# SuperYOLO: Super Resolution Assisted Object Detection in Multimodal Remote Sensing Imagery

Jiaqing Zhang, Jie Lei, *Member, IEEE*, Weiying Xie, *Member, IEEE*, Zhenman Fang, *Member, IEEE*, Yunsong Li, *Member, IEEE*, and Qian Du, *Fellow, IEEE*

***Abstract*—Accurately detecting multiscale small objects and accomplishing real-time detection using remote sensing imagery (RSI) remain challenging, especially for time-sensitive tasks such as military reconnaissance and emergency rescue. To obtain precise locations and classifications for those small objects, one of the most applicable solutions is to fuse the complementary information in multimodal images to enhance the detection capability. Most of the existing solutions primarily design a complex deep neural network to learn strong feature representations for objects separated from the background, which often results in a heavy computation burden.**

**In this paper, we propose an accurate yet fast small object detection method for RSI, named SuperYOLO, which fuses multimodal data and performs high resolution (HR) object detection on multiscale objects by utilizing the assisted super resolution (SR) learning and considering both the detection accuracy and computation cost. First, we construct a compact baseline by removing the Focus module to keep the HR features and significantly overcomes the missing error of small objects. Second, we utilize pixel-level multimodal fusion (MF) to extract information from various data to facilitate more suitable and effective features for small objects in RSI. Furthermore, we design a simple and flexible SR branch to learn HR feature representations that can discriminate small objects from vast backgrounds with low-resolution (LR) input, thus further improving the detection accuracy. Moreover, to avoid introducing additional computation, the SR branch is discarded in the inference stage and the computation of the network model is reduced due to the LR input. Experimental results show that, on the widely used VEDAI RS dataset, SuperYOLO achieves an accuracy of 73.61% (in terms of mAP$_{50}$), which is more than 10% higher than the SOTA large models such as YOLOv5l, YOLOv5x and RS designed YOLOrs. Meanwhile, the GFOLPs and parameter size of SuperYOLO are about 18.1x and 4.2x less than YOLOv5x. Our proposed model shows a favorable accuracy-speed trade-off compared to the state-of-art models. The code will be open sourced at https://github.com/icey-zhang/SuperYOLO.**

***Index Terms*—Object detection, multimodal remote sensing image, super resolution, feature fusion.**

## I. Introduction

OBJECT detection plays an important role in various fields involving computer-aided diagnosis or autonomous piloting. Over the past decades, numerous excellent deep neural network (DNN) based objection detection frameworks [1], [2], [3], [4], [5] have been proposed, updated, and optimized in computer vision. The remarkable accuracy enhancement of DNN-based object detection frameworks owes to the application of large-scale natural datasets with accurate annotations [6], [7], [8].

Compared with natural scenarios, there are several vital challenges for accurate object detection in remote sensing imagery (RSI). First, the number of labeled samples is relatively short, which limits the training of DNNs to achieve high detection accuracy. Second, the size of objects in RSI is much smaller, accounting for merely tens of pixels in relation to the complicated and broad backgrounds [9], [10]. Moreover, the scale of those objects is diverse with multiple categories [11]. As shown in Fig. 1(a), the object car is considerably small within a vast area. While shown in Fig. 1(b), the objects have large-scale variations, to which the scale of a car is smaller than that of a camping vehicle.

Currently, most object detection techniques are solely designed and applied for a single modality such as RGB and Infrared (IR) [12], [13]. Consequently, with respect to object detection, its capability to recognize objects on the Earth's surface remains insufficient due to the deficiency of complementary information between different modalities [14]. As imaging technology flourishes, RSIs collected from multimodality become available and provide an opportunity to improve the detection accuracy. For example, as shown in Fig. 1, the fusion of two different multimodalities (RGB and IR) can effectively enhance the detection accuracy in RSI.

In this study, our motivation is to *propose an on-board real-time object detection framework for multimodal RSIs to achieve high detection accuracy and high inference speed without introducing additional computation overhead.* Inspired by recent advances in real-time compact neural network models, we choose small-size YOLOv5s [15] structure as our detection baseline. It can reduce deployment costs and facilitate rapid deployment of the model. Considering the high resolution (HR) retention requirements for small objects, we remove the Focus module in the baseline YOLOv5s model, which not only benefits defining the location of small dense objects but also enhances the detection performance. Considering the complementary characteristics in different modalities, we propose a multimodal fusion (MF) scheme to improve the detection performance for RSI. We evaluate different fusion alternatives (pixel-level or feature-level) and choose pixel-level fusion for low computation cost.

This work was supported in part by the National Natural Science Foundation of China under Grant 62071360.

J. Zhang, J. Lei, W. Xie, Y. Li are with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China (e-mail: jqzhang_2@stu.xidian.edu.cn; jielei@mail.xidian.edu.cn; wyxie@xidian.edu.cn; ysli@mail.xidian.edu.cn).

Z. Fang is with School of Engineering Science, Simon Fraser University, Burnaby, BC, Canada (e-mail: zhenman@sfu.ca).

Q. Du is with the Department of Electronic and Computer Engineering, Mississippi State University, Starkville, MS 39759 USA (e-mail: du@ece.msstate.edu).

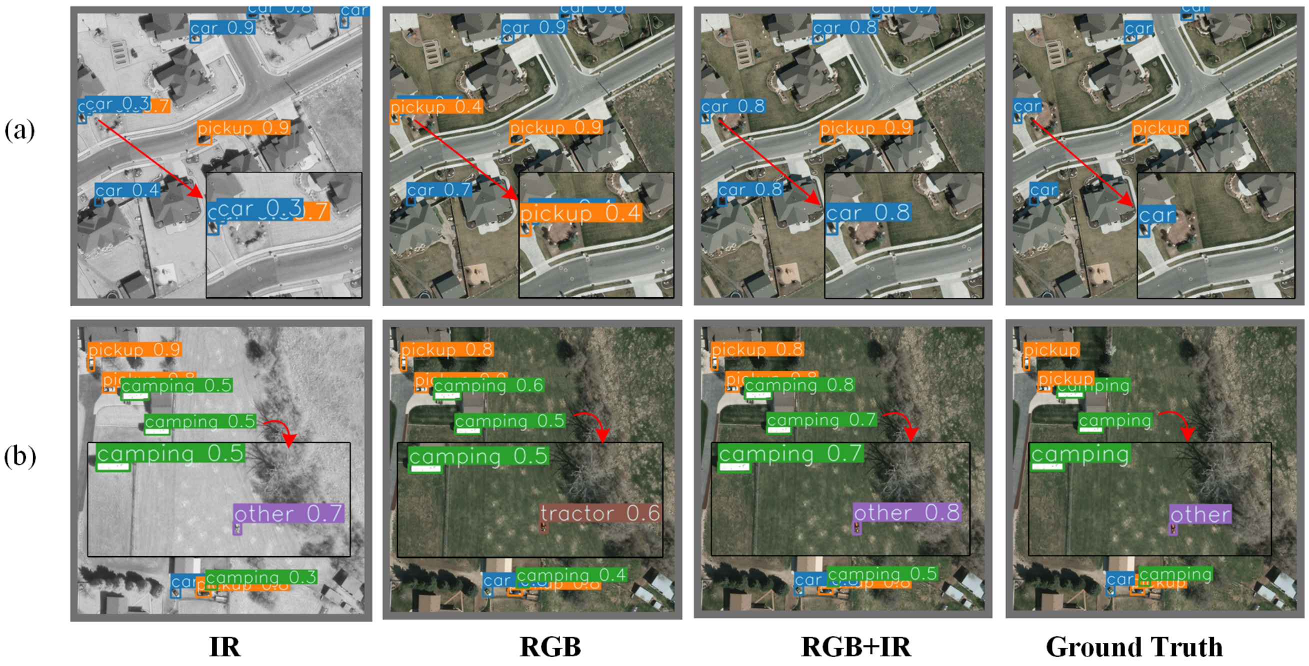

Fig. 1. Visual comparison of RGB image, IR image, and ground truth (GT). The IR image provides vital complementary information for resolving the challenges in RGB detection. The object car in (a) is considerably small within a vast area. In (b), the objects have large-scale variation, to which the scale of a car is smaller than that of a camping vehicle. The fusion of RGB and IR modalities effectively enhances the detection performance.

Lastly and most importantly, we develop a super resolution (SR) assurance module to guide the network to generate HR features that are capable of identifying small objects in vast backgrounds, thereby reducing false alarms induced by background-contaminated objects in RSI. Nevertheless, a naive SR solution can significantly increase the computation cost. Therefore, we set the auxiliary SR branch engaged in the training process and remove it in the inference stage, facilitating spatial information extraction in HR without increasing computation cost.

In summary, this paper makes the following contributions.

