YOLO-HR: Improved YOLOv5 for Object Detection in High-Resolution Optical Remote Sensing Images

Dahang Wan¹,², Rongsheng Lu¹,²,*, Sailei Wang¹,², Siyuan Shen¹,², Ting Xu¹,² and Xianli Lang¹,²

¹ School of Instrument Science and Opto-Electronic Engineering, Hefei University of Technology, Hefei 230009, China
² Anhui Province Key Laboratory of Measuring Theory and Precision Instrument, Hefei University of Technology, Hefei 230009, China
* Correspondence: rslu@hfut.edu.cn

Abstract: Object detection is essential to the interpretation of optical remote sensing images and can serve as a foundation for research into additional visual tasks that utilize remote sensing. However, the object detection network currently employed in optical remote sensing images underutilizes the output of the feature pyramid, so there remains potential for an improved detection. At present, a suitable balance between the detection efficiency and detection effect is difficult to attain. This paper proposes an enhanced YOLOv5 algorithm for object detection in high-resolution optical remote sensing images, utilizing multiple layers of the feature pyramid, a multi-detection-head strategy, and a hybrid attention module to improve the effect of object-detection networks for use with optical remote sensing images. According to the SIMD dataset, the mAP of the proposed method was 2.2% better than YOLOv5 and 8.48% better than YOLOX, achieving an improved balance between the detection effect and speed.

Keywords: object detection; remote sensing image; attention mechanism; large resolution image; feature reuse; deep learning

1. Introduction

With the rapid development of remote sensing technology, high-resolution optical remote sensing images have been utilized to depict numerous items on the Earth’s surface, including aircrafts, automobiles, buildings, etc. [1]. Object detection plays a crucial role in the interpretation of remote sensing images and can be used for their segmentation [2,3], description [4,5], and target tracking [6]. However, aerial optical remote sensing images manifest a diversity of scale, viewpoint specificity, random orientation, and high background complexity due to their relatively large field of view and the necessity for a high altitude [7,8], whereas the majority of conventional datasets contain ground-level views. Therefore, the object-detection techniques used in the construction of artificial features traditionally have a poor track record for accuracy and speed. The target-detection algorithm based on a convolutional neural network is significantly more efficient and effective than traditional target-detection algorithms. Due to the needs of society and supported by the development of deep learning, the use of neural networks for target detection in optical remote sensing images is a necessity.

Current object-detection algorithms incorporating deep learning to analyze optical remote sensing photographs can be classified as supervised, poorly supervised, or unsupervised. However, due to the complexity and instability of unsupervised and weakly supervised algorithms, supervised algorithms are the most frequently used. Moreover, supervised object-detection algorithms can be classified as either single-stage or two-stage. For instance, the two-stage capsule network SAHR-CapsNet [9] can accurately detect targets in remote sensing pictures. Due to the comparatively late discovery of capsule networks, however, the vast majority of modern two-stage object-detection algorithms...
have been based on the RCNN [10–12] series. The methods described by previous researchers [13] have integrated the techniques of dilation convolution [14] and OHEM [15] into Faster RCNN [12] frameworks to improve the detection precision of small objects with a high density in optical remote sensing images. A similar practice is described in reference [16]. Wang et al. [17] proposed a fine detector with contextual information to improve the region suggestion network (RPN) in Faster RCNN to overcome background clutter and difficulties in recognition of foreground items in remote sensing images. The detection of airports and ports in downsampled satellite images, followed by mapping the discovered items back to the original ultra-high resolution satellite images, can successfully enable the concurrent detection of objects of varying sizes, according to research [18], based on the assumption that airplanes are often located in airports and ships are in ports and oceans. Weng et al. [19] proposed a rotating object-detection approach based on RCNN [10] to improve the accuracy of object detection in remote sensing images by addressing the randomization of target orientation.

Although two-stage object-detection algorithms are comparatively advantageous in terms of the accuracy and detection effect, they tend to involve complex models and operate at a slower speed, so researchers have focused on single-stage algorithms allowing for a trade-off of the speed and accuracy, such as YOLO [20–25], SSD [26,27], MobileNet [28–30], etc., adding an attention mechanism that improves the network’s ability to detect remote sensing images. Previous research [31] integrated the attention mechanism CBAM [32] into the lightweight YOLOX network [33] to improve its detection accuracy for small targets in remote sensing image datasets. Another study [34] replaced the backbone of YOLOv3 with the backbone of MobileNetV3. It combined the attention mechanism to develop a lightweight single-stage object detection network SeMo-YOLO with an increased detection speed in the remote sensing object-detection network. However, instead of using the output of FPN and PANET simultaneously, the aforementioned YOLO-based network uses only one. However, the feature pyramid’s output has been underutilized, and the detection impact could still be enhanced. The single-stage object detection network is a dense anchor box network with the problem of a positive and negative sample imbalance [35], so previous researchers [36] added focal loss [35] into the training process to alleviate the positive and negative sample imbalance and used the self-attention mechanism to extract high-level semantic information from the depth-feature map for target spatial localization, thereby improving the accuracy of the SSD model [26] for target localization. The aforementioned single-stage target-detection technique, which is used in remote sensing photographs, detects targets more quickly than a two-stage target detection network as it fixes the network input to a particular size, such as 640 × 640 or 512 × 512. However, small targets (dozens of pixels or even fewer) tend to be lost when this method is employed for high-resolution remote sensing target-detection datasets, leading to the poor detection of small targets and a deterioration in the model’s overall detection effect caused by a reduction in the resolution [36–38].

