remote sensing

MDPI

Article

A Parameter-Free Pixel Correlation-Based Attention Module for Remote Sensing Object Detection

Xin Guan [1,2], Yifan Dong [1,2,*], Weixian Tan [1,2], Yun Su [1,2] and Pingping Huang [1,2]

[1] College of Information Engineering, Inner Mongolia University of Technology, Hohhot 010051, China; 20211100103@imut.edu.cn (X.G.); wxtan@imut.edu.cn (W.T.); suyun@imut.edu.cn (Y.S.); hwangpp@imut.edu.cn (P.H.)
[2] Inner Mongolia Key Laboratory of Radar Technology and Application, Hohhot 010051, China
* Correspondence: yfdong@imut.edu.cn

Abstract: Remote sensing image object detection is a challenging task in the field of computer vision due to the complex backgrounds and diverse arrangements of targets in remote sensing images, forming intricate scenes. To overcome this challenge, existing object detection models adopt rotated target detection methods. However, these methods often lead to a loss of semantic information during feature extraction, specifically regarding the insufficient consideration of element correlations. To solve this problem, this research introduces a novel attention module (EuPea) designed to effectively capture inter-elemental information in feature maps and generate more powerful feature maps for use in neural networks. In the EuPea attention mechanism, we integrate distance information and Pearson correlation coefficient information between elements in the feature map. Experimental results show that using either type of information individually can improve network performance, but their combination has a stronger effect, producing an attention-weighted feature map. This improvement effectively enhances the object detection performance of the model, enabling it to better comprehend information in remote sensing images. Concurrently, this also improves missed detections and false alarms in object detection. Experimental results obtained on the DOTA, NWPU VHR-10, and DIOR datasets indicate that, compared with baseline RCNN models, our approach achieves respective improvements of 1.0%, 2.4%, and 1.8% in mean average precision (mAP).

Keywords: remote sensing images; object detection; attention mechanism; deep learning

Citation: Guan, X.; Dong, Y.; Tan, W.; Su, Y.; Huang, P. A Parameter-Free Pixel Correlation-Based Attention Module for Remote Sensing Object Detection. *Remote Sens.* **2024**, *16*, 312. https://doi.org/10.3390/rs16020312

Academic Editor: Mohammad Awrangjeb

Received: 7 November 2023
Revised: 6 January 2024
Accepted: 8 January 2024
Published: 12 January 2024

1. Introduction

Remote sensing object detection is a crucial task in the field of computer vision, aimed at automatically identifying and locating objects of interest in aerial images. This task becomes particularly challenging when dealing with high-resolution, large-area-coverage, and multispectral remote sensing images. In recent years, with the continuous advancement of Convolutional Neural Networks (CNNs), there have been significant improvements in the performance of object detection. In the domain of aerial object detection, the Region-Based Convolutional Neural Network (RCNN) series of methods has achieved remarkable success [1].

However, as these methods have evolved, especially with the emergence of many two-stage object detection algorithms within the RCNN series, such as Fast RCNN [2] and Faster RCNN [3], there remain significant challenges in accurately localizing objects, particularly when dealing with oriented and densely distributed targets. Early oriented object proposals [4] adopted a rotational region proposal strategy. However, with the development of the RCNN series, particularly with the prominence of two-stage object detection algorithms like Fast RCNN and Faster RCNN, the challenge of accurate localization of oriented and densely distributed targets has become more prominent. Currently, in the field of remote sensing object detection, a prevalent strategy involves introducing

orientation regression methods based on classical object detectors. Noteworthy methods such as SCRDet [5], CADNet [6], DRN [7], R3 Det [8], ReDet [9], RoI Transformer[10], and Oriented RCNN [11] have achieved significant performance improvements by predicting the rotation angles of target bounding boxes.

Despite the significant advancements achieved by these methods in capturing surface-level information, they tend to overlook the potential semantic information correlations during the deep learning process. The performance of convolutional networks is significantly influenced by the network architecture. Therefore, constructing powerful convolutional network modules is crucial in improving performance. Modern convolutional networks often comprise multiple blocks, which include combinations of convolutional layers, pooling layers, activation functions, and other specific modules. Researchers have devoted efforts to designing advanced modules to enhance the performance of convolutional networks, such as Residual Networks [12] and Dense Connections [13]. However, these module designs typically require domain expertise and considerable tuning. To overcome this challenge, recent research has focused on building plug-and-play modules that allow the fine-tuning of convolutional outputs, enabling the network to learn rich semantic features. Among these, the Squeeze-and-Excitation (SE) module introduced by Hu et al. [14] has been widely adopted. It explicitly models the dependencies between convolutional feature channels, thus enhancing the network's representational capacity. Importantly, the SE module is flexible with regard to the network architecture and can be seamlessly embedded into various networks.

While SE modules have enabled significant advances in capturing channel information, they do not fully account for spatial information. The Convolutional Block Attention Module (CBAM) [15] extends this by introducing spatial information and adding max-pooling features to SE networks. However, complex models often come with an increased computational overhead. To address this while achieving substantial performance gains with a minimal parameter increase, Wang et al. proposed the Efficient Channel Attention (ECA) module [16]. Furthermore, traditional convolutional layers are limited in their ability to capture long-range semantic information, as they mainly focus on local details. Hou et al. introduced the Coordinate Attention (CA) module [17], which encodes both horizontal and vertical position information into channel attention. This allows the network to attend to a significant amount of position information without introducing excessive computational costs. On a different note, inspired by neuroscientific theories, Yang et al. constructed the Self-Importance Modulation Attention (SimAM) module [18], which is a parameterless attention module. Unlike existing channel or spatial attention modules, SimAM derives 3D attention weights directly, without the need for additional parameters. Additionally, Liu et al. proposed the Normalization-Based Attention Module (NAM) [19], which utilizes the scaling factor of batch normalization to implement a parameterless attention mechanism.

In addressing the ongoing environmental challenges associated with climate change, our proposed attention mechanism holds tremendous potential for extending its application scope to monitor variations in the high-mountain cryosphere, especially in regions characterized by diverse and complex backgrounds. It is noteworthy that the study by DH Shugar and colleagues, published in the journal *Science* in 2021 [20], underscores the increasing relevance of our work in light of the growing frequency of natural disasters linked to climate change.

However, conventional attention modules typically focus on weight relationships between channels, often overlooking the intrinsic pixel-wise correlations. Taking SimAM as an example, while it considers pixel-wise correlations, it falls short of fully harnessing the distance information between elements. This limitation adversely affects the comprehensive utilization of element-wise relationships within convolutional neural networks, ultimately compromising the accuracy of object detection. In response to these challenges, we introduce the Euclidean–Pearson (EuPea) attention module, specifically designed to address the issue of element-wise correlations and the influence of pixel distances on these relationships. Our main contributions are as follows:

- We propose a parameter-free, plug-and-play attention module for remote sensing image object detection that evaluates pixel correlations by computing Euclidean distances and Pearson coefficients between pixels.
- We integrate various attention modules into the Oriented RCNN model and conduct comparative experiments. The experimental results on the DOTA and NWPU VHR-10 datasets indicate that the proposed attention module achieves the highest mAP.

These innovations are expected to bring significant breakthroughs in the field of remote sensing object detection, enhancing accuracy while reducing computational costs.

2. Related Work

In this section, we briefly discuss representative works on network architectures and plug-and-play attention modules.

2.1. Network Architecture

The year 2012 saw a significant breakthrough in the field of deep learning in computer vision, with the introduction of AlexNet [21,22]. It represented a crucial milestone and found application in large-scale image classification tasks. Subsequently, researchers designed deeper convolutional neural networks, including VGG, DenseNet, and ReNet [12,13,23], further propelling the advancement of deep convolutional neural networks. Research on ResNet revealed that increasing the depth of neural networks can significantly enhance their representational capabilities, thereby allowing more complex visual tasks to be tackled. As the field of object detection gained prominence, representative two-stage object detection methods such as Fast RCNN and Faster RCNN [2,3] emerged. These methods introduced innovative techniques, such as Region Proposal Networks (RPNs), leading to substantial enhancements in object detection performance. However, traditional object detection algorithms faced challenges when handling rotated objects and similar unique cases involving object rotation and tilting. This challenge is particularly pronounced in remote sensing imagery, as these images are often captured under the influence of factors such as atmospheric conditions, terrain, and the Earth's curvature, resulting in objects appearing at varying rotation and tilt angles. To address this issue, researchers gradually developed rotated object detection algorithms. The Rotation Region Proposal Network [4,24,25] (RRPN) is the first model designed specifically for the detection of rotated objects. It builds upon Faster RCNN but introduces a rotation region proposal network, allowing the generation of inclined region proposals, which is crucial for objects in remote sensing images. Moreover, aside from RRPN, the research field has witnessed the emergence of several other rotated object detection networks, which have allowed significant progress in solving the rotated object detection problem. For example, Rotated RetinaNet-OBB/HBB is an improvement over RetinaNet [26] that incorporates information about the orientation angles of rotated objects to more accurately detect and localize them. RepPoints [27] is a point-based object detection method, and Rotated RepPoints-OBB extends this concept to adapt to rotated object detection by using points to represent the four corners of objects, enabling the effective detection of rotated objects. CSL (CoupleNet with Selective Loss) [28] is a coupled network integrated with selective loss, exhibiting excellent performance in rotated object detection. It can simultaneously detect horizontally and vertically oriented objects, making it particularly effective in scenes with complex layouts. R3Det [8] is a rotated object detection network based on a three-stage detection method, utilizing cascade detectors to enhance the object detection performance, especially in tasks requiring the highly precise detection of rotated objects. ReDet [9] is a deep learning model designed for the detection and localization of rotated objects in remote sensing imagery. It boasts strong versatility and can adapt to diverse object detection requirements across different scenarios. Oriented RCNN [11] (Rotated Region-Based Convolutional Neural Network) is an important approach built upon the RCNN series of algorithms. However, it focuses specifically on the detection and localization of rotated objects. Unlike traditional object detection methods, it takes into account the possibility of objects appearing in

rotated orientations, leading to the more accurate recognition and positioning of rotated objects. These networks have demonstrated outstanding performance in various application domains, offering a diverse array of choices addressing the challenge of rotated object detection. Their continuous development and improvement are of significant importance in handling complex scenarios, such as remote sensing imagery with rotated objects.