- We design a compact baseline detection network to achieve higher accuracy of small multiscale objects in RSIs and realize real-time detection.
- We explore different fusion alternatives and choose the computation-friendly pixel-level fusion method for multimodal information combinations to further enhance the detection accuracy. The proposed pixel-level efficiently decreases the computation cost compared with feature-level fusion.
- We further introduce an assisted SR branch into multimodal object detection for the first time. Our approach not only makes a breakthrough in limited detection performance but also paves a more flexible way to study outstanding HR feature representations that are capable of discriminating small objects from vast backgrounds with LR input.
- Considering the demand of high-quality results and low-computation cost, the SR module functioning as an auxiliary task is removed during the inference stage without introducing additional computation. Our proposal can greatly improve the detection performance while retaining similar FLOPs with those of the baseline framework. The SR branch is general and extensible and can be utilized in the existing fully convolutional network (FCN) framework.
- The proposed SuperYOLO markedly improves the performance of object detection, outperforming SOTA detectors in real-time multimodal object detection. On the widely used VEDAI RS dataset, SuperYOLO accomplishes **73.61%** mAP$_{50}$, exceeding the YOLOv5s framework by 16.68%, and exceeding YOLOv5l, YOLOv5x and YOLOrs by more than 10%. The GFOLPs and parameters, and are about 18.1x and 4.2x less than YOLOv5x. Our proposed model shows a favorable accuracy-speed trade-off compared to the state-of-art models.

The rest of this paper is organized as follows. Section II outlines the related work of object detection using multimodal data and SR technique. Section III briefly describes the baseline architecture of the widely used YOLOv5 model. Section IV presents our proposed SuperYOLO architecture. Section V conducts experiments and analyzes the results. Section VI concludes this paper and discusses the future work.

## II. Related Work

### A. Object Detection with Multimodal Data

Recently, multimodal data has been widely leveraged in numerous practical application scenarios, including visual question answering [16], auto-pilot vehicles [17], saliency detection [18], and remote sensing classification [19]. It is found that combining the internal information of multimodal data can efficiently transfer complementary features to avoid certain information of a single modality from being omitted.

In the field of RSI processing, there exist various modalities (e.g., Red-Green-Blue (RGB), Synthetic Aperture Radar

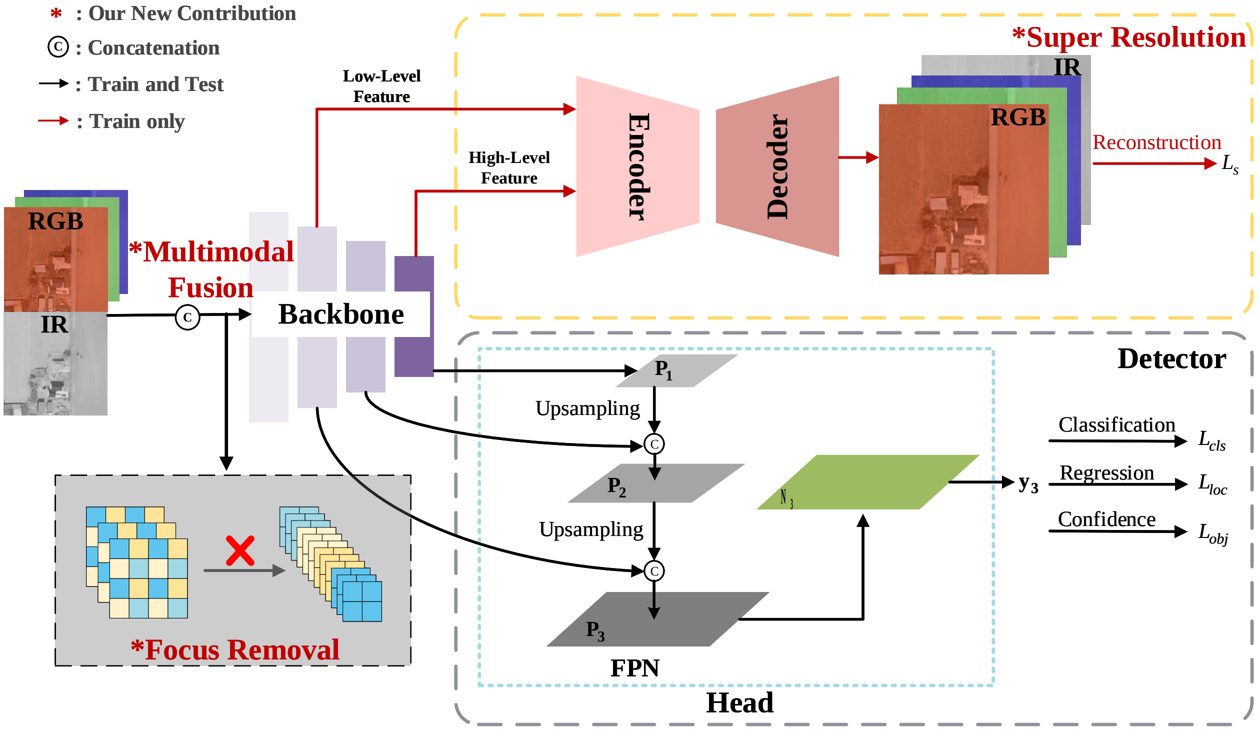

Fig. 2. The overview of the proposed SuperYOLO framework. Our new contributions include 1) removal of the Focus module to reserve high resolution, 2) multimodal fusion, and 3) assisted SR branch. The architecture will be optimized in terms of Mean Square Error (MSE) loss for the SR branch and task-specific loss for object detection. During the training stage, the SR branch guides the related learning of the spatial dimension to enhance the high resolution information preservation for the backbone. During the test stage, the SR branch is removed to accelerate the inference speed equal to the baseline.

(SAR), Light Detection and Ranging (LiDAR), Infrared (IR), panchromatic (PAN) and multispectral (MS) images) from diverse sensors, which can be fused complementary characteristics to enhance the performance of various tasks [20], [21], [22]. For example, the additional IR modality [23] captures longer thermal wavelengths to improve the detection under difficult weather conditions. Manish *et al.* [23] proposed a real-time framework for object detection in multimodal remote sensing imaging, in which the extended version conducted mid-level fusion and merged data from multiple modalities. Despite that multi-sensor fusion can enhance the detection performance as shown in Fig 1, hardly can its low-accuracy detection performance and to-be-improved computing speed meet the requirements of real-time detection tasks.

The fusion methods are primarily grouped into three strategies, i.e., pixel-level fusion, feature-level fusion, and decision-level fusion methods [24]. The decision-level fusion methods fuse the detection results during the last stage, which may consume enormous computation resources due to repeated calculation for different multimodal branches. In the field of remote sensing, feature-level fusion methods are mainly adopted with multi branches. The multimodal images will be input into the parallel branches to extract respective independent features of different modalities, and then these features will be combined by some operations, such as attention module or simple concatenation. The parallel branches bring a repeated computation as the modalities increase, which is not friendly in the real-time tasks in remote sensing.

In contrast, the adoption of pixel-level fusion methods can reduce unnecessary computation. In this paper, our proposed SuperYOLO fuses the modalities at pixel-level to significantly reduce the computation cost.

### B. Super Resolution in Object Detection

Conducted in a pre-processing step, SR has proven to be effective and efficient in various object detection tasks [25], [26]. Shermeyer *et al.* [27] quantified its effect on the detection performance of satellite imaging by multiple resolutions of RSI. Based on generative adversarial networks (GANs), Courtrai *et al.* [28] utilized SR to generate HR images, which were fed into the detector to improve its detection performance. Rabbi *et al.* [29] leveraged a Laplacian operator to extract edges from the input image to enhance the capability of reconstructing HR images, thus improving its performance in object localization and classification. Hong *et al.* [30] introduced a cycle-consistent GAN structure as an SR network and modified faster R-CNN architecture to detect vehicles from enhanced images that are produced by the SR network. In these works, the adoption of SR structure has effectively addressed the challenges regarding small objects. However, compared with single detection models, additional computation is introduced, which attributes to the enlarged scale of the input image by HR design.

Recently, Wang *et al.* [31] proposed an SR module that can maintain HR representations with LR input while reducing the model computation in segmentation tasks. Inspired by the [31], we design an SR assisted branch. In contrast to the aforementioned work in which the SR is realized in the start stage, the assisted SR module guides the learning of high-quality HR representations for the detector, which not only strengthens the response of small dense objects but also improves the performance of object detection in spatial space. Moreover, the SR module is removed in the inference stage to avoid extra computation.

## III. Baseline YOLOv5s Architecture

YOLOv5 model is known as an advanced structure employed to generate low-level texture features and high-level semantic features. It is one of the most widely used object detection frameworks. We follow the YOLOv5 [15] design and take it as our baseline framework. Unlike the previous generation of YOLO models, YOLOv5 [15] releases four models: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, where the basic structures are identical. The depths and widths of the models depend on the number of bottleneck layers and convolution kernels, respectively. The multi-model characteristics make YOLOv5 have higher flexibility and versatility in practical applications. To realize real-time object detection, the small-scale model size and fast reasoning speed are the fundamental reasons we choose YOLOv5s as the baseline.

As shown in Figure 2, the baseline YOLOv5 network consists of two main components: the Backbone and Head (including Neck). The backbone is designed to extract low-level texture and high-level semantic features. Next, these hint features are fed to Head to construct the enhanced feature pyramid network from top to bottom to transfer robust semantic features and from bottom to top to propagate a strong response of local texture and pattern features. This resolves the various scale issue of the objects by yielding an enhancement of detection with diverse scales.