The aforementioned techniques are crucial for object detection in remote sensing images, yet the following issues remain.

1. Some researchers employed a two-stage model for object recognition, which is characterized by a high model complexity, a large number of parameter calculations, and a sluggish performance.
2. Some researchers used single-stage networks for object detection in optical remote sensing images. However, most of them are scaled down to a lower input resolution, which diminishes the effectiveness of model detection.
3. The majority of YOLO-based networks utilize either the output of FPN [39] or PAFPN [40], but not both.
4. Some researchers have demonstrated that the hybrid attention mechanism can enhance the precision of object detection in optical remote sensing images. Nevertheless, hybrid attention modules are scarce now.
We proposed a lightweight object detection network for high-resolution remote sensing images based on the YOLOv5 framework in order to balance the detection accuracy, speed, the number of model parameters, and the existing features. The following is a summary of the contributions of this paper:

1. Based on the YOLOv5 network topology, a single-stage object detection network named YOLO-HR for high-resolution optical remote sensing photographs was suggested.
2. A multi-detection-head approach that can exploit the features of both FPN and PANET was proposed.
3. A lightweight hybrid attention module was proposed.
4. The model’s efficiency and viability were validated using the SIMD [41] dataset.

The article is structured as follows: Section 2 presents a brief summary of the related works from the remote sensing object detection dataset, object detection network, and attention mechanism. Based on an analysis of the existing detection head output strategy in single-stage algorithms, Section 3 offers a multi-detection-head strategy and the YOLO-HR network. Section 4 evaluates the proposed technique on the SIMD datasets and presents the experimental results. The paper is summarized in Section 5.

2. Related Work


Traditional remote sensing image datasets for object detection are reviewed in Table 1. Where Dataset represents the name of the dataset, Categories denotes the number of categories in the dataset, Images is the number of images, Instances represents the total number of targets, and Year represents the release year of the dataset. The recently published SIMD [41] dataset was utilized for this study, with the majority of its images measuring 1024 by 768 pixels.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Categories</th>
<th>Images</th>
<th>Instances</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAS [42]</td>
<td>1</td>
<td>30</td>
<td>1319</td>
<td>2008</td>
</tr>
<tr>
<td>NWPU VHR-10 [44]</td>
<td>10</td>
<td>800</td>
<td>3775</td>
<td>2014</td>
</tr>
<tr>
<td>VEDAI [45]</td>
<td>9</td>
<td>1210</td>
<td>3640</td>
<td>2015</td>
</tr>
<tr>
<td>UCAS-AOD [46]</td>
<td>2</td>
<td>910</td>
<td>6029</td>
<td>2015</td>
</tr>
<tr>
<td>DLR-MVDA [47]</td>
<td>2</td>
<td>20</td>
<td>14,235</td>
<td>2015</td>
</tr>
<tr>
<td>HRSC-2016 [48]</td>
<td>1</td>
<td>1070</td>
<td>2976</td>
<td>2016</td>
</tr>
<tr>
<td>RSOD [1]</td>
<td>4</td>
<td>976</td>
<td>6950</td>
<td>2017</td>
</tr>
<tr>
<td>DOTA [49]</td>
<td>15</td>
<td>2806</td>
<td>188,282</td>
<td>2017</td>
</tr>
<tr>
<td>DIOR [1]</td>
<td>20</td>
<td>23,463</td>
<td>192,472</td>
<td>2018</td>
</tr>
<tr>
<td>LEVIR [50]</td>
<td>3</td>
<td>21,952</td>
<td>10,069</td>
<td>2018</td>
</tr>
<tr>
<td>ITCVD [51]</td>
<td>1</td>
<td>173</td>
<td>29,088</td>
<td>2018</td>
</tr>
<tr>
<td>SIMD [52]</td>
<td>15</td>
<td>5000</td>
<td>45,303</td>
<td>2020</td>
</tr>
</tbody>
</table>

2.2. Attention Mechanism

Currently, attention mechanisms in deep learning are commonly categorized as soft attention, hard attention, and self-attention. The soft attention mechanism assigns a weight between 0 and 1 to each input item and evaluates the majority of the data, but not equally. The hard attention mechanism assigns a weight of 0 or 1 to each input item. Unlike soft attention, hard attention just considers the component that demands attention and promptly discards unnecessary information. The self-attention mechanism assigns a weight to each input item based on the interaction between the input items, i.e., the “voting” between the input items determines which input items receive attention. The soft attention method is the most widespread in the field of remote sensing image object detection, and its representative articles include SE [53], CBAM [32], ECA [54], Co-Attention [55], Reverse Attention [56], Cross Attention [57], etc. Numerous articles [31,58–62] have demonstrated that mixed attention mechanisms improved the effect of a remote sensing target detection network,
including strengthening the detection effect and increasing the detection accuracy. In this study, the hybrid attention module MAB also consisted of hybrid soft attention mechanisms.