In this work, Oriented RCNN [11] (Oriented Region-Based Convolutional Neural Network) is adopted as the baseline model. It is a deep learning model designed for object detection in remote sensing images, effectively detecting and localizing rotated objects, including buildings, vehicles, and offset objects, in aerial data. The research in this field aims to address a broader and more complex set of challenges in remote sensing object detection.

2.2. Attention Mechanism Module

Chun et al.'s [29] research findings underscore the paramount importance of the human attention mechanism as a selection mechanism. The human brain exhibits a natural inclination to prioritize information that is pertinent to the current task while concurrently dampening signals that are extraneous to the task at hand. Drawing inspiration from this intrinsic human cognitive trait, a series of noteworthy attention mechanisms have been developed, with the Squeeze-and-Excitation (SE) mechanism standing out as the most prominent example. The SE attention mechanism achieves its efficacy by capturing contextual information at a global scale and learning to assign significance to various channels within a feature representation. In building upon the SE attention mechanism, numerous refinement techniques have been proposed. These include augmenting the spatial focus and extending the receptive field to further enhance model performance. The CBAM [15] module introduces spatial information and incorporates max-pooling feature extraction on the basis of SEnet; however, this leads to a significant computational overhead for complex models. In order to achieve notable performance improvement with a minimal increase in parameters, Wang et al. [16] proposed the ECA module. Due to the traditional convolutional layers being limited to extracting local information, Hou et al. [17] introduced the Coordinate Attention (CA) module, encoding horizontal and vertical positional information into channel attention. This allows the network to focus on a large amount of positional information without introducing excessive computational complexity. Nevertheless, these methods share a common constraint: they treat all neurons within the same channel or all neurons within the same spatial location equivalently. This uniform treatment fails to effectively capture the intricate three-dimensional weight relationships inherent in the data. To overcome this limitation and to better model the interplay between elements, Yang et al. [18] introduced the SimAM attention module. This module draws inspiration from select principles in prominent neuroscience theories. Notably, the SimAM attention module stands out as a parameter-free attention mechanism, which ensures that it does not introduce additional computational complexity. However, it is worth noting that the SimAM module exhibits a potential limitation in that it does not fully account for differences in the distances between elements. Such disparities in distance could potentially impact the accurate modeling of correlations. To address this concern, we have devised a function that incorporates distance-related factors. This innovative approach allows us to create a new attention module without the need to introduce extra trainable parameters. Meanwhile, for enhanced interpretability of attention mechanisms, we adopt a different approach from the CAM method proposed by B. Zhou et al. [30], which involves cumbersome modifications to the original model structure and retraining. Instead, we utilize the GradCAM algorithm improved by Selvaraju R R et al. [31] to provide heatmaps, ensuring the model's interpretability.

3. Method

3.1. Euclidean–Pearson Attention Mechanism Module

In the realm of deep learning, grasping the interplay between neurons holds paramount importance for model performance. Distance measurement methods serve as common

tools to gauge the similarity or correlation between data points. Within neural networks, assessing the correlation between distinct neurons is a necessity in order to enhance comprehension of the model's internal mechanisms. Consequently, this section introduces two distinct distance measurement methods aimed at assessing neuron-to-neuron correlation. Following this, we amalgamate these two approaches to craft a new attention mechanism called "Euclidean–Pearson Attention", or simply "EuPea Attention". This innovative attention mechanism thoroughly captures the extent of the correlation between neurons, exerting a profound influence on the performance of deep learning models.

3.1.1. Euclidean Distance

The Euclidean distance, often denoted as "d", is a widely employed method to measure the distance between two points. In the context of two points, P and Q, residing in an N-dimensional space, the Euclidean distance is mathematically represented by the following Equation (1):

$$d(P,Q) = \sqrt{\sum_{i=1}^{N}(P_i - Q_i)^2} \tag{1}$$

In this equation, $d(P,Q)$ signifies the Euclidean distance between point P and point Q, while N stands for the dimensionality of the space. The terms P_i and Q_i represent the respective coordinates of points P and Q along the i-th dimension. The essence of this formula lies in its step-by-step computation: it first calculates the square of the difference between corresponding coordinates in each dimension, and then aggregates these squared differences across all dimensions, and, finally, it computes the square root. This process yields the Euclidean distance between point P and point Q.

Based on the Euclidean distance formula, we can derive the Euclidean distance between each pixel and the channel-wise average value within a specific channel. Specifically, we first calculate the average value for each channel in the input feature map. Subsequently, we compute the Euclidean distance between each pixel and the average value within its respective channel. This relationship can be expressed using Equation (2):

$$D_{b,c,h,w} = \sqrt{\sum_{i=1}^{h}(x_{b,c,i,w} - \bar{x}_{b,c,1,1})^2} \tag{2}$$

Here, $D_{b,c,h,w}$ signifies the Euclidean distance for a specific pixel located at batch b, channel c, row h, and column w within the input tensor. It is calculated by comparing the pixel's value to the average value of all pixels within channel c of the input tensor at batch b.

3.1.2. Pearson Correlation

The Pearson correlation coefficient is a tool used to quantitatively measure the degree of correlation between two variables, with values ranging between -1 and 1. For two variables, X and Y, their Pearson correlation is mathematically represented by Equation (3). This correlation coefficient serves as a measure of the relationship between elements. As per Equation (4), in the energy function of SimAM, lower energy values imply greater dissimilarity between neuron t and its neighboring neurons, signifying higher importance. Consequently, the correlation between elements is defined by the following Equation (5):

$$\rho_{X,Y} = \frac{\sigma_X \sigma_Y}{\mathrm{cov}(X,Y)} \tag{3}$$

$$e_t^* = \frac{4(\hat{\sigma}^2 + \lambda)}{(t - \hat{\mu})^2 + 2\hat{\sigma}^2 + 2\lambda} \tag{4}$$

$$\rho_{b,c,h,w} = \frac{(x_{b,c,h,w} - \bar{x}_{b,c,1,1})^2}{4(\frac{\sum_{i=1}^{h}\sum_{j=1}^{w}(x_{b,c,i,j} - \bar{x}_{b,c,1,1})^2}{n} + \varepsilon)} + 0.5 \tag{5}$$

Here, $\rho_{b,c,h,w}$ represents the Pearson correlation coefficient of the pixel values in channel c at batch b, row h, and column w within the input tensor x. n denotes the total number of pixels, where $n = w \cdot h - 1$, indicating the count of pixels minus one. ε is a small constant, typically set to 0.001.

3.1.3. Information Integration

We can assess the correlation between neurons by comparing the outcomes of the Euclidean distance and the Pearson correlation coefficient. A lower Euclidean distance signifies greater dissimilarity between two pixels, whereas a higher Pearson correlation coefficient suggests a stronger correlation between them. This approach aids in identifying correlated pixels among neurons. Subsequently, we normalize the two sets of information mentioned above to facilitate their fusion, resulting in new information functions for both the distance, as expressed in Equation (6), and the Pearson correlation, as denoted in Equation (7). Normalizing Distance Information:

$$D_N = \frac{D - D_{min}}{D_{max} - D_{min}} \tag{6}$$

Normalizing Pearson Correlation Coefficient:

$$\rho_N = \frac{\rho - \rho_{min}}{\rho_{max} - \rho_{min}} \tag{7}$$

In these equations , D_N and ρ_N represent the normalized Euclidean distance and Pearson correlation coefficient, respectively. $D_{\min}$ and $P_{\min}$ denote the minimum values for the Euclidean distance and Pearson correlation coefficient, while $D_{\max}$ and $P_{\max}$ represent their maximum values. Finally, by combining the normalized distance information and the normalized Pearson correlation coefficient, a new information function I (Equation (8)) is derived. This novel information function integrates both distance and Pearson correlation information, effectively capturing the correlation between elements from these two aspects.

$$I = D_N + \rho_N \tag{8}$$

In our experiments, we compared the training of attention mechanisms that exclusively incorporate either inter-element distance information or Pearson correlation information. Encouragingly, we found that combining these two types of information significantly improved the accuracy of the results. Distance information reflects the spatial proximity or separation between elements, while Pearson correlation information signifies the numerical relationships between elements. The amalgamated information, I, can be regarded as a comprehensive feature representation in which each element reflects its degree of association with others. This integrated information provides a more holistic insight into the inter-relationships between elements. In Equation (8), I integrates information across both channel and spatial dimensions. A sigmoid function is applied to confine excessively large values within I. The primary purpose of employing the sigmoid function is to transform the amalgamated information I into attention weights, which are subsequently used to modulate the input features.

In conclusion, after scaling and fine-tuning the fused information I, we derive the feature weights as described in Equation (9). This process involves computing attention weights based on the pixel values within each channel of the input tensor x. Subsequently, these attention weights are applied to the input features, leading to outputs that have been adjusted with attention. This crucial step helps the model to prioritize important information within the input features, ultimately contributing to an enhancement in the model's performance.

$$\widetilde{x} = sigmoid(I) \odot x \tag{9}$$

The network architecture of the Euclidean–Pearson attention module is illustrated in Figure 1. Its core purpose is to calculate attention weights for features by amalgamating both distance and correlation information. This empowers the model to gain a deeper understanding of the input data, resulting in more precise and focused outputs.