### A. Backbone Module and Its Limitations

In Fig. 3, CSPNet [32] is utilized as the Backbone to extract the feature information, consisting of numerous sample Convolution-Batch-normalization-SiLu (CBS) components and Cross Stage Partial (CSP) modules. The CBS is composed of operations of convolution, batch normalization, and activation function SiLu [33]. The CSP duplicates the feature map of the previous layer into two branches and then halves the channel numbers through $1 \times 1$ convolution, by which the computation is therefore reduced. With respect to the two copies of the feature map, one is connected to the end of the stage, and the other is sent into ResNet blocks or CBS blocks as the input. Finally, the two copies of the feature map are concatenated to combine the features, which is followed by a CBS block. The SPP (Spatial Pyramid Pooling) module [34] is composed of parallel Maxpool layers with different kernel sizes and is utilized to extract multiscale deep features. The low-level texture and high-level semantic features are extracted by stacked CSP, CBS, and SPP structures.

**Limitation 1:** It is worth mentioning that the Focus module is introduced to decrease the number of computation. As shown in Fig. 2 (bottom left), inputs are partitioned into individual pixels and reconstructed at intervals and finally concatenated in the channel dimension. The inputs are resized to a smaller scale to reduce the computation cost and accelerate the network training and inference speed. However, this may sacrifice object detection accuracy to a certain extent, especially for small objects vulnerable to resolution.

**Limitation 2:** It is known that the backbone of YOLO employs deep convolutional neural networks to extract hierarchical features with a stride step of 2, through which the size of the extracted features is halved. Hence, the feature size retained for multiscale detection is far smaller than that of the original input image. For example, when the input image size is 608, the sizes of output features for the last detection layer are 76, 38, and 19, respectively. LR features may result in the missing of some small objects.

### B. Head Module

The Head (including the Neck module) module is devised to efficiently combine the multiscale and multilevel features generated by the Backbone. It integrates FPN [35] and PANet [36] (shown in Fig. 2) to generate a feature pyramid network. It can enhance the detection with diverse receptive fields and therefore recognize the same object with different scales and sizes. The design of this structure is based on the following rationale: neurons in higher layers strongly respond to the entire object while other neurons are more likely to be activated by local texture and patterns [37]. This implies that deeper-layer neurons carry increasingly less information about the contents of an entire object and increasingly more information related to the class of the object, whereas lower-layer neurons are more likely to be activated by local representations such as textures and patterns. On one hand, the FPN module mainly acts to transfer top-down robust semantic features to enhance detection, especially for small-size targets. On the other hand, the PANet module further enhances the localization capability of the entire feature hierarchy by propagating a solid response of low-level features, including bottom-up local textures and patterns, thus improving the detection effect on large objects.

As shown in Fig. 2, three scale features $\mathbf{y_1}$, $\mathbf{y_2}$ and $\mathbf{y_3}$ from the Head are used to complete the last detection work. The output feature map at each detection head is a 4D tensor denoted as $\mathbf{y_1}, \mathbf{y_2}, \mathbf{y_3} \in \mathbb{R}^{A \times F \times F \times V}$. Specifically, mode-1 corresponds to the serial number of the anchor of each layer, mode-2 the width-index of each cell, mode-3 the height-index of each cell, and mode-4 the bounding boxes. $A = 3$ denotes the 3 anchors, the $F$ denotes the feature map size, $V$ is the sum of 4 bounding box offsets, 1 objectness prediction, and the number of class predictions. For each predicted bounding box, our method returns an output feature vector in mode-4, which is written as:

$$v_k|_{k=1,2,3} = [t_x, t_y, t_w, t_h, o, p_1, p_2, p_{3,} ......, p_N] \,, \quad (1)$$

where $(t_x, t_y)$ are the output corresponding to the center coordinates of the bounding box in the top-left corner of the cell, $t_w$ and $t_h$ are the output corresponding to the width and the height of the predicted bounding box , respectively, $o$ is the objectness score in $[01]$ about the possibility of the bounding box containing an object , and $p_j (j = 1, 2, ......, N)$ is the class score indicating the categories of the predicted object between the range of $[0, 1]$ with $N$ being the number of categories..

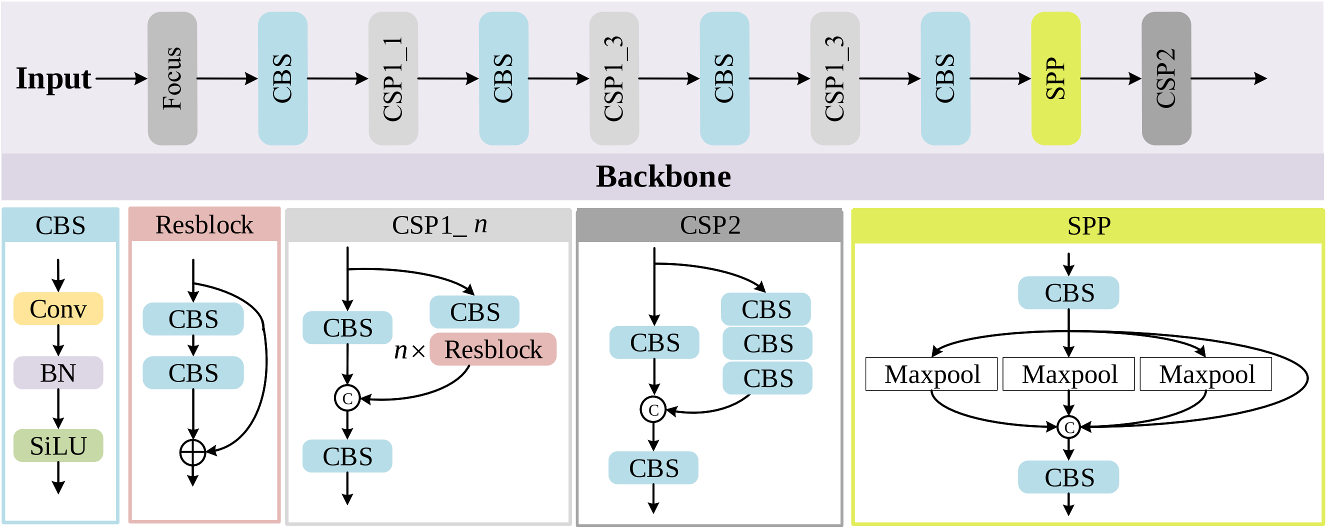

Fig. 3. The backbone structure of YOLOv5s. The low-level texture and high-level semantic features are extracted by stacked CSP, CBS, and SPP structures.

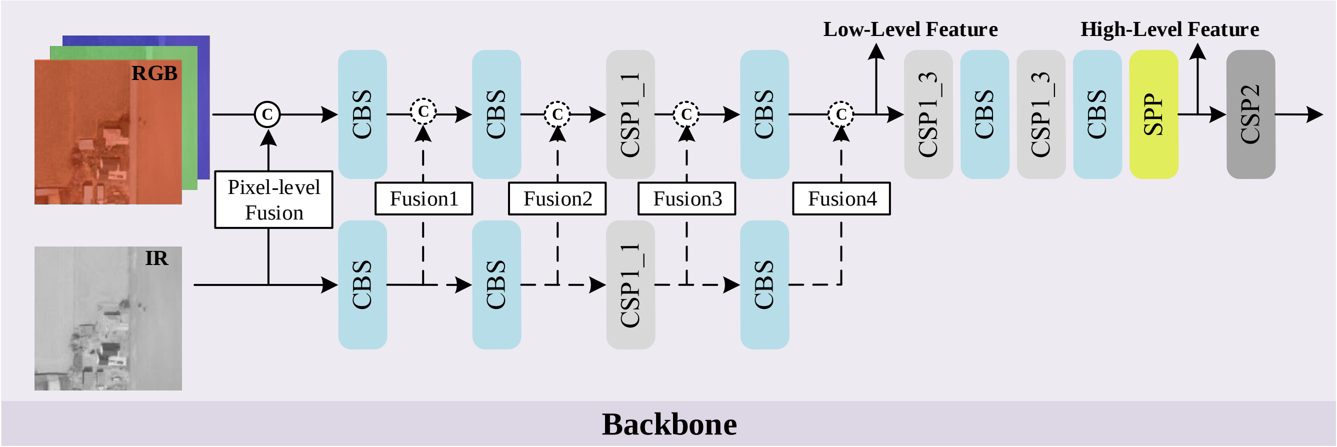

Fig. 4. The pixel-level fusion and feature-level fusion of different blocks. To be fair, the IR image is expanded to three bands in feature-level fusion. The Fusion1, Fusion2, Fusion3, and Fusion4 represent a feature-level fusion operation after the first, second, third, and fourth block, respectively.

## IV. SuperYOLO Architecture

As summarized in Fig. 2, we introduce three new contributions in our SuperYOLO network architecture. First, we remove the Focus module in the Backbone and replace it with a CBS module, to avoid the resolution degradation and thus accuracy degradation. Second, we explore different fusion methods and choose the computation-efficient pixel-level fusion to fuse RGB and IR modalities to refine dissimilar and complementary information. Finally, we add an assisted SR module in the training stage, which reconstructs the HR images to guide the related Backbone learning in spatial dimension and thus maintain HR information. In the inference stage, the SR branch is discarded to avoid introducing additional computation overhead.