2.3. Object Detection Networks in Remote Sensing Image

Radar-based object detection and optical remote sensing object detection are the two types of remote sensing picture object detection. Optical sensors require favorable weather conditions and ample sunshine to produce high-quality photographs. The most notable advantage of radar sensors is that they are unaffected by the weather [63]. For example, the synthetic aperture radar (SAR) is a high-resolution image radar that can detect camouflage and penetrate masking objects in all-weather situations. One of the current hot topics in remote sensing is the application of neural networks to detect SAR images with complex and variable scenes [63–66]. This paper focuses on object detection in optical remote sensing images. Typically, image data are often derived from satellite photographs, such as Google Earth, or aerial images, such as UAS. The recent applications of deep learning to recognize objects in optical remote sensing images have produced satisfactory results. Wang et al. [67–69] took advantage of the advancements in the Faster RCNN [12], RetinaNet [35], and YOLOv3 [22] networks to detect wildlife in high-resolution UAV images. Sun et al. proposed a comprehensive partial-based convolutional neural network called PBNet for composite object detection in high-resolution optical remote sensing images [70]. In the past, RCNN was used to identify aircraft targets in very high-resolution remote sensing photographs with a poor precision and sluggish speed. Therefore, a mix of dense convolutional networks, multi-scale representation methods, and a number of enhancement techniques were utilized to strengthen the fundamental VGG16-Net’s structure, raise accuracy, and more effectively recognize the target in satellite optical remote sensing images. [13]. The experiments mentioned [13,35,63–70] above used horizontal boundary boxes (HBB), which sometimes do not offer precise direction and scale information and have an excessive number of superfluous pixels in the backdrop. In addition, HBB and non-maximal inhibition (NMS) collaboration usually leads to missing detection when detecting objects with high aspect ratios and dense parking. In recent years, the recognition of directional objects (OBB) in RS images has garnered growing attention [71–76]. OBBs are often slower than HBBs in training and deployment. Hence, HBBs remain the focus of the current research. The algorithm of this study was also based on HBB.

The majority of those above high-resolution optical remote sensing target detection algorithms, which are typically classified into single-stage and double-stage target detection networks, are based on the existing mainstream target detection networks. RCNN [10], SPPNet [77], Fast RCNN [11], Faster RCNN [12], Cascade RCNN [78], etc., are examples of two-stage ones. The single-stage products include RetinaNet [35], SSD [26,27], FCOS [79], CenterNet [80], CornerNet [81], YOLOv1–7 [20–25,82], YOLOx [33], and YOLOF [83]. Traditional lightweight online models such as MobileNet [28–30], ShuffleNet [84,85], and Efficientdet [86] are also available.

3. Materials and Methods

3.1. YOLO-HR

3.1.1. Comparison of Prediction Head

The majority of the current YOLO series detection heads are based on the output feature of FPN and PAFPN, where FPN-based networks such as YOLOv3 and its variants are shown in Figure 1a, which directly utilize the one-way fused features for the output, and the PAFPN-based algorithms of YOLOv4 and YOLOv5 add a low-level to high-level channel on top of this, which directly transmits the low-level information upwards (Figure 1b). As demonstrated in Figure 1c and similarly in some studies [87–89], Zhu et al. added a detection head for a particular detection task in the TPH-YOLOv5 model. In Figure 1b,c, only the PAFPN features are used for the output, while the FPN features are underutilized. Therefore, YOLOv7 attaches three auxiliary heads to the FPN output, as depicted in
Figure 1d, although the auxiliary heads are only used for a “rough selection” and have a low weight. The detecting head of SSD was proposed to improve the too-coarse design of the anchor set by the YOLO network, as depicted in Figure 1e, and the design concept consists mostly of a dense anchor design with multiple aspect ratios at multiple scales. Inspired by Figure 1c–e, this paper proposed a multi-detection-head strategy for the YOLO detection head, as depicted in Figure 1f, which could utilize the feature information of PANet and FPN simultaneously. Additionally, an output head was added directly at the 64-fold downsampling, which caused the network to contain the prior global information.

3. Materials and Methods

3.1. YOLO-HR

3.1.1. Comparison of Prediction Head

The majority of the current YOLO series detection heads are based on the output feature of FPN and PAFPN, where FPN-based networks such as YOLOv3 and its variants are shown in Figure 1a, which directly utilize the one-way fused features for the output, and the PAFPN-based algorithms of YOLOv4 and YOLOv5 add a low-level to high-level channel on top of this, which directly transmits the low-level information upwards (Figure 1b). As demonstrated in Figure 1c and similarly in some studies [87–89], Zhu et al. added a detection head for a particular detection task in the TPH-YOLOv5 model. In Figure 1b, c, only the PAFPN features are used for the output, while the FPN features are underutilized. Therefore, YOLOv7 attaches three auxiliary heads to the FPN output, as depicted in Figure 1d, although the auxiliary heads are only used for a “rough selection” and have a low weight. The detecting head of SSD was proposed to improve the too-coarse design of the anchor set by the YOLO network, as depicted in Figure 1e, and the design concept consists mostly of a dense anchor design with multiple aspect ratios at multiple scales. Inspired by Figure 1c–e, this paper proposed a multi-detection-head strategy for the YOLO detection head, as depicted in Figure 1f, which could utilize the feature information of PANet and FPN simultaneously. Additionally, an output head was added directly at the 64-fold downsampling, which caused the network to contain the prior global information.

Figure 1. Comparison of the output of various detection heads. (a) FPN-based, (b) PANet-based, (c) TPH-YOLOv5, (d) Lead Head + Auxiliary Head, (e) SSD-based, (f) YOLO-HR (Ours).