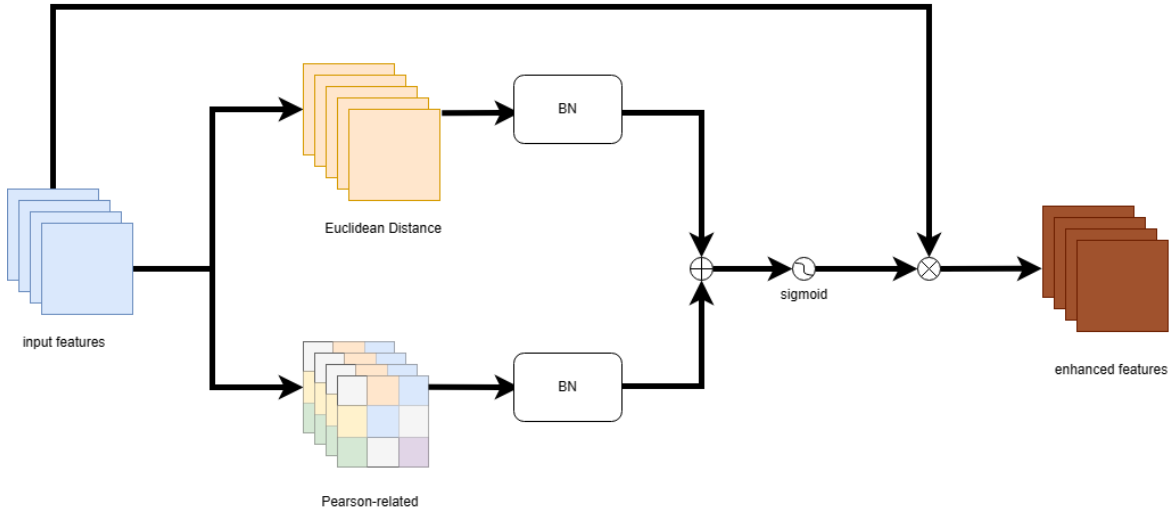


Figure 1. Euclidean–Pearson Attention Module.

3.2. Overall Architecture

We chose Oriented RCNN as the framework and baseline model. In the current stage of oriented object detection algorithms, Oriented RCNN is capable of generating high-quality anchor boxes, and its detection speed is relatively fast compared to other detection algorithms. The network architecture is shown in Figure 2.

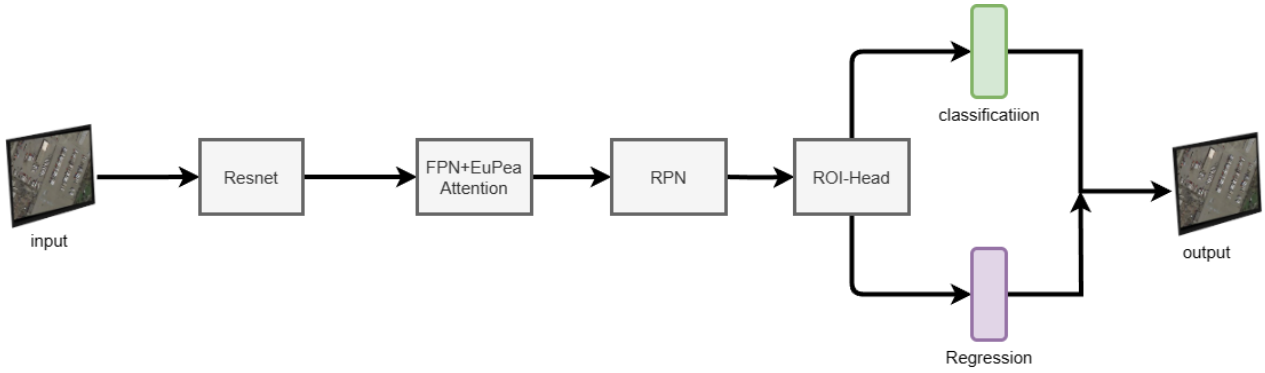


Figure 2. Overall architecture.

3.2.1. Feature Extraction Module

The feature extraction module utilizes Resnet-50 [12] and FPN [32] for the extraction of multi-scale features, facilitating the detection of objects of different sizes. Our module is placed after the FPN output, and we apply weight adjustments to the resulting features (see Figure 3).

In the bottom-up phase, the input image undergoes a series of convolutional layers, yielding low-level semantic feature maps and high-level semantic feature maps (C1, C2, C3, C4, C5). Moving on to the top-down phase, more robust features are obtained through feature fusion. Specifically, M4 is derived from the 1×1 convolutional output of C5, while M5 results from upsampling and the addition of the 1×1 convolutional output of C4. Following this progression, M3 and M2 are generated to effectively retain both high-level and low-level semantic information. P2, P3, P4, and P5 are achieved by applying 3×3 convolutions to M2, M3, M4, and M5, respectively. Additionally, P6 is generated

by applying max pooling to P5. Ultimately, the fused multi-scale features produced by this module serve as input to the attention module for the purpose of adjusting the feature weights.

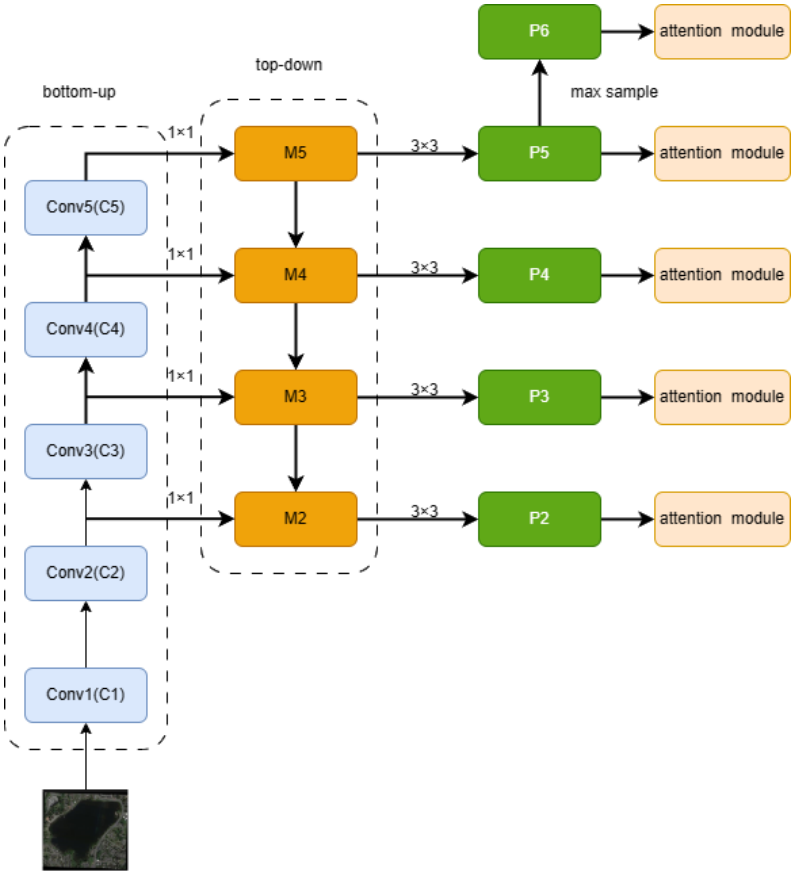


Figure 3. FPN structure diagram.

3.2.2. Oriented RPN

The Oriented Region Proposal Network (RPN) serves as a crucial component in the detection and localization of rotated objects. It leverages position information and objectness scores to generate region proposals, accommodating the presence of rotated objects. The RPN takes adjusted features from P2, P3, P4, P5, and P6 as the input. The structure of the RPN is depicted in Figure 4.

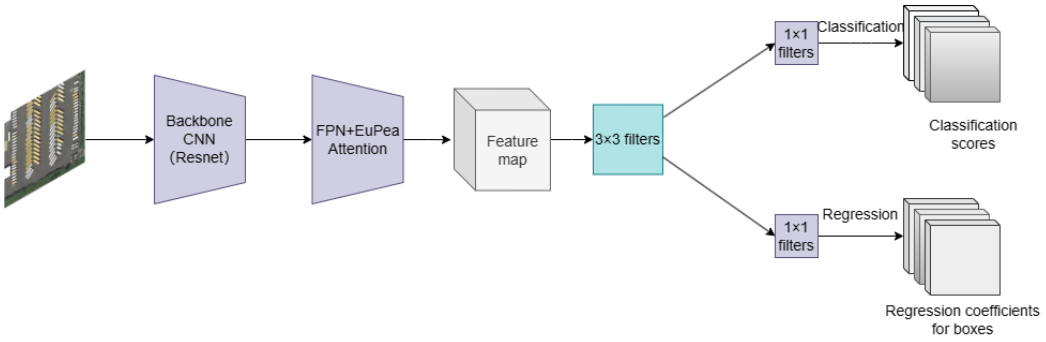


Figure 4. Region Proposal Network architecture consists of a 3 × 3 convolutional layer and two 1 × 1 convolutional layers, employed for classification and regression purposes.