### A. Focus Removal

As presented in Section III-A and Fig. 2 (bottom left), the Focus module in the YOLOv5 backbone partitions images at intervals on spatial domain and then reorganizes the new image to resize the input images. Specifically, this operation is to collect a value for each pixel in an image and then reconstruct it to obtain smaller complementary images. The size of the rebuilt images decreases with the increase of the number of channels. As a result, it causes resolution degradation and spatial information loss for small targets. Considering that the detection of small targets depends more heavily on higher resolution, the Focus module is abandoned and replaced by a CBS module with convolution operations (shown in Fig. 4) to prevent the resolution from being degraded. In addition, we deleted the PANet structure and two detectors, which are responsible for enhancing large-scale target detection because we mainly focus on small objects in remote sensing.

### B. Multimodal Fusion

The more information is utilized to distinguish objects, the better performance can be achieved in object detection. Multimodal fusion is an effective path for merging different information from various sensors. The decision-level, feature-level and pixel-level fusions are the three mainstream fusion methods that can be deployed at different depths of

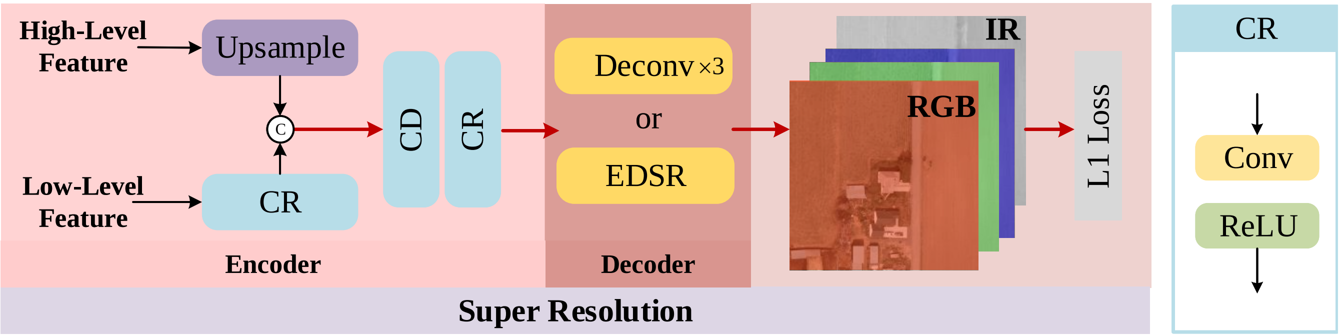

Fig. 5. The super resolution (SR) structure of SuperYOLO. The SR structure can be regarded as a simple Encode-Decoder model. The low-level and high-level features of the backbone are selected to fuse local textures patterns and semantic information, respectively.

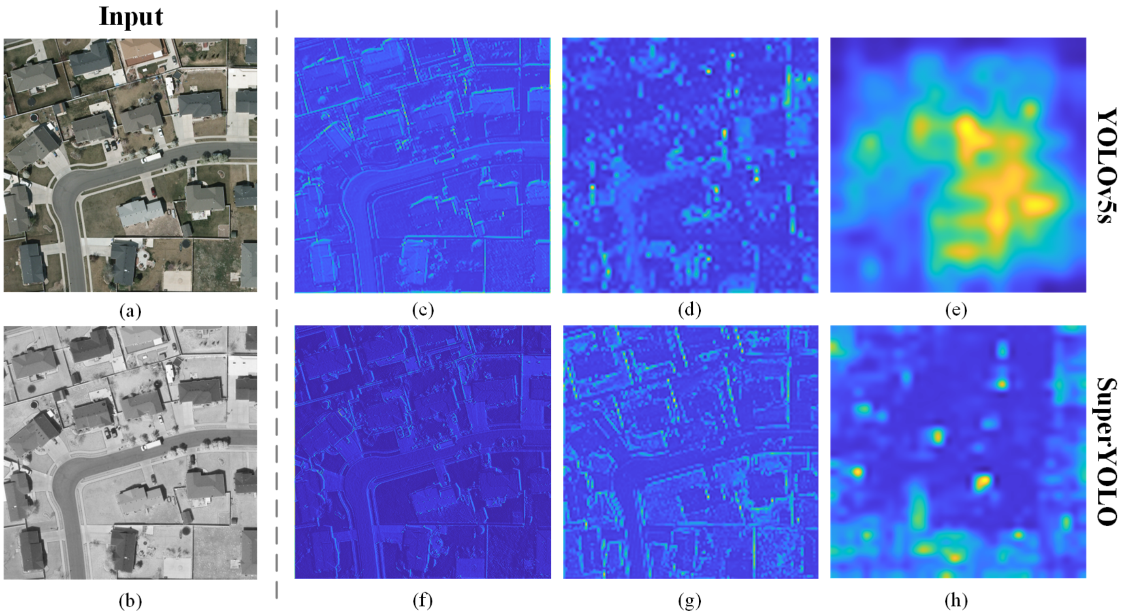

Fig. 6. Feature-level visualization of backbone for SuperYOLO and YOLOv5s with the same input: (a) RGB input, (b) IR input; (c), (d), and (e) are the features of YOLOv5s; (f), (g), and (h) are the features of SuperYOLO. The features are upsampled to the same scale as the input image for comparison. (c) and (f) are the features in the first layer. (d) and (g) are the low-level features. (e) and (h) are the high-level features in layers at the same depth.

the network. Since decision-level fusion requires enormous computation, it is not considered in SuperYOLO. Next, we describe how we perform feature-level fusion and pixel-level fusion in SuperYOLO.

The feature-level fusion of different blocks is demonstrated in Fig. 4. For a fair comparison, the IR image is expanded to three bands. The fusion1, fusion2, fusion3, and fusion4 represent the fusion operation performed in the first, second, third, and fourth blocks, respectively. The concatenation is regarded as the fusion operation.

For the pixel-level fusion, we first normalize an input RGB image and an input IR image into two intervals of $[0, 1]$, and then we concatenate them with relatively low computation compared with the other two fusion methods that conduct the fusion operation in later procedures to accelerate the inference. As will be presented in Section V-D, the pixel-level fusion achieves superior performance than feature-level fusion to merge different complementary information.

To be more specific, the fusion image is defined as:

$$\mathbf{X} = Concat(\mathbf{R}, \mathbf{G}, \mathbf{B}, \mathbf{I}), \tag{2}$$

where, fusion images are $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$, $C$ represents the channel number, $H$ and $W$ represent the image height and width, respectively, $\{\mathbf{R}, \mathbf{G}, \mathbf{B}\}$ and $\{\mathbf{I}\}$ represent the RGB image and IR image, respectively, and $Concat(\cdot)$ denotes the concatenation operation along the channel-axis. Then, $\mathbf{X}$ is subsampled to $1/n$ size of the original image to accomplish the SR module discussed in subsection IV-C and to accelerate the training process. The sampled image is denoted as $\mathbf{X}' \in \mathbb{R}^{C\times \frac{H}{n}\times \frac{W}{n}}$ and generated by:

$$\mathbf{X}' = D(\mathbf{X}), \tag{3}$$

where $D(\cdot)$ represents $n$ times downsampling operation using bilinear interpolation. The subsampled result is then fed to the Backbone to produce multi-level features, as shown in Fig. 4.

### C. Super Resolution

As mentioned in Section III-A, the feature size retained for multiscale detection in the backbone is far smaller than that of the original input image. Most of the existing methods conduct upsampling operations to recover the feature size.

TABLE I. The Influence of Removing the Focus Module in the Network on the First Fold of the Validation Set.

| Method | | Parameters ↓ | GFLOPs↓ | Mean Precision ↑ | Mean Recall ↑ | mAP$_{50}$ ↑ |
|---|---|---|---|---|---|---|
| YOLOv5s | Focus | 7.0739M | 5.3 | 70.8 | 51.1 | 62.2 |
| | noFocus | 7.0705M | 20.4 | **69.3** (-1.5) | **65.4** (+14.3) | **69.5** (+7.3) |
| YOLOv5m | Focus | 21.0677M | 16.1 | 60.8 | 62.1 | 64.5 |
| | noFocus | 21.0625M | 63.6 | **69.4** (+8.6) | **68.3** (+6.2) | **72.2** (+7.7) |
| YOLOv5l | Focus | 46.6406M | 36.7 | 72.6 | 55.6 | 63.7 |
| | noFocus | 46.6337M | 145.0 | **76.0** (+4.5) | **64.6** (+8.9) | **72.5** (+8.6) |
| YOLOv5x | Focus | 87.2487M | 69.7 | 73.6 | 60.4 | 64.0 |
| | noFocus | 87.2400M | 276.6 | **71.9** (-1.7) | **62.6** (+2.2) | **69.2** (+5.2) |

Unfortunately, this approach has produced limited success due to the information loss in texture and pattern, which explains that it is inappropriate to employ this operation to detect small targets in RSI that require HR preservation.