3.1.2. Overall Structure of YOLO-HR

The multi-detection-head method could efficiently use the network’s output features. YOLO-HR was an object detection network for high resolution remote sensing photographs. As depicted in Figure 2, the YOLO-HR network described in this paper can be separated into Backbone, Neck, and Head. The basic structure of Backbone was a CSP-DenseNet with C3 and Convolutional modules at its core. After the data enhancement, images were fed into the network and numerous convolutional modules retrieved features after channel mixing by the Conv module with a kernel size = 6. They were connected to PANet in Neck after the feature enhancement module named SPPF. Bidirectional feature fusion was undertaken to enhance the network’s detecting capability. Conv2d was used to independently scale the fused feature layers to generate the multi-layer outputs. As depicted in Figure 3a, the NMS algorithm combined the outputs of all single-layer detectors to produce the final detection frame.
Figure 2. The overall structure of YOLO-HR.

Figure 3. Composition modules of YOLO-HR. (a) The principle of YOLO-HR multi-head output; (b) the other composition modules of YOLO-HR.

Figure 3b depicts the structural composition of each module of the YOLO-HR network. Conv comprises a 2D convolutional layer, BN layer batch normalization, and Silu activation function, C3 comprises two 2D convolutional layers plus a bottleneck layer, and Upsample is the upsampling layer. The SPPF module is a sped-up version of the SPP module, and the MAB module is depicted in Figure 2, where the ECA [54] is depicted in the bottom left corner. After channel-level global average pooling without dimension reduction, the ECA is efficiently performed using the rapid 1D convolution of size k to capture local cross-channel interaction information, taking into account each channel’s relationship with its k neighbors. The CA attention mechanism [55] is depicted in Figure 1’s lower right corner, which encodes each channel along the horizontal and vertical coordinates, respectively, using a channel-level global average pooling of size (H,1) or (1, W) pooling kernel. The above two transformations collect features along two spatial directions to produce a pair of direction-aware feature maps, which are then concatenated and modified with convolution and Sigmoid functions to provide the attention output.

Table 2 displays the parameter settings for the entire network’s structure. Input displays the input size of the image, Output displays the output size of the current layer,
Argvs are the input parameters of the current module, From represents the input source of the current layer, N represents the number of repetitions of the current module, and Parameters displays the size of the parameter number of the current layer.

Table 2. Parameter setting of the network structure.