For each feature layer and scale, we set three different aspect ratios (1:2, 1:1, and 2:1) for horizontal anchors. These anchors are distributed across five different feature layers (P2, P3, P4, P5, and P6). The areas of the anchors on each feature layer are (32^2, 64^2, 128^2, 256^2, 512^2),

respectively. Each anchor can be represented by its center coordinates and dimensions, using a four-dimensional vector $a = (ax, ay, aw, ah)$. Here, ax and ay denote the horizontal and vertical positions of the anchor's center, while aw and ah represent its width and height. The regression branch (bottom branch) outputs six offsets $\delta = (\delta_x, \delta_y, \delta_w, \delta_h, \Delta\alpha, \Delta\beta)$, making a total of 18 dimensions. We decode these offsets using the following formula to obtain the coordinates and angle information for oriented proposals $(x, y, w, h, \Delta\alpha, \Delta\beta)$, as shown in Equation (10), where (x, y) denotes the center point coordinates of the proposal box and w and h represent the width and height of the outer bounding box of the proposal. $\Delta\alpha$ and $\Delta\beta$ represent the vertex offsets of the proposal box. The top branch is the classification branch, which outputs scores related to the target's properties.

$$\begin{cases} \Delta\alpha = \delta_a \cdot w, \Delta\beta = \delta_\beta \cdot h \\ w = a_w \cdot e^{\delta_w}, h = a_h \cdot e^{\delta_h} \\ x = \delta_x \cdot a_w + a_x, y = \delta_y \cdot a_h + a_y \end{cases} \tag{10}$$

To represent anchor boxes in an oriented fashion, we utilize a novel approach referred to as "center point offset", as depicted in Figure 5. The black dot signifies the center point of the horizontal bounding box, while the oriented bounding box, denoted as o, encompasses this bounding box. The orange dots denote the four vertices of the oriented bounding box. Specifically, we employ six parameters to describe the oriented bounding box $(x, y, w, h, \Delta\alpha, \Delta\beta)$. These six parameters enable us to compute the coordinates of the four vertices of each proposal, designated as $v = (v_1, v_2, v_3, v_4)$. In this context, $\Delta\alpha$ represents the offset of v_1 relative to the upper midpoint $(x, y - \frac{h}{2})$ of the horizontal box. Due to symmetry, $-\Delta\alpha$ signifies the offset of v_3 relative to the lower midpoint $(x, y + \frac{h}{2})$. $\Delta\beta$ denotes the offset of v_2 relative to the right-side midpoint $(x + \frac{w}{2}, y)$, and $-\Delta\beta$ represents the offset of v_4 relative to the left-side midpoint $(x - \frac{w}{2}, y)$. As a result, the coordinates of the four vertices can be expressed as follows:

$$\begin{cases} v_1 = (x, y - \frac{h}{2}) + (\Delta\alpha, 0) \\ v_2 = (x + \frac{w}{2}, y) + (0, \Delta\beta) \\ v_3 = (x, y + \frac{h}{2}) + (-\Delta\alpha, 0) \\ v_4 = (x - \frac{w}{2}, y + (0, -\Delta\beta) \end{cases} \tag{11}$$

By employing this representation, we achieve the regression of each oriented proposal by predicting its external rectangle parameters (x, y, w, h) and inferring the center-point offset parameters $(\Delta\alpha, \Delta\beta)$.

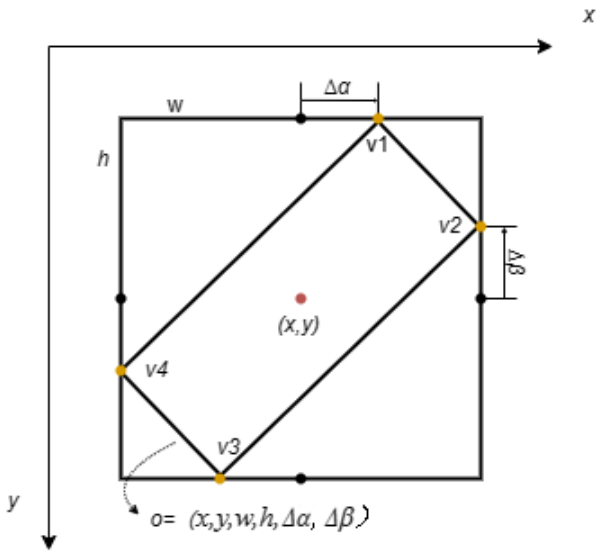


Figure 5. The representation of center-point offsets.

During the training process, we employ the following rules for positive and negative sample assignment:

1. Anchor boxes exhibiting an Intersection over Union (IoU) greater than 0.7 with any ground-truth box are categorized as positive samples.
2. The anchor box with the highest IoU overlap with a ground-truth box, provided that its IoU surpasses 0.3, is selected.
3. Anchor boxes with an IoU less than 0.3 are designated as negative samples.
4. Anchor boxes failing to meet the aforementioned criteria are excluded from the training process.

3.2.3. Region-of-Interest Detection Module

The baseline introduces the Rotated RoIAlign (Rotated Region of Interest Align) operation to handle RoIs in the context of rotated object detection. Unlike the conventional RoIAlign method [33], Rotated RoIAlign is specifically tailored to rotated objects, enabling the more precise alignment of features within the regions of interest. This operation begins by rotating the RoI to the horizontal orientation based on the target's rotation angle. It then partitions the RoI into fixed-size grids and interpolates the features within each grid to produce an aligned RoI feature representation. This alignment approach takes into full consideration the target's shape and rotation, thus mitigating information loss and enhancing the detection performance.

Next, we will describe the Rotated RoIAlign process based on Figure 6. In the context of Rotated RoIAlign, proposals generated by the oriented RPN are typically represented as parallelograms. To simplify the computational processing, these parallelograms are adjusted to oriented rectangles by elongating the shorter diagonal to match the length of the longer diagonal. Consequently, we obtain oriented rectangles characterized by parameters including center coordinates (x, y), width and height (w, h), and a rotation angle θ. The rotation angle θ typically falls within the range of $[-\frac{\pi}{2}, \frac{\pi}{2}]$, indicating the degree of rotation relative to the horizontal axis.

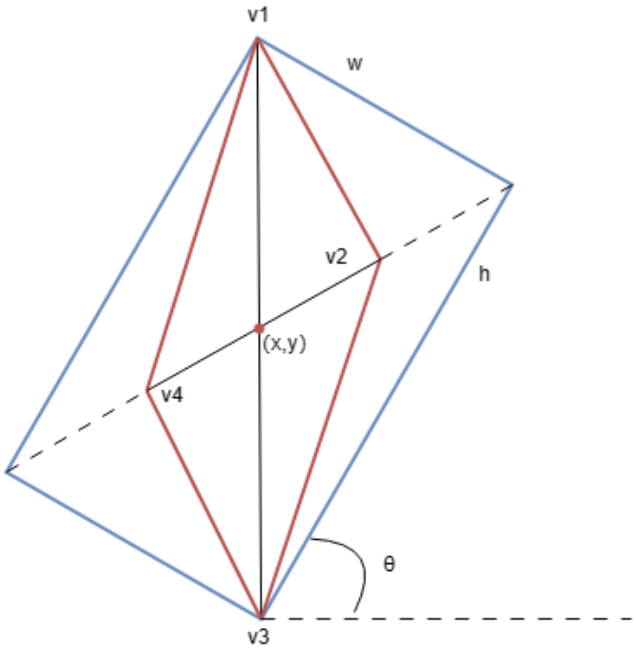


Figure 6. Rotated Region of Interest Align.

Through the joint optimization of the Rotated Region of Interest Align (Rotated ROIAlign) operation with the oriented RCNN head, we have achieved end-to-end training. During the inference stage, we retained 2000 proposals from each FPN level and mitigated

redundancy using non-maximum suppression (NMS). We applied horizontal NMS with an IoU threshold of 0.8, merged the remaining proposals, and selected the top 1000 proposals based on their classification scores as inputs for the second stage. In this second stage, we utilized strategy-based NMS to further refine the predicted oriented bounding boxes, thereby enhancing the overall performance.

3.3. Loss Function

To train the Oriented RPN and Oriented RCNN head, we introduce the Cross-Entropy Loss L_{cls} for the classification task and the Smooth L1 Loss L_{reg} for the regression task. The overall loss function L is defined as follows:

$$L(p_i, t_i) = \frac{1}{N}\sum_i L_{cls}(p_i, p_i^*) + \frac{1}{N}\sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{12}$$

$$L_{cls}(p_i, p_i^*) = -[p_i^* \log(p_i) + (1 - p_i^*)\log(1 - p_i)] \tag{13}$$

$$L_{reg}(t_i, t_i^*) = \begin{cases} 0.5(t_i - t_i^*)^2 & \text{if } |t_i - t_i^*| < 1 \\ |t_i - t_i^*| - 0.5 & \text{otherwise} \end{cases} \tag{14}$$

where N represents the total number of predicted anchors in an image, and i is the index of each predicted anchor. p_i stands for the probability associated with the predicted anchors, while p_i^* corresponds to the ground-truth label, which can take values of 0 (indicating a negative anchor) or 1 (indicating a positive anchor). Furthermore, t_i and t_i^* refer to the predicted bounding box and the ground-truth bounding box, respectively.

4. Experiments

In this section, we will introduce the datasets, evaluation metrics, and experimental details used in this study.

4.1. Datasets

To evaluate the proposed method, we conducted experiments on three publicly available aerial image datasets: DOTA, NWPU VHR-10, and DIOR.

The DOTA dataset [34] is employed for object detection in aerial images, consisting of 2806 aerial images that cover 15 common object categories. These categories include bridges, ports, ships, airplanes, helicopters, small vehicles, large vehicles, baseball fields, ground tracks, tennis courts, basketball courts, soccer fields, roundabouts, swimming pools, and storage tanks. In our research, we used the training set for model training and the validation set to assess the performance.

The NWPU VHR-10 dataset comprises a total of 800 ultra-high-resolution (VHR) remote sensing images, encompassing 10 common recognizable object categories. These ten object classes include airplanes, ships, storage tanks, baseball fields, tennis courts, basketball courts, ground runways, ports, bridges, and vehicles. We utilized 600 of these images for training and 200 for validation in our experiments.

DIOR is a large-scale benchmark dataset designed for optical remote sensing image object detection. The dataset comprises 23,463 images with 192,472 instances, covering 20 object classes. These 20 object classes include airplanes, airports, baseball fields, basketball courts, bridges, chimneys, dams, expressway service areas, expressway toll stations, harbors, golf courses, ground track fields, overpasses, ships, stadiums, storage tanks, tennis courts, train stations, vehicles, and windmills.

4.2. Evaluation Metrics

To assess the performance of the proposed method, we employed four common evaluation metrics, precision, recall, average precision, and mean average precision, as described in 'The Elements of Statistical Learning' by Hastie et al. [35] and the evaluation method proposed by Muhuri et al. [36]. Their formulas are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

In this context, TN represents the number of true negatives, FN represents the number of false negatives, and FP represents the number of false positives. Precision measures the proportion of correctly detected positives among all positive detections, while recall measures the proportion of correctly detected positives among all positive samples.