To address this issue, as shown in Fig. 2, we introduce an auxiliary SR branch. First, the introduced branch shall facilitate the extraction of HR information in the backbone and achieve satisfactory performance. Second, the branch should not add more computation to reduce the inference speed. It shall realize a trade-off between accuracy and computation time during the inference stage. Inspired by the study of Wang *et al.* [31] where the proposed super resolution succeeded in facilitating segmentation tasks without additional requirement, we introduce a simple and effective branch named SR into the framework. Our proposal can improve the detection accuracy without computation and memory overload, especially under circumstances of LR input.

Specifically, the SR structure can be regarded as a simple Encode-Decoder model. We select the backbone's low-level and high-level features to fuse local textures and patterns and semantic information, respectively. As depicted in Fig. 4, we select the result of the fourth and ninth modules as the low-level and high-level features, respectively. The Encoder integrates the low-level feature and high-level feature generated in the backbone. As illustrated in Fig. 5, in Encoder, the first CR module is conducted on low-level feature. For high-level feature, we use a Upsampling operation to match the spatial size of low-level feature and then we use concatenation operation and two CR modules to merge the low-level and high-level features. The CR module includes a convolution and ReLU. For the Decoder, the LR feature is upscaled to the HR space in which the SR module's output size is twice larger than that of the input image. As illustrated in Fig. 5, the Decoder is implemented using three deconvolutional layers. The SR guides the related learning of spatial dimension and transfers it to the main branch, thereby improving the performance of object detection. In addition, we introduce EDSR [38] as our Encoder structure to explore the SR performance and its influence on detection performance.

To present a more visually interpretable description, we visualize the features of backbones for YOLOv5s and SuperYOLO in Fig. 6. The features are upsampled to the same scale as the input image for comparison. By comparing the pairwise images of (b) and (f), (c) and (g), (d) and (h) in Fig. 6, it can be observed that SuperYOLO contains clearer object structures with higher resolution with the assistance of the SR. Eventually, we obtain a bumper harvest in high-quality HR representation with the SR branch and utilize the Head of YOLOv5 to detect small objects.

### D. Loss Function

The overall loss of our network consists of two components: detection loss $L_o$ and SR construction loss $L_s$, which can be expressed as

$$L_{total} = c_1 L_o + c_2 L_s, \tag{4}$$

where $c_1$ and $c_2$ are the coefficients for a balance of the two training tasks. The L1 loss (rather than L2 loss) [39] is used to calculate the SR construction loss $L_s$ between the input image $\mathbf{X}$ and SR result $\mathbf{S}$ , to which the expression is written as

$$L_s = \|\mathbf{S} - \mathbf{X}\|_1 . \tag{5}$$

The detection loss involves three components [15]: loss of judging whether there is an object $L_{obj}$, loss of object location $L_{loc}$, and loss of object classification $L_{cls}$, which are used to evaluate the loss of the prediction as

$$L_o = \lambda_{loc} \sum_{l=0}^{3} a_l L_{loc} + \lambda_{obj} \sum_{l=0}^{3} b_l L_{obj} + \lambda_{cls} \sum_{l=0}^{3} c_l L_{cls}. \tag{6}$$

Here, Equation 6, $l$ represents the layer of the output in head, $a_l$, $b_l$, and $c_l$ are the weights of different layers for the three loss functions, the weights $\lambda_{loc}$, $\lambda_{obj}$, and $\lambda_{cls}$ regulate error emphasis among box coordinates, box dimensions, objectness, no-objectness and classification.

## V. Experimental Results

### A. Dataset

The popular Vehicle Detection in Aerial Imagery (VEDAI) dataset [40] is used in the experiments, which contains cropped images obtained from the much larger Utah Automated Geographic Reference Center (AGRC) dataset. Each image collected from the same altitude in AGRC has approximately $16,000 \times 16,000$ pixels, with a resolution of about $12.5cm \times 12.5cm$ per pixel. RGB and IR are the two modalities for each image in the same scenes. The VEDAI dataset consists of 1246 smaller images that focus on diverse backgrounds involving grass, highway, mountains, and urban areas. All images are in resolution of $1024 \times 1024$ or $512 \times 512$. The task is to detect 11 classes of different vehicles such as car, pickup, camping, and truck. In this study, $512 \times 512$ and $1024 \times 1024$ images are used to validate the importance of the resolution in object detection assignment. The default image resolution used in the testing process is $512 \times 512$ unless otherwise specified.

TABLE II. Distribution of Available Class Instances in the VEDAI Dataset Across 10 Folds.

| Classes | Total Instances | Distribution Across 10 Folds |
|---|---|---|
| Car | 1349 | 9 folds of 135; 1 fold of 134 |
| Pickup | 941 | 9 folds of 94; 1 fold of 95 |
| Camping car | 390 | 10 folds of 39 |
| Truck | 300 | 10 folds of 30 |
| Other | 200 | 10 folds of 20 |
| Tractor | 190 | 10 folds of 19 |
| Boat | 170 | 10 folds of 17 |
| Van | 100 | 10 folds of 10 |

The classes with less than 50 instances are neglected in our study. Hence, we solely follow the 8 classes in the dataset shown in TABLE II, where the available instances per class used in the test are divided into 10 folds. Images without annotation are removed. The annotations for each object in the image contain the coordinates of the bounding box center, the orientation of the object concerning the positive $x$-axis, the four corners of the bounding box, the class ID, a binary flag identifying whether an object is occluded, and another binary flag identify whether an object is cropped. The annotations of the VEDAI dataset are converted to YOLOv5 format, and the We transfer the ID of the interested class are transferred to $0, 1, ..., 7$, i.e., $N = 8$. Then the center coordinates of the bounding box are normalized and absolute coordinate is transformed to relative coordinate. Similarly, the length and width of the bounding box are normalized to $[0, 1]$.

TABLE III. The Comparison Results of Model Size and Inference Ability in Different Baseline YOLO Frameworks.

| Method | Layers ↓ | Parameters ↓ | GFLOPs ↓ | mAP$_{50}$ ↑ |
|---|---|---|---|---|
| YOLOv3 | 270 | 61.5M | 52.8 | 62.6 |
| YOLOrs | 241 | 20.2M | 46.4 | 55.8 |
| YOLOv4 | 393 | 52.5M | 38.2 | **65.7** |
| YOLOv5s | **224** | **7.1M** | **5.32** | 62.2 |
| YOLOv5m | 308 | 21.1M | 16.1 | 64.5 |
| YOLOv5l | 397 | 46.6M | 36.7 | 63.9 |
| YOLOv5x | 476 | 87.3M | 69.7 | 64.0 |

### *B. Accuracy Metrics*

The accuracy assessment measures the agreements and differences between the detection result and the reference mask. The recall, precision, and mAP (mean Average Precision) are used as accuracy metrics to evaluate the performance of the methods to be compared with. The calculations of the precision and recall metrics are defined as

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN}. \tag{8}$$

where the true positive (TP) and true negative (TN) denote correct prediction, and the false positive (FP) and false negative (FN) denote incorrect outcome. The precision and recall are correlated with the commission and omission errors, respectively. The mAP is a comprehensive indicator obtained by averaging AP values, which uses an integral method to calculate the area enclosed by the Precision-Recall curve and coordinate axis of all categories. Hence, the mAP can be calculated by

$$mAP = \frac{AP}{N} = \frac{\int_0^1 p(r)dr}{N}, \tag{9}$$

where $p$ denotes Precision, $r$ denotes Recall, and $N$ is the number of categories.

In addition, PSNR and SSIM are used for image quality evaluation of SR branch. Generally, higher PSNR values and SSIM values represent the better quality of the generated image.

### *C. Implementation Details*

Our proposed framework is implemented in PyTorch and runs on a workstation with an NVIDIA 2080Ti GPU. Following [23], the VEDAI dataset is used to train our SuperYOLO. In order to realize the SR assisted branch, the input images of the network are downsampled from $1024 \times 1024$ size to $512 \times 512$ during the training process. In the test process, the image size is $512 \times 512$, which is consistent with the input of other algorithms compared. In addition, data is augmented with Hue Saturation Value (HSV), multi-scale, translation, left-right flip, and mosaic. The augmentation strategy is canceled in the test stage. The standard Stochastic Gradient Descent (SGD) [41] is used to train the network with the momentum of 0.937, weight decay of 0.0005 for the Nesterov accelerated gradients utilized, and the batch size of 2. The learning rate is set to 0.01 initially. The entire training process involves 300 epochs, which cost nearly 6 hours.

### *D. Ablation Study*

First of all, we verify the effectiveness of our proposed method by designing a series of ablation experiments which are conducted on the first fold of the validation set.

*1) **Validation of the Baseline Framework:*** In Table III, the model size and inference ability of different base frameworks are evaluated in terms of the number of layers, parameter size and GFLOPs. The detection performances of those models are measured by mAP$_{50}$, i.e., detection metric of mAP at IOU (Intersection over Union) = 0.5. Although YOLOv4 achieves the best detection performance, it has 169 more layers than YOLOv5s (393 vs. 224), its parameter size is 7.4 times larger than that of YOLOv5s (52.5M vs. 7.1M), and its GFLOPs is 7.2 times higher than that of YOLOv5s (38.2 vs. 5.3). With respect to YOLOv5s, although its mAP is slightly lower than those of YOLOv4 and YOLOv5m, its number of layers, parameter size and GFLOPs are much smaller than those of other models. Therefore, it is easier to deploy YOLOv5s on board to achieve real-time performance in practical applications. The above fact verifies the rationality of YOLOv5s as the baseline detection framework.