<table>
<thead>
<tr>
<th>ID</th>
<th>Module</th>
<th>From</th>
<th>N</th>
<th>Args</th>
<th>Output</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Input</td>
<td></td>
<td></td>
<td></td>
<td>1024 x 1024 3</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Conv</td>
<td>−1</td>
<td>1</td>
<td>(3, 32, 6, 2, 2)</td>
<td>512 x 512 x 32</td>
<td>3520</td>
</tr>
<tr>
<td>2</td>
<td>C3</td>
<td>−1</td>
<td>1</td>
<td>(32, 64, 3, 2)</td>
<td>256 x 256 x 64</td>
<td>18,560</td>
</tr>
<tr>
<td>3</td>
<td>Conv</td>
<td>−1</td>
<td>1</td>
<td>(64, 128, 3, 2)</td>
<td>128 x 128 x 128</td>
<td>73,984</td>
</tr>
<tr>
<td>4</td>
<td>C3</td>
<td>−1</td>
<td>2</td>
<td>(128, 128, 2)</td>
<td>128 x 128 x 128</td>
<td>115,712</td>
</tr>
<tr>
<td>5</td>
<td>Conv</td>
<td>−1</td>
<td>1</td>
<td>(128, 256, 3, 2)</td>
<td>64 x 64 x 256</td>
<td>295,454</td>
</tr>
<tr>
<td>6</td>
<td>C3</td>
<td>−1</td>
<td>3</td>
<td>(256, 256, 3)</td>
<td>64 x 64 x 256</td>
<td>625,152</td>
</tr>
<tr>
<td>7</td>
<td>Conv</td>
<td>−1</td>
<td>1</td>
<td>(256, 384, 3, 2)</td>
<td>32 x 32 x 384</td>
<td>885,504</td>
</tr>
<tr>
<td>8</td>
<td>C3</td>
<td>−1</td>
<td>1</td>
<td>(384, 384, 1)</td>
<td>32 x 32 x 384</td>
<td>665,856</td>
</tr>
<tr>
<td>9</td>
<td>Conv</td>
<td>−1</td>
<td>1</td>
<td>(384, 512, 3, 2)</td>
<td>16 x 16 x 512</td>
<td>1,770,496</td>
</tr>
<tr>
<td>10</td>
<td>C3</td>
<td>−1</td>
<td>1</td>
<td>(512, 512, 1)</td>
<td>16 x 16 x 512</td>
<td>1,182,720</td>
</tr>
<tr>
<td>11</td>
<td>SPPF</td>
<td>−1</td>
<td>1</td>
<td>(512, 512, 5)</td>
<td>16 x 16 x 512</td>
<td>656,896</td>
</tr>
<tr>
<td>12</td>
<td>Conv</td>
<td>−1</td>
<td>1</td>
<td>(512, 384, 1, 1)</td>
<td>16 x 16 x 374</td>
<td>197,376</td>
</tr>
<tr>
<td>13</td>
<td>Upsample</td>
<td>−1</td>
<td>1</td>
<td>(None, 2, 'nearest')</td>
<td>32 x 32 x 384</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>Concat</td>
<td>(−1, 8)</td>
<td>1</td>
<td>(1)</td>
<td>32 x 32 x 768</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>C3</td>
<td>−1</td>
<td>1</td>
<td>(768, 384, 1, False)</td>
<td>32 x 32 x 384</td>
<td>813,312</td>
</tr>
<tr>
<td>16</td>
<td>Conv</td>
<td>−1</td>
<td>1</td>
<td>(384, 256, 1, 1)</td>
<td>32 x 32 x 256</td>
<td>98,816</td>
</tr>
<tr>
<td>17</td>
<td>Upsample</td>
<td>−1</td>
<td>1</td>
<td>(None, 2, 'nearest')</td>
<td>64 x 64 x 256</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>Concat</td>
<td>(−1, 6)</td>
<td>1</td>
<td>(1)</td>
<td>64 x 64 x 512</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>C3</td>
<td>−1</td>
<td>1</td>
<td>(512, 256, 1, False)</td>
<td>64 x 64 x 256</td>
<td>361,984</td>
</tr>
<tr>
<td>20</td>
<td>Conv</td>
<td>−1</td>
<td>1</td>
<td>(256, 128, 1, 1)</td>
<td>64 x 64 x 128</td>
<td>33,024</td>
</tr>
<tr>
<td>21</td>
<td>Upsample</td>
<td>−1</td>
<td>1</td>
<td>(None, 2, 'nearest')</td>
<td>128 x 128 x 128</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>Concat</td>
<td>(−1, 4)</td>
<td>1</td>
<td>(1)</td>
<td>128 x 128 x 256</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>C3</td>
<td>−1</td>
<td>1</td>
<td>(256, 128, 1, False)</td>
<td>128 x 128 x 128</td>
<td>90,880</td>
</tr>
<tr>
<td>24</td>
<td>Conv</td>
<td>−1</td>
<td>1</td>
<td>(128, 128, 3, 2)</td>
<td>64 x 64 x 128</td>
<td>147,712</td>
</tr>
<tr>
<td>25</td>
<td>Concat</td>
<td>(−1, 20)</td>
<td>1</td>
<td>(1)</td>
<td>64 x 64 x 256</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>C3</td>
<td>−1</td>
<td>1</td>
<td>(256, 256, 1, False)</td>
<td>64 x 64 x 256</td>
<td>296,448</td>
</tr>
<tr>
<td>27</td>
<td>Conv</td>
<td>−1</td>
<td>1</td>
<td>(256, 256, 3, 2)</td>
<td>32 x 32 x 256</td>
<td>590,336</td>
</tr>
<tr>
<td>28</td>
<td>Concat</td>
<td>(−1, 16)</td>
<td>1</td>
<td>(1)</td>
<td>32 x 32 x 512</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>C3</td>
<td>−1</td>
<td>1</td>
<td>(512, 384, 1, False)</td>
<td>32 x 32 x 384</td>
<td>715,008</td>
</tr>
<tr>
<td>30</td>
<td>Conv</td>
<td>−1</td>
<td>1</td>
<td>(384, 384, 3, 2)</td>
<td>16 x 16 x 384</td>
<td>1,327,872</td>
</tr>
<tr>
<td>31</td>
<td>Concat</td>
<td>(−1, 11)</td>
<td>1</td>
<td>(1)</td>
<td>16 x 16 x 896</td>
<td>0</td>
</tr>
<tr>
<td>32</td>
<td>C3</td>
<td>−1</td>
<td>1</td>
<td>(896, 512, 1, False)</td>
<td>16 x 16 x 512</td>
<td>1,379,328</td>
</tr>
<tr>
<td>33</td>
<td>MAB</td>
<td>11</td>
<td>1</td>
<td>(512, 512)</td>
<td>16 x 16 x 512</td>
<td>1,361,477</td>
</tr>
<tr>
<td>34</td>
<td>Detect</td>
<td></td>
<td></td>
<td></td>
<td>(23, 26, 20, 16, 29, 32, 33)</td>
<td>81406 x (5 + 15)</td>
</tr>
</tbody>
</table>

3.2. Data Augmentation

The essence of data augmentation is to artificially introduce human visual prior knowledge, which can improve the performance of the model very well, and it has basically become the standard for model training. The more commonly used geometric transformation methods are flip, rotate, crop, scale, pan, dither, etc. The pixel transformation methods include adding pretzel and Gaussian noise, performing a Gaussian blur, adjusting the HSV contrast, and adjusting the brightness, saturation, histogram equalization, white balance, etc. In addition to the above methods, this paper also uses a variety of data enhancement methods in the training phase, each with different random ratios, such as Mosica, CUTOUT, small target replication, etc. Among them, flip and rotation are used to solve the problem of Angle diversity in remote sensing images, zoom and shift are used to solve the problem of the multi-scale in remote sensing images, dithering and adding noise are used to improve the problem of a complex background in remote sensing images, and small target replication is used to expand the samples and improve the detection effect of small targets.
3.3. Loss Function

The loss function of YOLO-HR was composed of three components: target confidence loss, target category loss, and target positioning loss. The loss function could be expressed as follows:

\[ L_{all} = \lambda_{conf} L_{conf} + \lambda_{cls} L_{cls} + \lambda_{loc} L_{loc} \]  

where \( L_{all} \) contained three hyperparameters, the weight of each component, which could be modified before training based on the actual circumstances. In this work, the corresponding weights of the three sections were 1.0, 0.5, and 0.05. The target confidence loss utilized the BCE (binary cross-entropy) loss, with the following expression:

\[
\begin{align*}
L_{conf} &= - \sum_{i=1}^{K} \sum_{j=1}^{B} I_{ij}^{obj} [C_i \log(C_i') + (1 - C_i') \log(1 - C_i')] \\
&- \lambda_{noobj} \sum_{i=1}^{K} \sum_{j=1}^{B} I_{ij}^{noobj} [C_i \log(C_i') + (1 - C_i') \log(1 - C_i')] 
\end{align*}
\]

