

**Hanyu Wang** received the B.S. degree in weapon systems and utilization engineering in 2019 from the School of Mechatronical Engineering, Beijing Institute of Technology, Beijing, China, where she is currently working toward the Ph.D. degree in armament science and technology.

Her research interests include object detection of images captured by UAVs, image processing, and visual navigation.

**Qiang Shen** received the B.S., M.S., and Ph.D. degrees in measurement technology and instruments from Beijing Institute of Technology, Beijing, China, in 1999, 2002 and 2005, respectively.

Since 2005, he has been with the School of Mechatronical Engineering, Beijing Institute of Technology, Beijing, China, where he has been a Professor since 2021. His research interests include intelligent unmanned system design, navigation, measurement and control technology, and projectile correction and precision guidance technology.

**Juan Li** received the B.S. degree in statistics and the Ph.D. degree in control science and engineering from Beijing Institute of Technology, Beijing, China, in 2013 and 2019, respectively.

She is currently an Assistant Professor with the School of Mechatronical Engineering, Beijing Institute of Technology. Her research interests include multiobjective evolutionary optimization, combinatorial and uncertain optimization, and swarm intelligence.

**Zihao Chen** received the B.S. degree in mechanical and electronic engineering in 2021 from Beijing Institute of Technology, Beijing, China, where he is currently working toward the M.S. degree in armament science and technology with the School of Mechatronical Engineering.

His research interests are UAV swarm mission planning in complex environments.

**Yiran Guo** received the B.S. degree in mechanical and electronic engineering in 2022 from Beijing Institute of Technology, Beijing, China, where he is currently working toward the M.S. degree in mechanical and electronic engineering with the School of Mechatronical Engineering.

His research interests include computer science, artificial intelligence, and image processing.

**Shouyi Zhang** received the B.S. degree in electronic information engineering from the China University of Mining and Technology, Xuzhou, China, in 2022. He is currently working toward the M.S. degree in armament science and technology with the School of Mechatronical Engineering, Beijing Institute of Technology, Beijing, China.

His research interests include deep learning and visual navigation.