TABLE IV. The Influence of Different Resolutions for Input Image on Network Performance on the First Fold of the Validation Set.

| Method | Train-Val Size | Test Size | Parameters ↓ | GFLOPs ↓ | Mean Precision ↑ | Mean Recall ↑ | $mAP_{50}$ ↑ |
|---|---|---|---|---|---|---|---|
| YOLOv5s | 512-512 | 512 | 7.0739M | **5.3** | **70.8** | **51.1** | **62.2** |
| | | 1024 | 7.0739M | 21.3 | 32.7 | 12.4 | 10.6 |
| | 1024-1024 | 1024 | 7.0739M | 21.3 | **78.1** | **69.8** | **77.7** |
| | | 512 | 7.0739M | **5.3** | 45.4 | 57.2 | 48.2 |
| YOLOv5s (noFocus) | 512-512 | 512 | 7.0705M | **20.4** | **69.3** | **65.4** | **69.5** |
| | | 1024 | 7.0705M | 81.5 | 20.8 | 16.9 | 13.4 |
| | 1024-1024 | 1024 | 7.0705M | 81.5 | **83.7** | **71.3** | **79.3** |
| | | 512 | 7.0705M | **20.4** | 64.2 | 59.2 | 62.9 |
| YOLOv5s (noFocus) +SR | 512-512 | 512 | 7.0705M | **20.4** | **74.5** | **73.4** | **78.0** |

*2) **Impact of Removing Focus Module**:* As presented in Section IV-A, the Focus module reduces the resolution of input images, which imposes encumbrance on the detection performance of small objects in RSI. To investigate the influence of the Focus module, we conduct experiments on the four YOLOv5 network frameworks: YOLOv5s, YOLOV5m, YOLOv5l, and YOLOv5x. Note that the results here are collected after the pixel-level fusion of RGB and IR modalities. As listed out in Table I, the mean recall scores of those frameworks are improved by removing the Focus module. In particular, the mean recall score of YOLOv5s is improved by 14.3% (51.1%→69.3%). This is because by removing the Focus module, not only can the resolution degradation be avoided, but also the spatial interval information be retained for small objects in RSI, thereby reducing the missing errors of object detection. At the same time, the mean precision scores are improved for YOLOv5m and YOLOv5l, and slightly dropped for YOLOv5s and YOLOv5x. Overall, after removing the Focus module, we observe the noticeable improving of the detection performance of YOLOv5s (62.2%→69.5% in $mAP_{50}$), YOLOv5m (64.5%→72.2%), YOLOV5l (63.7%→72.5%), YOLOv5x (64.0%→69.2%). Generally, removing the Focus module brings more than 5% improvement in the detection performance ($mAP_{50}$) of the whole frameworks.

TABLE V. The Comparison Result of Pixel-level and Feature-level Fusions in YOLOv5s for Multimodal Dataset on the First Fold of the Validation Set.

| Method | | Parameters ↓ | GFLOPs ↓ | $mAP_{50}$ ↑ |
|---|---|---|---|---|
| Pixel-level Fusion | | **7.0705M** | **20.37** | **69.5** |
| Feature-level Fusion | Fusion1 | 7.0887M | 21.76 | 66.0 |
| | Fusion2 | 7.0744M | 22.04 | 68.5 |
| | Fusion3 | 7.1442M | 24.22 | 64.8 |
| | Fusion4 | 7.0870M | 24.50 | 63.8 |

Meanwhile, we notice that the above removal increases the inference computation cost (GFLOPs) in YOLOv5s (5.3→20.4), YOLOv5m (16.1→63.6), YOLOV5l (36.7→145), YOLOv5x (69.7→276.6). However, the GFLOPs of YOLOv5s-noFocus (20.4) is smaller than those of YOLOv3 (52.8), YOLOv4 (38.2), and YOLOrs (46.4), as shown in Table III. The parameters of these models are slightly reduced after removing the Focus module. In summary, in order to retain the resolution to better detect more smaller objects, priority shall be given to the detection accuracy, for which the convolution operation is adopted to replace the Focus module.

TABLE VI. The Effective Validation of the Super Resolution branch for the Different Baseline.

| Method | Layers | Parameters | GFLOPs | $mAP_{50}$ |
|---|---|---|---|---|
| **YOLOv3** | 270 | 61.5M | 52.8 | 62.6 |
| **YOLOv3+SR** | 270 | 61.5M | 52.8 | 71.8 |
| **YOLOv4** | 393 | 52.5M | 38.2 | 65.7 |
| **YOLOv4+SR** | 393 | 52.5M | 38.2 | 69.0 |
| **YOLOv5s** | 224 | 7.1M | 5.3 | 62.2 |
| **YOLOv5s+SR** | 224 | 7.1M | 5.3 | 64.4 |

*3) **Comparison of Different Fusion Methods**:* To evaluate the influence of the devised fusion methods, we conduct experiments with pixel-level and feature-level fusion methods on YOLOv5-noFocus, as presented in Section IV-B. The final result is listed out in TABLE V. The parameter size, GFLOPs, $mAP_{50}$ of pixel-level fusion are 7.0705M, 20.37 and 69.5%, respectively, which are the best among all the compared methods. The above results suggest that the pixel-level fusion can accurately detect objects while reducing the computation. Therefore, we choose the pixel-level fusion as our final fusion strategy, which exhibits relatively competitive performance for the VEDAI multimodal dataset with objects that are difficult to distinguish.

*4) **Impact of High Resolution**:* We compare different training and test modes to explore more possibilities in terms of the input image resolution in TABLE IV. First, we compare for cases where the image resolutions of the training set and test set are the same. By comparing the result of YOLOv5s, the detection metric $mAP_{50}$ is improved from 62.2% to 77.7%, causing 15.6% increase when the image resolution (size) is doubled from 512 to 1024. Similarly, YOLOv5s-noFocus (1024) outperforms YOLOv5s-noFocus (512) by 9.8% $mAP_{50}$ score (79.3% vs. 69.5%). The mean recall and mean precision increase simultaneously, suggesting that ensuring resolution

TABLE VII. The Ablation Experiment Results about the Influence of SR Branch on Detection Performance.

| **YOLOv5s (noFocus) +SR** | **Small-scale Detector** | **Decoder (EDSR)** | **L1 Loss** | **Parameters ↓** | **GFLOPs ↓** | **mAP$_{50}$ ↑** | **PSNR ↑** | **SSIM ↑** |
|---|---|---|---|---|---|---|---|---|
| ✓ | ✓ | | | 4.8259M | 16.68G | 79.0 | 23.811 | 0.602 |
| ✓ | ✓ | ✓ | | 4.8259M | 16.68G | 79.9 | 23.902 | 0.604 |
| ✓ | ✓ | ✓ | ✓ | 4.8259M | 16.68G | 80.9 | 26.203 | 0.659 |

TABLE VIII. Class-wise Average Precision AP, Mean Average Precision mAP$_{50}$, Parameters and GFLPs for Proposed SuperYOLO, YOLOv3, YOLOv4, YOLOv5s-x Including Unimodal And Multimodal Configurations on VEDAI Dataset.