Among them, \( K \times K \) could take on three distinct values, with the particular size being dependent on the image size. Taking 1024 × 1024 as an example, they were 16 × 16, 32 × 32, 64 × 64, and 128 × 128, respectively, illustrating the number of grids on the feature maps generated by YOLO-HR at three different scales. \( B \) represented the number of preceding boxes. \( I_{ij}^{obj} \) specifies whether the \( j \)th previous box of the \( i \)th grid had a prediction target. \( I_{ij}^{obj} \) is 1 if the condition was met; else, it was 0. \( I_{ij}^{noobj} \) indicated if the \( j \)th previous box of the \( i \)th grid did not contain a predicted target. If not, \( I_{ij}^{noobj} \) was 1; otherwise, it was 0. \( C_i' \) and \( C_i \) represented the actual and expected confidence values, respectively. \( \lambda_{noobj} \) was a constant coefficient, typically assumed to be 0.5, that was used to balance the positive and negative samples.

The target confidence loss was also the BCE loss, and the expression is as follows:

\[
L_{conf} = - \sum_{i=1}^{K} \sum_{j=1}^{B} I_{ij}^{obj} \left\{ P_i^c(c) \log[P_i^c(c)] + \left[1 - P_i^c(c)\right] \log[1 - P_i^c(c)] \right\} 
\]

where \( K \times K, B \) and \( I_{ij}^{obj} \) were consistent with Equation (1), \( c \) was the target category, and \( P_i^c(c) \) and \( P_i^c(c) \) were the probability that the target in the \( j \)th prediction box in the \( i \)th grid belongs to the real value and the predicted value of a certain category, respectively.

The \( \text{SloU} \) loss [90] replaced the \( \text{Clou} \) loss [91] function for the target positioning loss in order to increase the training speed and reasoning precision in this paper. The following is the formula:

\[
L_{box} = 1 - \text{IoU} + \frac{\Omega + \Delta}{2} 
\]

\[
\text{IoU} = \frac{|B \cap GT|}{|B \cup GT|} 
\]

\[
\Delta = \sum_{x,y} 1 - e^{-\gamma t} 
\]

\[
\Omega = \sum_{w,h} (1 - e^{-\theta}) \theta 
\]

where \( \Delta \) represented the Distance cost, \( \gamma \) the Angle cost, \( \Omega \) the Shape cost, and \( \theta \) expressed the Shape cost level of concern.

4. Experiment
4.1. Experimental Platform and Related Indexes
4.1.1. Experimental Platform

The experimental platform of this paper is shown in Table 3.
The SIMD dataset is a multi-category, open-source, high-resolution remote sensing object detection dataset containing a total of 15 classes, as illustrated in Figure 4. Additionally, the SIMD dataset is more distributed with small- and medium-sized targets \((w < 0.4, h < 0.4)\), and the detection head used by YOLO-HR proposed in this paper to detect this region is double the number of detection heads used by the common YOLO algorithm, so YOLO-HR has greater advantages on this dataset.

![Figure 4](image-url) Distribution of targets in SIMD dataset. (a) shows the distribution of the number of categories; (b) shows the distribution of target width and height in the image; the color from white to blue (from light to dark) indicates a more concentrated distribution.

### 4.1.2. Related Indexes

The network performance evaluation is mainly based on the mAP (mean average accuracy) during training and the performance of the trained network in the validation set. To measure the detection results quantitatively, the accuracy Precision, Recall, and mAP are used here as the performance evaluation metrics. The expressions of \(P\) and \(R\) are as follows.

\[
\begin{align*}
Recall &= \frac{TP}{TP + FN} \\
Precision &= \frac{TP}{TP + FP}
\end{align*}
\]

where True positives \((TP)\) are the number of samples that are actually positive and classified as positive by the classifier; True negatives \((TN)\) are the number of samples that are actually negative and classified as negative by the classifier; False positives \((FP)\) are the number of samples that are actually negative but classified as positive by the classifier; and False negatives \((FN)\) are the number of samples that are actually positive but classified as negative by the classifier.

Average Precision (AP) is the area enclosed by the P-R curve. Usually, the better the classifier, the higher the AP value. Mean Average Precision (mAP) is the AP of each category taken separately, and then the average of the AP of all categories is calculated, representing a composite measure of the average precision of the detected targets. AP50 in the later text means that the IoU threshold is greater than 0.5, mAP is mAP 0.5:0.95, and step is 0.05.
4.2. Experiments on SIMD
4.2.1. Ablation Test

It was possible to connect the output of the SPPF module to the output head and thus identify large targets in the image. However, the output of the SPPF module had multiple connections and is concerned with targets of multiple scales, so using it directly for the detection head to identify large objects would result in a poor model representation, as shown in Figure 5. Figure 5 depicts a visual comparison of the heat map of some detection findings prior to and following the addition of the MAB module. After adding the MAB module, this detection head focused on detecting large objects, while the prediction of small targets was assigned to other prediction heads and the expression effect of the model was improved, which was also more in line to divide the detection head based on the target size in the YOLO algorithm.

![Image](image_url)

Figure 5. Heat map visualization of the partial detection results before and after adding the MAB module. Both are visualized with Grad-CAM [92].