Average precision (AP) is calculated by computing the average precision across the range of recall values from 0 to 1.

$$AP = \int_0^1 P(R)dR \tag{17}$$

Mean average precision (mAP) is used to describe the performance of multi-class object detection and is calculated as follows:

$$mAP = \frac{1}{N_{\text{class}}} \sum_{j=1}^{N_{class}} \int_0^1 P_j(R_j)dR_j \tag{18}$$

where Nclass is the number of classes in the dataset, j represents the index of a particular class, and Pj and Rj are the precision and recall for class j.

4.3. Implementation Details

"Our experiments were conducted using the mmrotate framework [37]. The implementation was carried out in Python 3.8.10, utilizing Torch 1.10.0+cu113. Initially, we segmented the original DOTA images, and subsequently, the training images were resized to 1024 × 1024 pixels. The experiments were performed on a single GPU, specifically the NVIDIA RTX 3090, running Ubuntu 18.04 as the operating system. We employed Stochastic Gradient Descent (SGD) as the optimizer, with a learning rate of 0.005, momentum of 0.9, and weight decay of 0.0001. The batch size was set to 1, and we carried out a total of 12 training epochs."

5. Results

This section commences with a thorough comparison of the attention mechanisms utilized in the model at hand and those employed in other models. Following this, the experimental results of the model on the DOTA dataset, NWPU VHR-10 dataset and DIOR datasetwill be visualized and comprehensively compared.

5.1. Results on the DOTA Dataset

We integrated our attention mechanism into the baseline model, resulting in a notable, 1.0%, increase in mAP. The comparative results in Table 1 illustrate significant improvements in the detection rates for specific categories, such as "airplane" (+3.9%), "baseball field" (+2.1%), and "helicopter" (+11.3%), when compared to the baseline model.

To assess the enhancement brought by EuPea in the model performance, we conducted visual experiments by analyzing various features, including those processed through the FPN (Feature Pyramid Network). As depicted in Figure 7, our EuPea attention model significantly intensifies the focus on primary objects. These results imply that our model can effectively augment the representative power of targets within the network, all without the need to introduce any additional parameters.

Through the comparison in Figure 7, it is evident that the model, after incorporating the Euclidean–Pearson attention mechanism, exhibits a significantly improved focus on targets compared to the baseline model. While the SE attention mechanism can boost the attention on detected objects, its performance in detecting small targets is not very pronounced. In contrast, our EuPea attention mechanism noticeably increases the attention on various target categories

and significantly improves the detection performance for small targets. It is worth noting that all these improvements are achieved without introducing extra parameters.

Table 1. Comparing the performance of our proposed method and the baseline model on the DOTA dataset.

METHOD	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
R3Det [8]	88.8	67.4	44.1	69.0	62.9	71.7	78.7	89.9	47.3	61.2	47.4	59.3	59.2	51.7	24.3	61.5
CSL [38]	88.1	72.2	39.8	63.8	64.3	71.9	78.5	89.6	52.4	61.0	50.5	66.0	56.6	50.1	27.5	62.2
RoI Trans [10]	**89.9**	76.5	48.1	73.1	68.7	78.2	88.7	**90.8**	73.6	62.7	62.0	63.4	73.7	57.2	47.9	70.3
Oriented rcnn [11]	85.9	75.5	**51.8**	78.0	69.1	84.0	89.3	90.7	73.9	62.6	62.8	66.4	75.4	56.9	44.0	71.3
Oriented rcnn + SE [14]	89.7	**80.6**	50.8	74.1	67.9	84.2	89.2	**90.8**	72.3	62.7	61.0	65.5	**75.5**	57.6	50.5	71.5
Oriented rcnn + CBAM [15]	89.8	75.4	50.9	70.8	68.7	84.4	89.1	90.6	73.8	62.7	59.3	**66.9**	75.1	**59.0**	49.6	71.6
Oriented rcnn + SimAM [18]	89.8	75.4	50.9	**78.8**	68.7	84.4	89.1	90.6	73.8	62.7	59.3	**66.9**	75.1	**59.0**	49.6	71.6
Oriented rcnn + Eu	89.8	76.5	51.5	75.2	67.7	84.5	89.2	90.7	74.0	62.5	61.6	65.0	**75.5**	57.6	53.1	71.6
Oriented rcnn + Pea	89.4	75.0	50.0	76.8	68.8	84.5	**89.4**	90.7	73.0	**62.8**	61.5	66.3	75.3	57.5	53.7	71.7
Oriented rcnn + EuPea	89.6	77.6	51.3	76.6	**69.3**	**85.3**	89.3	90.7	**74.2**	62.6	**63.1**	66.4	75.4	58.4	**55.3**	**72.3**

The bold text in the table represents the optimal values for each category detection. The baseline model is Oriented RCNN [15]. "Eu" represents the results obtained when only the Euclidean distance is incorporated, and "Pea" represents the results obtained when only the Pearson correlation coefficient is incorporated. The following are the abbreviations for different categories: PL—planes; BD—baseball fields; BR—bridges; GTF—ground runways; SV—small vehicles; LV—large vehicles; SH—ships; TC—tennis courts; BC—basketball courts; ST—storage tanks; SBF—soccer fields; RA—roundabouts; HA—harbors; SP—swimming pools; HC—helicopters.

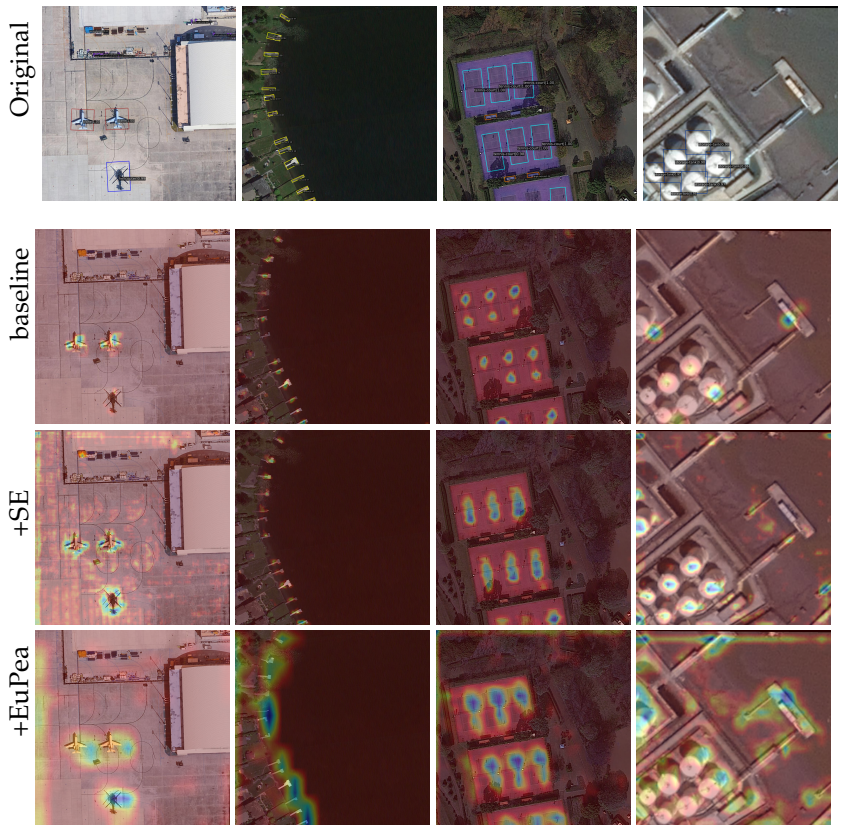


Figure 7. Visualizations of feature activation introduced through different networks.

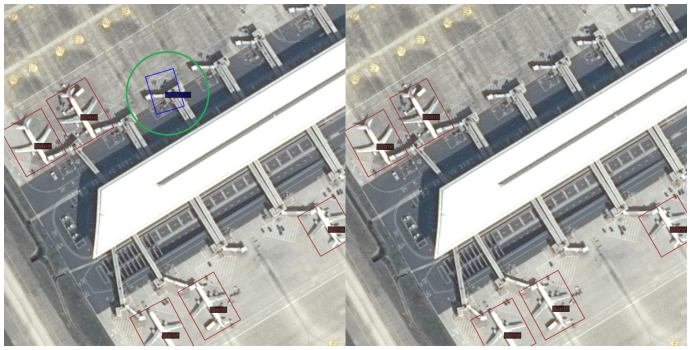
Furthermore, there has been a notable improvement in mitigating both missed detections (false negatives) and false alarms (false positives) to a certain extent. Figures 8 and 9, respectively, illustrate the reduction in missed detections and false alarms using Euclidean distance attention and Pearson attention. In each subfigure, the left section showcases the detection outcomes of the baseline model, while the right section displays the detection results produced by our proposed method. Missed detections are denoted in red, while false alarms are indicated in green. The positive results generated by the Euclidean–Pearson attention mechanism can be seen in Figure 10.

(**a**)

(**b**)

Figure 8. The impact of incorporating Euclidean distance attention into the model: (**a**) missed detection of small vehicles; (**b**) false detection of basketball court.

(**a**)



(**b**)

Figure 9. The impact of incorporating Pearson attention into the model: (**a**) missed detection of baseball diamond; (**b**) false detection of aircraft.