| **Method** | | **Car** | **Pickup** | **Camping** | **Truck** | **Other** | **Tractor** | **Boat** | **Van** | **mAP$_{50}$** | **Params. ↓** | **GFLOPs ↓** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| YOLOv3 | IR | 80.21 | 67.03 | 65.55 | 47.78 | 25.86 | 40.11 | 32.67 | 53.33 | 51.54 | **61.5351M** | **49.55** |
| | RGB | 83.06 | 71.54 | **69.14** | 59.30 | **48.93** | **67.34** | 33.48 | 55.67 | 61.06 | **61.5351M** | **49.55** |
| | Multi | **84.57** | **72.68** | 67.13 | **61.96** | 43.04 | 65.24 | **37.10** | **58.29** | **61.26** | 61.5354M | 49.68 |
| YOLOv4 | IR | 80.45 | 67.88 | 68.84 | 53.66 | 30.02 | 44.23 | 25.40 | 51.41 | 52.75 | **52.5082M** | **38.16** |
| | RGB | 83.73 | **73.43** | 71.17 | 59.09 | **51.66** | 65.86 | **34.28** | **60.32** | 62.43 | **52.5082M** | **38.16** |
| | Multi | **85.46** | 72.84 | **72.38** | **62.82** | 48.94 | **68.99** | **34.28** | 54.66 | **62.55** | 52.5085M | 38.23 |
| YOLOv5s | IR | 77.31 | 65.27 | 66.47 | 51.56 | 25.87 | 42.36 | 21.88 | 48.88 | 49.94 | **7.0728M** | **5.24** |
| | RGB | 80.07 | 68.01 | 66.12 | 51.52 | 45.76 | **64.38** | 21.62 | 40.93 | 54.82 | **7.0728M** | **5.24** |
| | Multi | **80.81** | **68.48** | **69.06** | **54.71** | **46.76** | 64.29 | **24.25** | **45.96** | **56.79** | 7.0739M | 5.32 |
| YOLOv5m | IR | 79.23 | 67.32 | 65.43 | 51.75 | 26.66 | 44.28 | 26.64 | 56.14 | 52.19 | **21.0659M** | **16.13** |
| | RGB | 81.14 | 70.26 | 65.53 | 53.98 | **46.78** | **66.69** | **36.24** | 49.87 | 58.80 | **21.0659M** | **16.13** |
| | Multi | **82.53** | **72.32** | **68.41** | **59.25** | 46.20 | 66.23 | 33.51 | **57.11** | **60.69** | 21.0677M | 16.24 |
| YOLOv5l | IR | 80.14 | 68.57 | 65.37 | 53.45 | 30.33 | 45.59 | 27.24 | **61.87** | 54.06 | **46.6383M** | **36.55** |
| | RGB | 81.36 | 71.70 | 68.25 | 57.45 | 45.77 | **70.68** | 35.89 | 55.42 | 60.81 | **46.6383M** | **36.55** |
| | Multi | **82.83** | **72.32** | **69.92** | **63.94** | **48.48** | 63.07 | **40.12** | 56.46 | **62.16** | 46.6046M | 36.70 |
| YOLOv5x | IR | 79.01 | 66.72 | 65.93 | 58.49 | 31.39 | 41.38 | 31.58 | 58.98 | 54.18 | **87.2458M** | **69.52** |
| | RGB | 81.66 | 72.23 | 68.29 | 59.07 | 48.47 | 66.01 | **39.15** | **61.85** | 62.09 | **87.2458M** | **69.52** |
| | Multi | **84.33** | **72.95** | **70.09** | **61.15** | **49.94** | **67.35** | 38.71 | 56.65 | **62.65** | 87.2487M | 69.71 |
| SuperYOLO | IR | 87.90 | 81.39 | 76.90 | 61.56 | 39.39 | 60.56 | 46.08 | **71.00** | 65.60 | **4.8256M** | **16.61** |
| | RGB | 90.30 | 82.66 | 76.69 | 68.55 | 53.86 | 79.48 | 58.08 | 70.30 | 72.49 | **4.8256M** | **16.61** |
| | Multi | **90.86** | **84.35** | **78.11** | **68.11** | **53.26** | **82.33** | **60.95** | 70.94 | **73.61** | 4.8259M | 16.68 |

reduces the commission and omission errors in object detection. Based on the above analysis, we argue that the characteristics of HR significantly influence the final performance of object detection.

However, it is noteworthy that maintaining an HR input image of the network introduces a certain amount of calculation. The GFLOPs with a resolution of 1024 × 1024 is higher than that with 512 × 512 in both YOLOv5s (21.3 vs. 5.3) and YOLOv5s-noFocus (81.5 vs. 20.4). Note that the GFLOPs is calculated in the test processing.

As shown in Table IV, the use of different sizes of image during the training process (train size) and the test process (test size) results in the score reduction of mAP, i.e., (10.6% vs. 62.1%), (48.2% vs. 77.7%), (13.4% vs. 69.5%) and (62.9% vs. 79.3%). This may attribute to the inconsistent scale of objects in the test process and in the training process, where the size of the predicted bounding box is not suitable for the objects of test images anymore.

Finally, the mAP$_{50}$ of YOLOv5s-noFocus+SR is close to the YOLOv5-noFocus (1024) HR one (78.0% vs. 79.3%), and the GFOLPs is equal to that of YOLOv5-noFocus (512) LR one (20.4 vs. 20.4). Our proposed network decreased the resolution of input images in the test process to reduce computation and maintain accuracy by remaining the identical resolution of the training and testing data, thereby highlighting the advantage of the proposed SR branch.

*5) **Impact of Super Resolution Branch**:* Table VI shows the favorable accuracy-speed tradeoff of SR branch. At the different baseline, the influence of the SR branch on object detection is positive. Compared with bare baseline, baseline added super resolution shows favorable performance: YOLOv3+SR performs mAP$_{50}$ 9.2% better than YOLOv3, YOLOv4+SR is mAP$_{50}$ 3.3% better than YOLOv4, YOLOv5s+SR performs mAP$_{50}$ 2% better than YOLOv5s. Notably, Super resolution can be removed in the inference stage. Hence no extra parameters and computation costs are introduced, which is impressive considering that the SR branch does not require a lot of manpower to refine the design of the detection network. The SR branch is general and extensible and can be utilized in the existing fully convolutional network (FCN) framework. In addition, some ablation experiences about the SR branch are completed in Table VII. When we utilize EDSR network as Decoder and L1 loss as SR loss function in the SR branch, which is powerful in the SR task, not only the performance of SR is improved but also the performance of the detection network enhanced meantime, because the SR branch helps

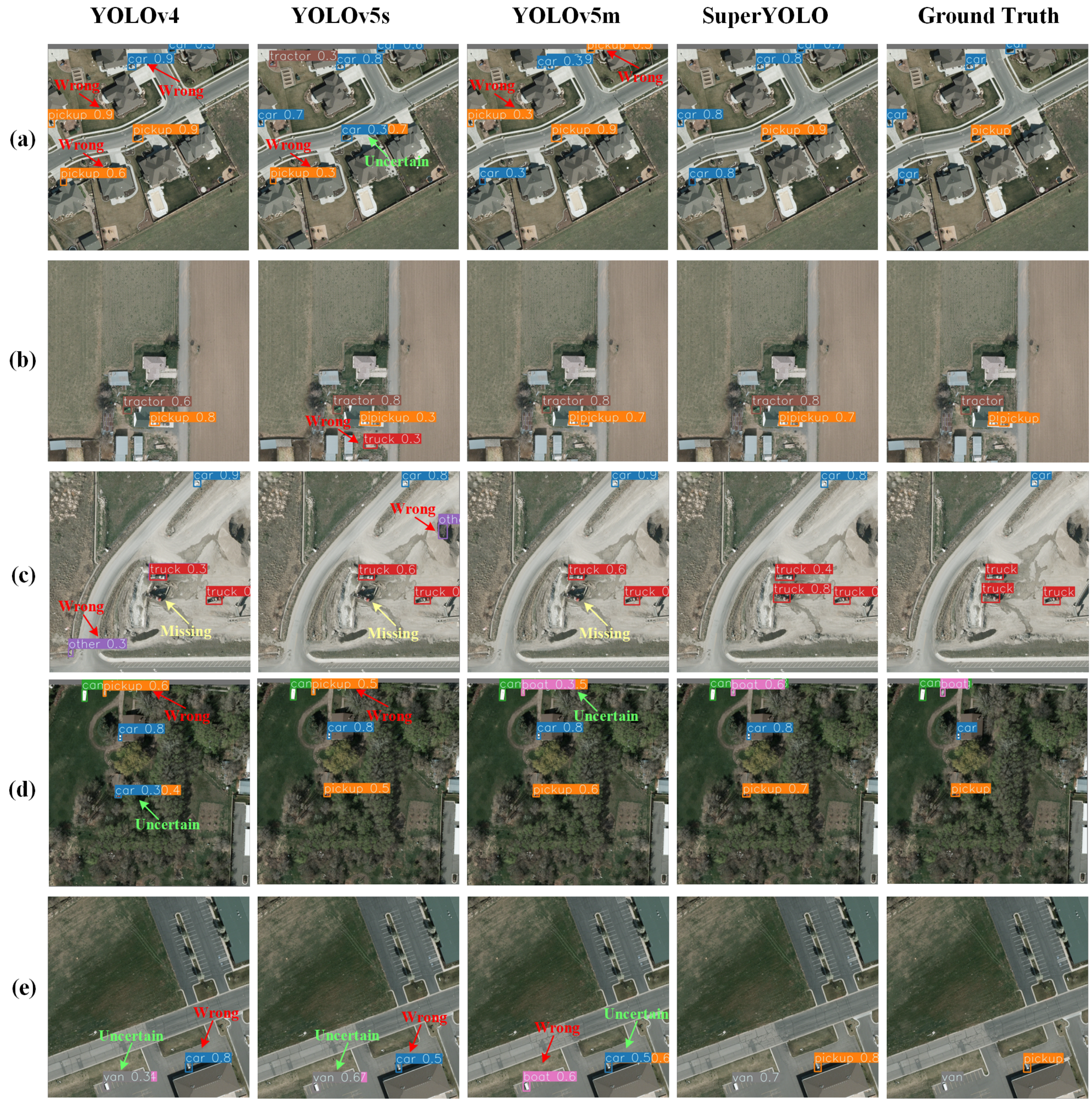

Fig. 7. Visual results of object detection using different methods involving YOLOv4, YOLOv5s, YOLOv5m and the proposed SuperYOLO.

the detection network to extract more effective and superior features in the backbone, accelerating the convergence of the detection network and thus improving the performance of the detection network. The performance of super resolution and object detection is complementary and cooperative.

## *E. Comparisons with Previous Methods*

The visual detection results of the compared YOLO methods and SuperYOLO are shown in Fig. 7, for a diverse set of scenes. It can be observed that SuperYOLO can accurately detect those objects that are not detected, or predicted into a wrong category or with uncertainty, in YOLOv4, YOLOv5s, and YOLOv5m. The objects in RSIs are challenging to detect on small scales. In particular, **Pickup** and **Car** or **Van** and **Boat** are easily confused in the detection process due to their similarities. Hence, improving the detection classification is of essential necessity in object detection tasks except for location detection, which can be accomplished by the proposed SuperYOLO with better performance.