Using the calculation results of the YOLOv5s algorithm as a reference, the effects of the MPH output strategy and MAB module on the calculation results were examined in the SIMD dataset and 1024 × 1024 image resolution, as shown in Table 4 and Figure 6, respectively, from top to bottom, indicating that the increase in the modules is in order. Finetune means the model was pre-trained on the ImageNet dataset, and then the trained model was fine-tuned on the SIMD dataset. The results showed that after the addition of the MPH strategy and MAB module, the number of parameters in the model increased by 2.5 M. Still, the increase in the number of parameters was negligible compared with the disk capacity of hundreds of G. The speed was not significantly improved, but the AP50 of the model increased by 2.1%, mAP increased by 2.2%, and the accuracy increased by 1.5%. The recall rate increased by 1.19%.

<table>
<thead>
<tr>
<th>Name</th>
<th>Params (M)</th>
<th>FLOPS (G)</th>
<th>Speed (ms)</th>
<th>AP50 (%)</th>
<th>mAP (%)</th>
<th>P (%)</th>
<th>R (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLOv5s</td>
<td>6.72</td>
<td>15.9</td>
<td>5.8</td>
<td>81.51</td>
<td>62.8</td>
<td>75.01</td>
<td>79.66</td>
</tr>
<tr>
<td>YOLOv5s + Finetune</td>
<td>6.72</td>
<td>15.9</td>
<td>5.8</td>
<td>83.85</td>
<td>66.05</td>
<td>83.10</td>
<td>79.65</td>
</tr>
<tr>
<td>YOLOv5s + MPH</td>
<td>11.9</td>
<td>16.3</td>
<td>6.5</td>
<td>82.96</td>
<td>65.12</td>
<td>80.73</td>
<td>79.40</td>
</tr>
<tr>
<td>YOLOv5s + MPH + Finetune</td>
<td>11.9</td>
<td>16.3</td>
<td>6.5</td>
<td>85.15</td>
<td>66.68</td>
<td>83.47</td>
<td>80.30</td>
</tr>
<tr>
<td>YOLO-HR</td>
<td>13.2</td>
<td>16.6</td>
<td>6.7</td>
<td>83.61</td>
<td>65.0</td>
<td>76.51</td>
<td>80.85</td>
</tr>
<tr>
<td>YOLO-HR + Finetune</td>
<td>13.2</td>
<td>16.6</td>
<td>6.7</td>
<td>85.39</td>
<td>67.31</td>
<td>85.95</td>
<td>81.28</td>
</tr>
</tbody>
</table>
Using the calculation results of the YOLOv5s algorithm as a reference, the effects of the MPH output strategy and MAB module on the calculation results were examined in the SIMD dataset and 1024 × 1024 image resolution, as shown in Table 4 and Figure 6, respectively, from top to bottom, indicating that the increase in the modules is in order. Finetune means the model was pre-trained on the ImageNet dataset, and then the trained model was fine-tuned on the SIMD dataset. The results showed that after the addition of the MPH strategy and MAB module, the number of parameters in the model increased by 2.5 M. Still, the increase in the number of parameters was negligible compared with the disk capacity of hundreds of G. The speed was not significantly improved, but the AP50 of the model increased by 2.1%, mAP increased by 2.2%, and the accuracy increased by 1.5%. The recall rate increased by 1.19%.

Table 4. Performance improvement of each part design on the result.

<table>
<thead>
<tr>
<th>Name</th>
<th>Params (M)</th>
<th>FLOPS (G)</th>
<th>Speed (ms)</th>
<th>AP50 (%)</th>
<th>mAP (%)</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLOv5s</td>
<td>6.72</td>
<td>15.9</td>
<td>5.8</td>
<td>81.51</td>
<td>62.8</td>
<td>75.01</td>
<td>79.66</td>
</tr>
<tr>
<td>YOLOv5s + Finetune</td>
<td>6.72</td>
<td>15.9</td>
<td>5.8</td>
<td>83.85</td>
<td>66.05</td>
<td>83.10</td>
<td>79.65</td>
</tr>
<tr>
<td>YOLOv5s + MPH</td>
<td>11.9</td>
<td>16.3</td>
<td>6.5</td>
<td>82.96</td>
<td>65.12</td>
<td>80.73</td>
<td>79.40</td>
</tr>
<tr>
<td>YOLOv5s + MPH + Finetune</td>
<td>11.9</td>
<td>16.3</td>
<td>6.5</td>
<td>85.15</td>
<td>66.68</td>
<td>83.47</td>
<td>80.30</td>
</tr>
<tr>
<td>YOLO-HR</td>
<td>13.2</td>
<td>16.6</td>
<td>6.7</td>
<td>83.61</td>
<td>65.0</td>
<td>76.51</td>
<td>80.85</td>
</tr>
<tr>
<td>YOLO-HR + Finetune</td>
<td>13.2</td>
<td>16.6</td>
<td>6.7</td>
<td>85.59</td>
<td>67.31</td>
<td>85.95</td>
<td>81.28</td>
</tr>
</tbody>
</table>

Figure 6. Results comparison chart. (a) AP50, (b) mAP, (c) Precision, (d) Recall.