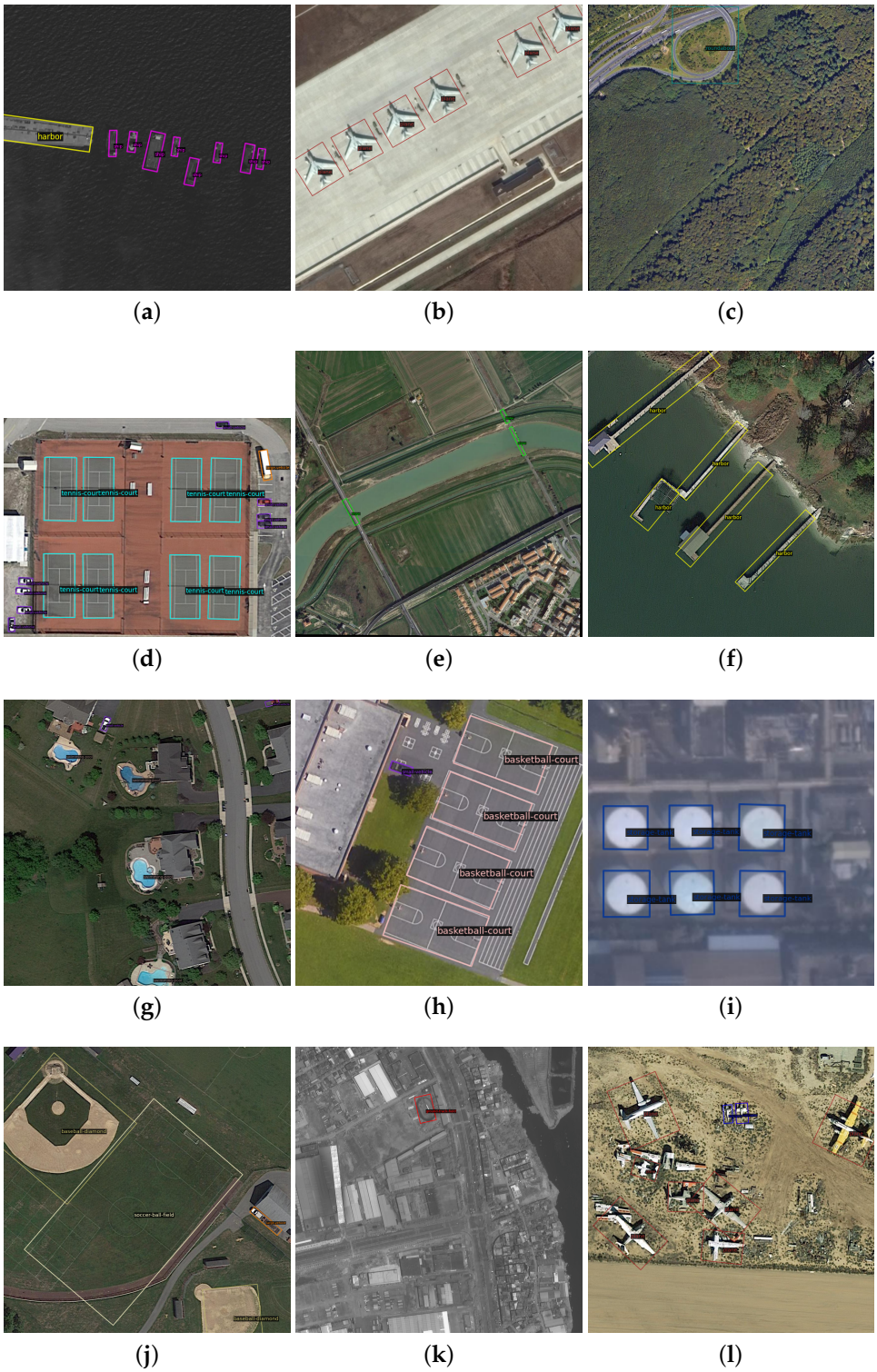


Figure 10. Visualization of the positive results: (**a**) ship and harbor; (**b**) plane; (**c**) roundabout; (**d**) small vehicle, tennis court, and large vehicle; (**e**) bridge; (**f**) harbor; (**g**) swimming pool; (**h**) basketball court; (**i**) storage tank; (**j**) baseball diamond and soccer field; (**k**) ground track field; (**l**) helicopter and plane.

The Euclidean–Pearson attention mechanism excels at weight allocation in the spatial domain, making efficient use of spatial semantic information. Moreover, due to its parameter-free nature, this attention mechanism achieves higher detection accuracy while maintaining the training speed. In terms of inference speed, we conducted a comparison between our model and other attention mechanism models, as illustrated in Table 2.

These experiments were conducted on the DOTA dataset, with input data shapes set to 1024 × 1024.

Table 2. A comparison of model complexity and training speed with other models.

METHOD	FPS	FLOPs	Parameters	mAP
Baseline	15.8	211.43G	41.14M	71.3
Baseline + SE	15.2	211.46G	41.84M	71.5
Baseline + CBAM	15.0	211.53G	42.08M	71.6
Baseline + SimAM	15.4	211.43G	41.14M	71.6
Baseline + EuPea	15.1	211.43G	41.14M	72.3

5.2. Results on the NWPU VHR-10 Dataset

The experiments conducted on the NWPU VHR-10 dataset yielded notable improvements in performance. The AP values and mAP values for various categories are presented in Table 3. It is evident that our approach enhanced the baseline model's mAP by 2.4%. The incorporation of distance information contributed to a 1.4% improvement in detection performance. Additionally, the utilization of Pearson correlation information led to significant improvements in AP, with an 8% increase for storage tanks, a 7.7% increase for basketball courts, an 8.6% increase for harbors, and a 3.9% increase for vehicles.

Table 3. Comparing the baseline model, Oriented RCNN, with our proposed model on the NWPU VHR-10 dataset.

METHOD	AP	SH	ST	BD	TC	BC	GTF	HA	BR	VE	mAP
R3Det [8]	97.8	87.5	80.8	88.6	71.2	80.1	85.2	72.3	84.2	82.1	82.9
CSL [38]	97.8	88.2	80.5	88.7	71.3	81.0	86.1	71.9	86.1	84.6	83.6
RoI Trans [10]	99.9	90.1	81.0	**90.7**	72.4	81.8	90.9	78.0	88.6	89.3	86.3
Oriented rcnn [11]	99.9	90.4	81.4	90.6	72.3	81.5	98.0	76.8	**90.9**	85.1	86.7
Oriented rcnn + SE [14]	99.9	**90.8**	81.0	**90.7**	**72.7**	81.3	98.9	78.4	89.1	89.0	87.2
Oriented rcnn + CBAM [15]	99.9	90.4	80.7	**90.7**	**72.7**	88.7	**100**	76.1	89.7	**89.7**	87.9
Oriented rcnn + SimAM [18]	99.9	90.4	80.9	**90.7**	**72.7**	**89.3**	99.8	77.9	89.8	89.6	88.1
Oriented rcnn + Eu	**100**	90.7	80.6	90.6	**72.7**	81.5	99.7	86.2	88.2	89.1	87.9
Oriented rcnn + Pea	99.9	90.7	80.9	**90.7**	**72.7**	87.2	**100**	**86.3**	87.8	88.5	88.5
Oriented rcnn + EuPea	99.9	90.2	**89.4**	**90.7**	**72.7**	89.2	**100**	85.4	84.4	89.0	**89.1**

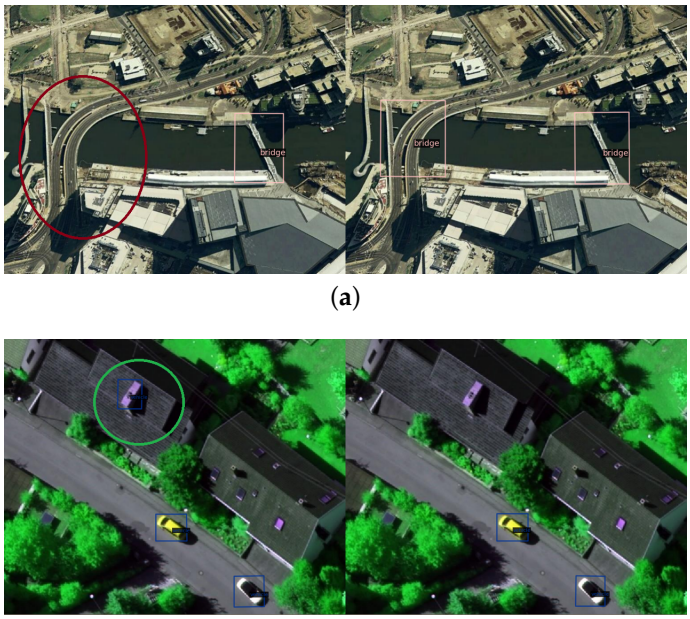
The bold text in the table represents the optimal values for each category detection. The baseline model is the Oriented RCNN model. "Eu" represents the results obtained when only Euclidean distance information is incorporated, while "Pea" represents the results obtained when only Pearson correlation information is incorporated. AP: airplane; SH: ship; ST: storage tank; BD: baseball diamond; TC: tennis court; BC: basketball court; GTF: ground track field; HB: harbor; BD: bridge; VE: vehicle.

For the NWPU VHR-10 dataset, we visualized the impact of distance information and Pearson correlation information on the baseline model separately, as shown in Figures 11 and 12. The left side represents the baseline model, while the right side represents the improved model. We use red circles to denote missed detections and green circles to denote false alarms. Positive detection results are illustrated in Figure 13.

We compared our model with other models on the NWPU VHR-10 dataset in terms of model complexity and inference speed, as shown in Table 4. It can be seen that while it improved the detection accuracy, the EuPea module did not reduce the inference speed.

Table 4. Comparison of model performance on the NWPU VHR-10 dataset.