TABLE VIII summarizes the performance on the YOLOv3, YOLOv4, and YOLOv5s-x and our proposed SuperYOLO. Note that the AP scores of multimodal modes are significantly higher than those of unimodal (RGB or IR) modes for most classes. The overall mAP$_{50}$ of multimodal modes outperforms those of RGB or IR modes. These results confirm that multimodal fusion is an effective and efficient strategy for object detection based on information complementation between multimodal input. However, it should be noted that the slight increase of parameters and GFLOPs with multimodal fusion reflects the necessity of choice pixel-level fusion.

It is obvious that the SuperYOLO achieves higher AP and mAP$_{50}$ than the other frameworks. In particular, the SuperYOLO outperforms the YOLOv5x by a 10.96% mAP$_{50}$ score in multimodal mode. Meanwhile, the GFOLPs and parameter size of SuperYOLO are about 18.1x and 4.2x less than YOLOv5x. In addition, it can be noticed that the top performance is achieved for the classes of **Car**, **Pickup**, **Tractor** and **Camping**, which has most training instances as shown in TABLE II. Especially, the detection performance in **Boat** and **Van** is significantly improved in the SuperYOLO compared with other methods. YOLOv5s performs superior on GFLOPs, which depends on the Focus module to slim the input image, but results in lousy detection performance, especially for small objects. The SuperYOLO performs 16.82% better than YOLOv5s. Our proposed SuperYOLO shows a favorable speed-accuracy trade-off compared to the state-of-the-art models.

## VI. Conclusion and Future Work

In this paper, we have presented SuperYOLO, a real-time lightweight network that is built on top of the widely-used YOLOv5s to improve the detection performance of small objects in RSI. First, we have modified the baseline network by removing the Focus module to avoid resolution degradation, through which the baseline is significantly improved and overcomes the missing error of small objects. Second, we have conducted pixel-level fusion of multimodality to improve the detection performance based on mutual information. Lastly and most importantly, we have introduced a simple and flexible SR branch facilitating the backbone to construct a HR representation feature, by which small objects can be easily recognized from vast backgrounds with merely LR input required. We remove the SR branch in the inference stage, accomplishing the detection without changing the original structure of the network to achieve the same GFOLPs. With joint contributions of these ideas, the proposed SuperYOLO achieves 73.61% mAP$_{50}$ with lower computation cost on VEDAI dataset, which is 16.82% higher than that of YOLOv5s, and 10.96% higher than that of YOLOv5x.

The performance and inference ability of our proposal highlight the value of SR in remote sensing tasks, paving way for the future study of multimodal object detection. Our future interests will be focusing on the design of low-parameter mode to extract HR features, thereby further satisfying real-time and high-accuracy motivations.

## References

[1] R. Girshick, D. Jeff, D. Trevor, and M. Jitendra, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 580–587.

[2] R. Girshick, "Fast r-cnn," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

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

[4] P. Tang, X. Wang, X. Bai, and W. Liu, "Multiple instance detection network with online instance classifier refinement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 3059–3067.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "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, 2016.

[6] D. Jia, D. Wei, R. Socher, J. Lili, K. Li, and F. Li, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2009, pp. 248–255.

[7] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[9] Z. Zheng, Y. Zhong, J. Wang, and A. Ma, "Foreground-aware relation network for geospatial object segmentation in high spatial resolution remote sensing imagery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 4096–4105.

[10] J. Pang, C. Li, J. Shi, Z. Xu, and H. Feng, "$\mathcal{R}^2$-CNN: Fast tiny object detection in large-scale remote sensing images," *IEEE Trans. on Geosci. and Remote Sens.*, vol. 57, no. 8, pp. 5512–5524, 2019.

[11] Z. Deng, H. Sun, S. Zhou, J. Zhao, L. Lei, and H. Zou, "Multi-scale object detection in remote sensing imagery with convolutional neural networks," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 3–22, 2018.

[12] J. Ding, N. Xue, Y. Long, G. Xia, and Q. Lu, "Learning RoI transformer for oriented object detection in aerial images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 2844–2853.

[13] Z. Liu, H. Wang, L. Weng, and Y. Yang, "Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 8, pp. 1074–1078, 2016.

[14] D. Hong, L. Gao, N. Yokoya, J. Yao, J. Chanussot, Q. Du, and B. Zhang, "More diverse means better: Multimodal deep learning meets remote-sensing imagery classification," *IEEE Trans. on Geosci. and Remote Sens.*, vol. 59, no. 5, pp. 4340–4354, 2021.

[15] G. J. et al., "ultralytics/yolov5: v5.0," 2021. [Online]. Available: https://github.com/ultralytics/yolov5

[16] S. Zhang, M. Chen, J. Chen, F. Zou, Y.-F. Li, and P. Lu, "Multimodal feature-wise co-attention method for visual question answering," *Inf. Fusion*, vol. 73, pp. 1–10, 2021.

[17] Y. Chen, J. Shi, C. Mertz, S. Kong, and D. Ramanan, "Multimodal object detection via bayesian fusion," *arXiv*, 2021. [Online]. Available: https://arxiv.org/abs/2104.02904

[18] Q. Chen, K. Fu, Z. Liu, G. Chen, H. Du, B. Qiu, and L. Shao, "EF-Net: A novel enhancement and fusion network for rgb-d saliency detection," *Pattern Recognit.*, vol. 112, p. 107740, 2021.

[19] H. Zhu, M. Ma, W. Ma, L. Jiao, S. Hong, J. Shen, and B. Hou, "A spatial-channel progressive fusion resnet for remote sensing classification," *Inf. Fusion*, vol. 70, pp. 72–87, 2021.

[20] Y. Sun, Z. Fu, C. Sun, Y. Hu, and S. Zhang, "Deep multimodal fusion network for semantic segmentation using remote sensing image and lidar data," *IEEE Trans. on Geosci. and Remote Sens.*, 2021.

[21] W. Li, Y. Gao, M. Zhang, R. Tao, and Q. Du, "Asymmetric feature fusion network for hyperspectral and sar image classification," *IEEE Trans. on Neural Netw. and Learn. Syst.*, 2022.

[22] Y. Gao, W. Li, M. Zhang, J. Wang, W. Sun, R. Tao, and Q. Du, "Hyperspectral and multispectral classification for coastal wetland using depthwise feature interaction network," *IEEE Trans. on Geosci. and Remote Sens.*, 2021.

[23] M. Sharma, M. Dhanaraj, S. Karnam, D. G. Chachlakis, R. Ptucha, P. P. Markopoulos, and E. Saber, "Yolors: Object detection in multimodal remote sensing imagery," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 14, pp. 1497–1508, 2021.

[24] L. Gómez-Chova, D. Tuia, G. Moser, and G. Camps-Valls, "Multimodal classification of remote sensing images: A review and future directions," *Proc. IEEE*, vol. 103, no. 9, pp. 1560–1584, 2015.

[25] J. Noh, W. Bae, W. Lee, J. Seo, and G. Kim, "Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 9725–9734.

[26] M. Haris, G. Shakhnarovich, and N. Ukita, "Task-driven super resolution: Object detection in low-resolution images," *arXiv*, 2018. [Online]. Available: https://arxiv.org/abs/1803.11316

[27] J. Shermeyer and A. Van Etten, "The effects of super-resolution on object detection performance in satellite imagery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2019, pp. 1432–1441.

[28] L. Courtrai, M. Pham, and S. Lefèvre, "Small object detection in remote sensing images based on super-resolution with auxiliary generative adversarial networks," *Remote Sens.*, vol. 12, no. 19, p. 3152, 2020.

[29] J. Rabbi, N. Ray, M. Schubert, S. Chowdhury, and D. Chao, "Small-object detection in remote sensing images with end-to-end edge-enhanced gan and object detector network," *Remote Sens.*, vol. 12, no. 9, p. 1432, 2020.

[30] H. Ji, Z. Gao, T. Mei, and B. Ramesh, "Vehicle detection in remote sensing images leveraging on simultaneous super-resolution," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 4, pp. 676–680, 2019.

[31] L. Wang, D. Li, Y. Zhu, L. Tian, and Y. Shan, "Dual super-resolution learning for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 3773–3782.

[32] C. Wang, H. Mark Liao, Y. Wu, P. Chen, J. Hsieh, and I. Yeh, "CSPNet: A new backbone that can enhance learning capability of cnn," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2020, pp. 1571–1580.

[33] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Netw.*, vol. 107, pp. 3–11, 2018.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, 2015.

[35] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 2117–2125.

[36] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 8759–8768.

[37] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[38] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2017, pp. 136–144.

[39] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans. on Comput. Imaging*, vol. 3, no. 1, pp. 47–57, 2016.

[40] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery: A small target detection benchmark," *J. Vis. Commun. Image Represent.*, vol. 34, pp. 187–203, 2016.

[41] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. 19th Int. Conf. Comput. Statist.*, 2010, pp. 177–186.