4.2.2. Comparison Experiments

Simultaneously, the classic YOLOv3-Tiny, Faster RCNN, YOLOv7, and YOLOX models were selected for comparison tests in this paper. The Yolov3-Tiny, YOLOv5 (DenseNet + PAN), and YOLO-HR codes and pre-training models utilized in this experiment were obtained through the YOLOv5 open-source framework and the YOLOv7 models through the YOLOv7 open-source framework. The YOLOX(Darknet-53 + FPN) algorithm was derived from the literature [33], while the other models, including Faster RCNN(Resnet-50 + FPN), were derived from the MMDetection [93] open-source framework. We tested and compared YOLO-HR and other algorithms at a 1024 by 1024 image resolution. We merely compared the number of parameters to prevent variations caused by the model storage methods of various formats. For instance, the amount of parameters in YOLOv5s is 7.11 M, but Pytorch’s model storage format is 14.4 M. In order to rule out randomness, the running time was computed as the average time for testing 1000 photos, as shown in Table 5. The suggested approach outperformed YOLOv5, YOLOv3-Tiny, YOLOv7-Tiny, YOLOX, and the Faster RCNN model using Resnet-50 as its backbone in terms of the detection outcomes (mAP and AP50). Although slightly more sophisticated than the YOLOv5 and YOLOX models, the number of references of a few meters was minimal, even compared to the modest storage space of edge devices such as Nvidia TX2 and NX, which was only 32 gigabytes, so it was more than sufficient. In terms of the speed, it was superior to YOLOv3-Tiny and Faster RCNN and the detection speed was only 0.5 ms higher than that of YOLOv5, without a substantial reduction in the detection speed. The complete detection findings of YOLO-HR proposed in this paper offered benefits over the appeal algorithm. The results of the experiments indicated that the YOLO-HR algorithm struck a more suitable balance between the reference number, speed, and detection effectiveness.
Table 5. Comparison with other algorithms.

<table>
<thead>
<tr>
<th>Name</th>
<th>AP50 (%)</th>
<th>mAP (%)</th>
<th>Params (M)</th>
<th>FLOPs (G)</th>
<th>Speed (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLOv3-Tiny [32,34]</td>
<td>77.23</td>
<td>54.53</td>
<td>8.3</td>
<td>12.9</td>
<td>4.3</td>
</tr>
<tr>
<td>YOLOv5n [24]</td>
<td>79.56</td>
<td>60.69</td>
<td>1.7</td>
<td>4.2</td>
<td>4.8</td>
</tr>
<tr>
<td>YOLOv5s [24]</td>
<td>83.85</td>
<td>66.05</td>
<td>6.72</td>
<td>15.9</td>
<td>5.8</td>
</tr>
<tr>
<td>YOLO-HR-n</td>
<td>83.01</td>
<td>64.04</td>
<td>3.34</td>
<td>4.4</td>
<td>6.3</td>
</tr>
<tr>
<td>YOLO-HR-s</td>
<td>85.59</td>
<td>67.31</td>
<td>13.2</td>
<td>16.6</td>
<td>6.7</td>
</tr>
<tr>
<td>YOLOX-s [33]</td>
<td>76.63</td>
<td>56.85</td>
<td>8.94</td>
<td>26.79</td>
<td>5.7</td>
</tr>
<tr>
<td>YOLOv7-s [25]</td>
<td>83.80</td>
<td>66.55</td>
<td>8.92</td>
<td>26.8</td>
<td>6.3</td>
</tr>
<tr>
<td>YOLOv7-tiny [25]</td>
<td>82.08</td>
<td>64.16</td>
<td>5.77</td>
<td>13.1</td>
<td>5.2</td>
</tr>
<tr>
<td>Faster RCNN [12]</td>
<td>77.74</td>
<td>-</td>
<td>41.19</td>
<td>198.47</td>
<td>26.3</td>
</tr>
</tbody>
</table>

Some of the detection results are shown in Figure 7. From each detection result, there was not much difference with other algorithms, but compared with other algorithms, the algorithm in this paper improved the detection effect of the model while ensuring no significant increase in the time consumption and enhanced the expression effect of the model by using the attention mechanism.

Figure 7. Comparison of the detection effect of different model detection parts.
4.2.3. Qualitative Results

Some qualitative results of the YOLO-HR algorithm proposed in this paper on the SIMD dataset are shown in Figure 8. As shown in the figure, the YOLO-HR model could better detect objects in remote sensing images with special viewing angles, including objects with complex backgrounds, random directions, and different scales.

![Some detection results on the SIMD dataset of YOLO-HR.](image)

Figure 8. Some detection results on the SIMD dataset of YOLO-HR.

5. Conclusions

To address the issue that the majority of the current models utilized for optical remote sensing image object detection underutilized the output features of the feature pyramid, we proposed a multi-head strategy based on prior work and we proposed a hybrid attention module, MAB, for the lack of hybrid attention mechanisms. Finally, we embedded the aforementioned two methods into the YOLOv5 network and presented a high resolution optical remote sensing target recognition algorithm named YOLO-HR. The YOLO-HR algorithm employed several detection heads for object detection and recycled the output features of the feature pyramid, allowing the network to enhance the detection effect further. The experiments indicate that the YOLO-HR algorithm allows for a greater number of downsampling multiples and faster detection results than other algorithms while preserving the original detection speed. In subsequent work, we plan to extend and apply the concept of modifying the network structure presented in this paper to other object detection algorithms, study other feature reuse strategies, and investigate the deployment and application issues of the algorithm presented in this paper in greater depth.

Author Contributions: D.W. and R.L. conceived and designed the experiments; D.W. and S.W. performed the experiments; T.X. and S.S. analyzed the data; X.L. contributed analysis tools; D.W. wrote the paper; R.L. supervised this work. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (NSFC) (Grant No. 51875164); National Key Research and Development Program of China (No. 2018YFB2003801).
Data Availability Statement: The datasets presented in this study are available through: https://github.com/hiians/simd (accessed on 12 August 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.