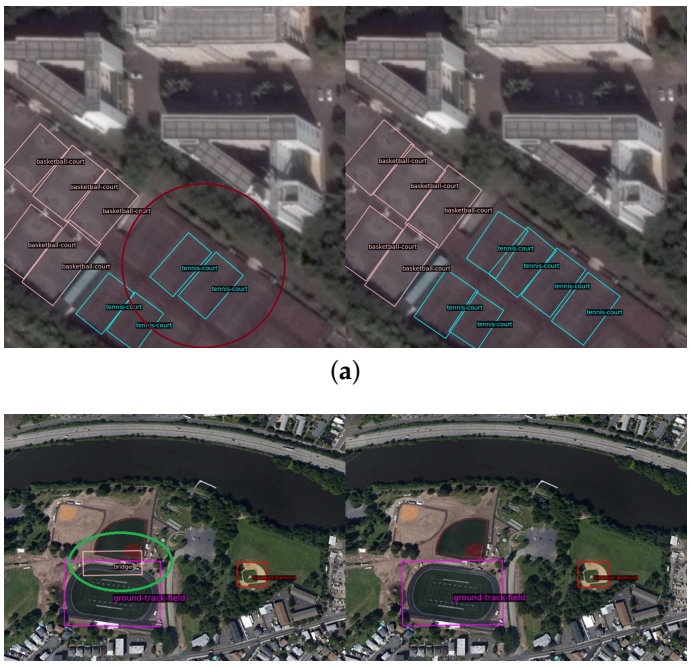
METHOD	FPS	FLOPs	Parameters	mAP
Baseline	21.5	211.43G	41.13M	86.7
Baseline + SE	21.4	211.46G	41.83M	87.2
Baseline + CBAM	21.3	211.53G	42.10M	87.9
Baseline + SIM	21.4	211.43G	41.13M	88.1
Baseline + EuPea	21.4	211.43G	41.14M	89.1



(**a**)

(**b**)

Figure 11. The impact of applying the Euclidean distance attention model to the baseline model: (**a**) missed detection of bridges; (**b**) false alarm of vehicle.



(**a**)

(**b**)

Figure 12. The impact of applying the Pearson correlation mechanism model to the baseline: (**a**) missed detection of bridges; (**b**) false detection of ground runways.

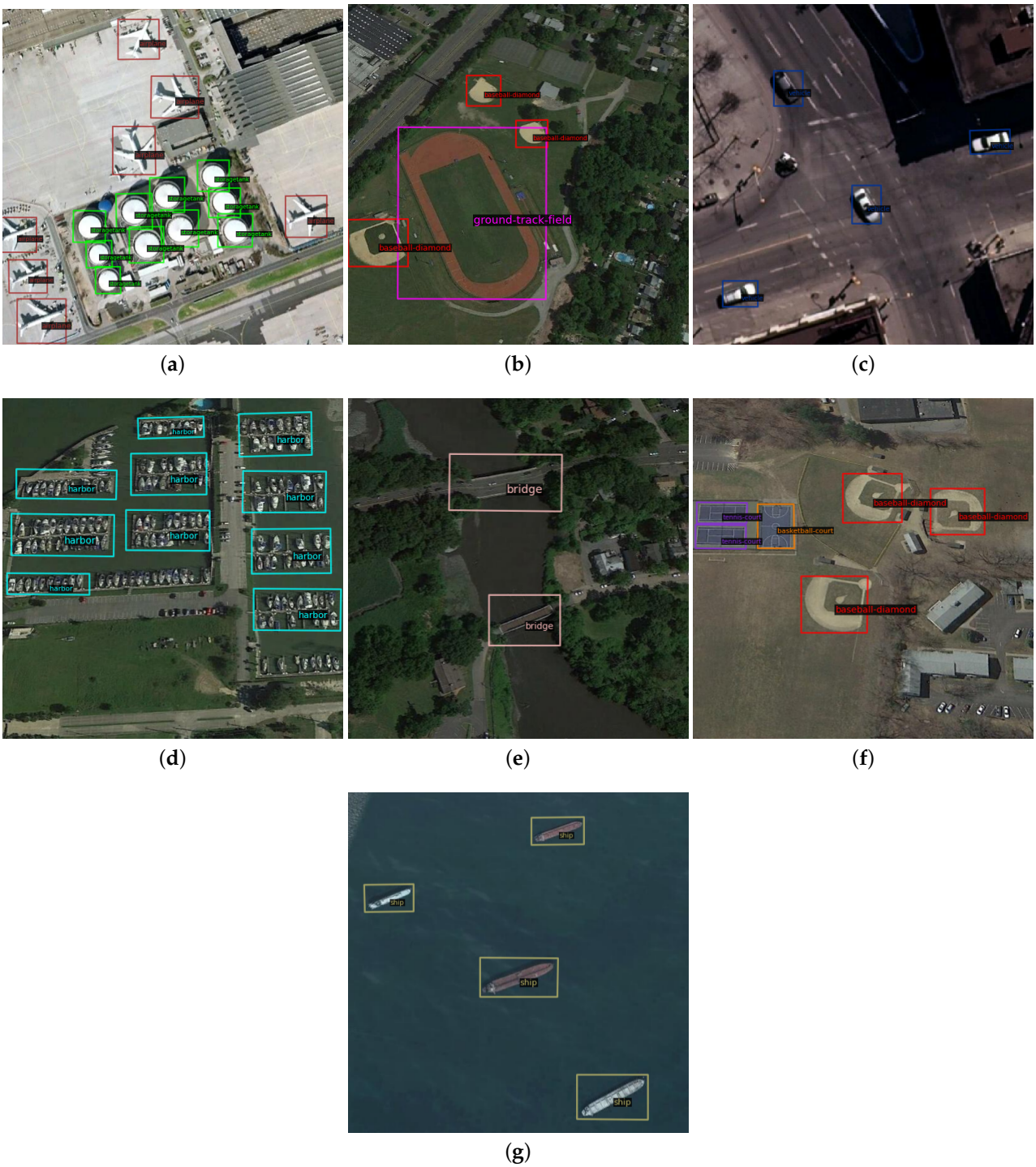


Figure 13. Visualization of the positive results: (**a**) airplane and storage tank; (**b**) baseball diamond and ground track field; (**c**) vehicle; (**d**) harbor; (**e**) bridge; (**f**) baseball diamond, basketball court, and tennis court; (**g**) ship.

5.3. Results on the DIOR Dataset

The experiments conducted on the DIOR dataset demonstrate an improvement in performance. Specifically, the AP values and mAP values for each category are shown in Table 5. Our Euclidean–Pearson attention module enhances the mAP of the baseline model by 1.8%. The distance information module performs well, leading to a 0.4% improvement in detection performance. Furthermore, the Pearson module further boosts detection performance, contributing an additional 0.9% to the mAP. It is worth noting that our

approach not only achieves improvements in overall performance but also effectively eliminates missed detections and wrong detections.

Table 5. Comparing the baseline model, Oriented RCNN, with our proposed model on the DIOR dataset.

METHOD	APL	APT	BF	BC	BR	CM	DA	ESA	EST	GF
R3Det [8]	89.6	6.40	89.5	71.2	14.4	81.7	8.90	26.5	48.1	31.8
CSL [38]	**90.9**	2.60	89.4	71.5	7.10	**81.8**	9.30	31.4	41.5	58.1
RoI Trans [10]	90.8	12.1	90.8	79.8	22.9	**81.8**	8.20	51.3	54.1	**60.7**
Baseline [11]	**90.9**	18.3	**90.8**	80.7	35.0	**81.8**	15.4	59.8	52.9	55.3
Baseline + SE [14]	**90.9**	19.5	**90.8**	80.4	33.8	**81.8**	15.3	59.7	52.6	57.9
Baseline + SimAM [18]	**90.9**	20.0	90.7	80.4	34.2	**81.8**	17.4	60.0	**53.7**	55.4
Baseline + Eu	**90.9**	20.2	**90.8**	80.1	35.4	**81.8**	17.6	59.7	**53.7**	58.9
Baseline + Pea	**90.9**	18.6	90.7	80.3	34.3	**81.8**	19.3	58.6	53.5	57.0
Baseline + EuPea	**90.9**	**21.6**	**90.8**	**80.9**	**37.0**	**81.8**	**20.5**	**62.0**	**53.7**	59.7
GTF	**HA**	**OPS**	**SP**	**STD**	**ST**	**TC**	**TS**	**VEH**	**WD**	**mAP**
65.6	8.20	33.4	69.2	51.9	72.9	81.1	21.4	54.2	44.7	48.5
63.2	17.5	26.6	69.3	53.8	72.8	81.6	18.4	47.2	46.3	49.0
75.8	30.6	40.5	89.9	88.2	79.5	81.8	20.6	67.9	55.1	59.1
76.3	21.8	53.2	89.8	85.6	79.3	81.8	36.3	68.8	56.2	61.5
76.4	24.6	56.1	89.7	**89.0**	79.7	81.8	29.6	68.4	56.0	61.7
76.5	23.5	54.4	89.8	88.5	80.0	81.8	40.4	68.7	55.8	62.2
76.2	28.3	56.1	89.9	86.3	**79.8**	81.8	29.7	**69.0**	55.6	61.9
76.7	25.8	55.1	89.7	88.7	**79.8**	81.8	41.3	68.7	55.8	62.4
78.9	**31.6**	**56.6**	**90.0**	88.3	79.7	**85.4**	**42.6**	59.2	**56.5**	**63.3**

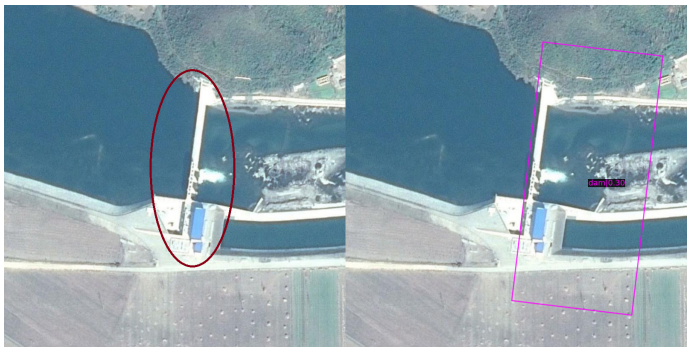
The bold text in the table represents the optimal values for each category detection. The baseline is the Oriented RCNN. Eu stands for stand-alone European module, Pea stands for stand-alone Pearson module. APL: airplane; APT: airport; BF: baseball field; BC: basketball court; BR: bridge; CM: chimney; DA: dam; ESA: expressway service area; ETS: expressway toll station; GF: golf field; GTF: ground track field; HA: harbor; OPS: overpass; SP: ship; STM: stadium; ST: storage tank; TC: tennis court; TS: train station; VEH: vehicle; WD: windmill.

For the DIOR dataset, we visualized the impact of the two modules on the baseline model. As illustrated in Figures 14 and 15, the left side represents the baseline model, while the right side represents the model with the improved modules. In the figures, red circles denote missed detections, and green circles denote false detections. Positive detection results are illustrated in Figure 16.

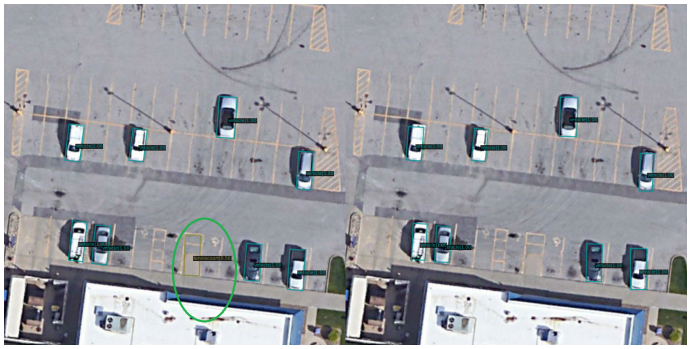
On the DIOR dataset, we conducted a comparison of the inference speeds of our model. As shown in Table 6, our model maintains its inference speed without an increase in parameters.

Table 6. Comparison of model performance on the DIOR dataset.

METHOD	FPS	FLOPs	Parameters	mAP
Baseline	20.9	211.44G	41.14M	61.5
Baseline + SE	19.4	211.46G	41.79M	61.7
Baseline + SIM	20.4	211.43G	41.13M	62.2
Baseline + EuPea	20.3	211.43G	41.14M	63.3



(**a**)



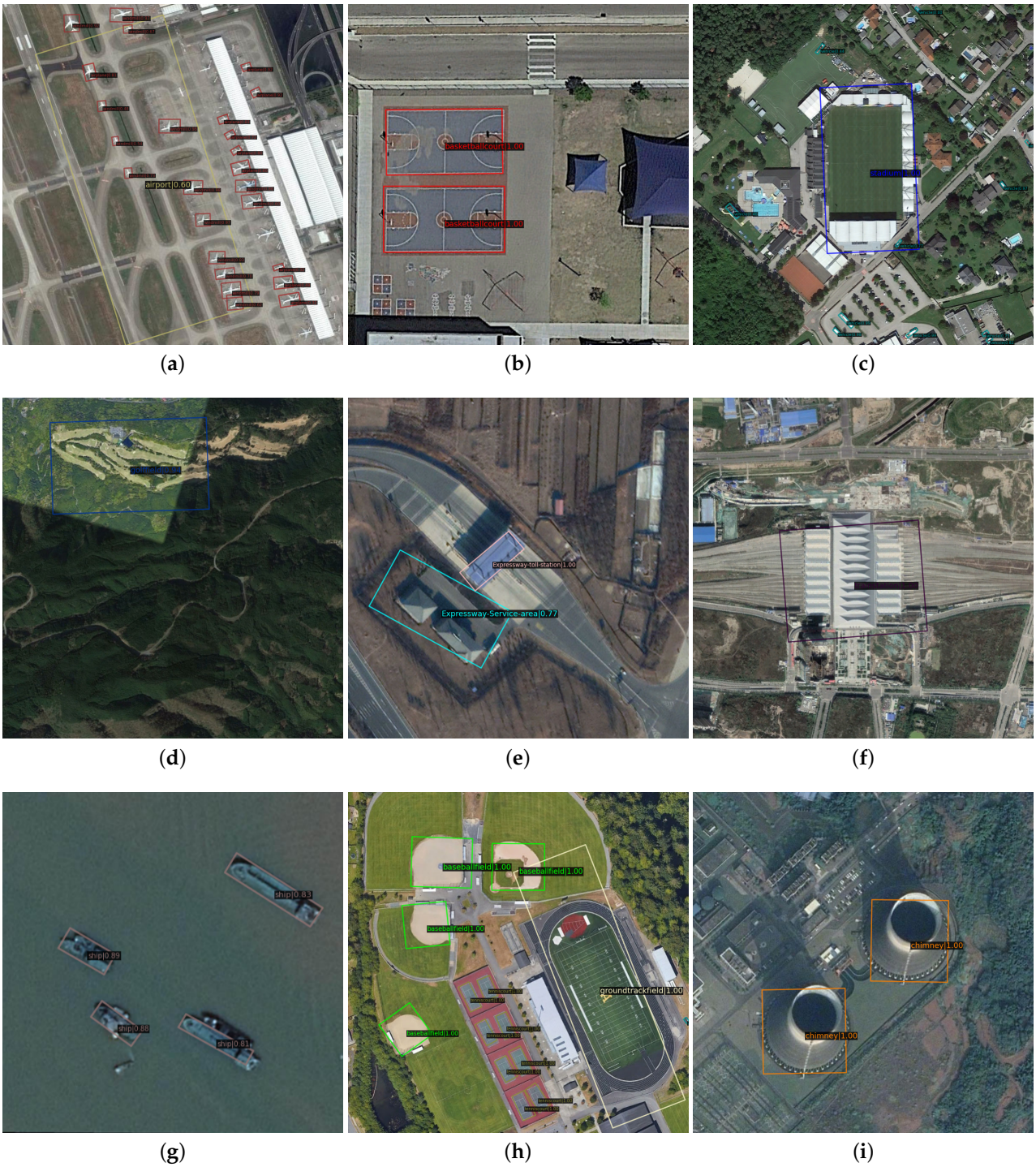
(**b**)

Figure 14. The impact of incorporating Euclidean distance attention into the model: (**a**) missed detection of dam; (**b**) false detection of tennis court.

(**a**)

(**b**)

Figure 15. The impact of applying the Euclidean distance attention model to the baseline model: (**a**) missed detection of bridge; (**b**) false detection of stadium.

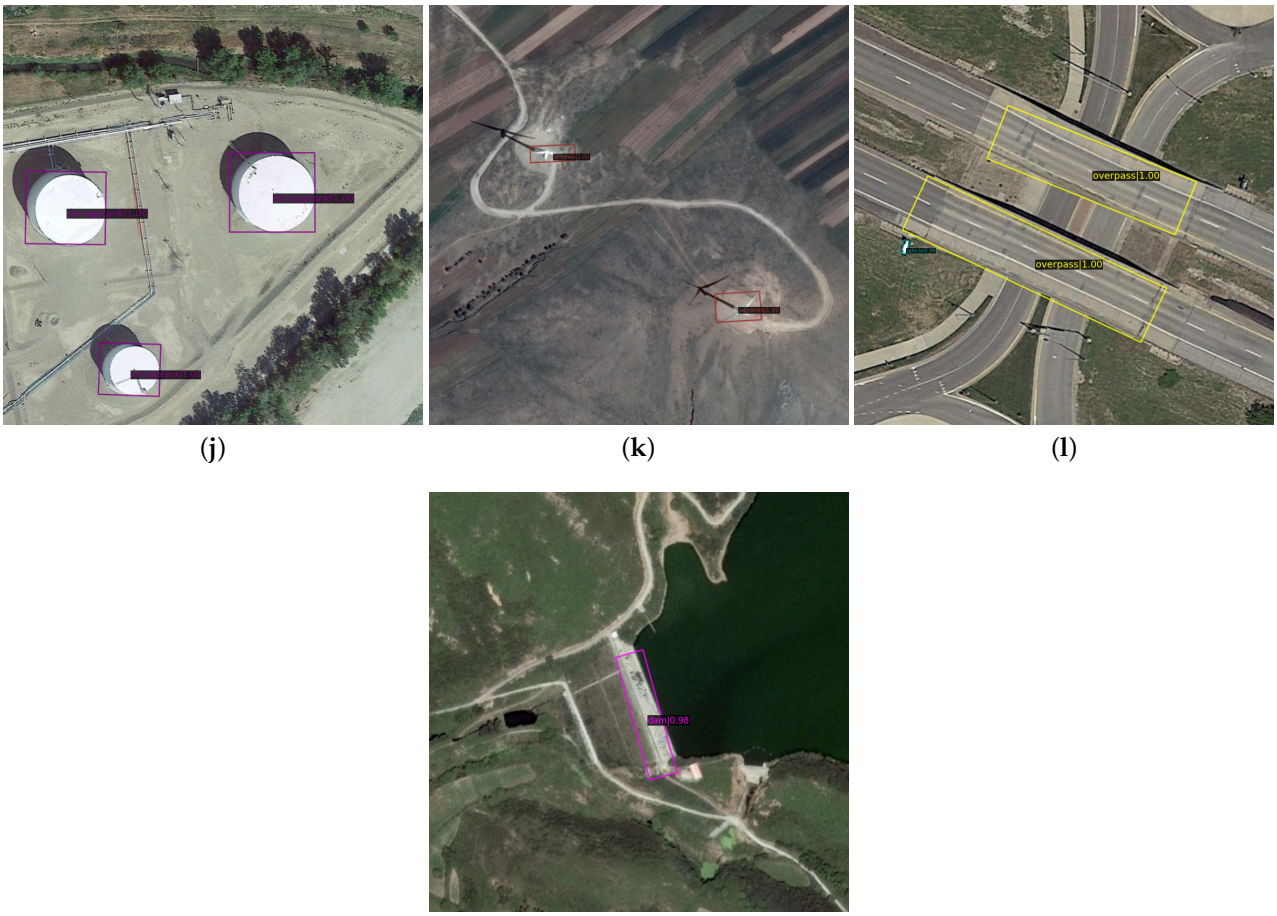


(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

(**g**) (**h**) (**i**)

Figure 16. *Cont.*



(**j**) (**k**) (**l**)

(**m**)

Figure 16. Visualization of the positive results: (**a**) airplane and airport; (**b**) basketball court; (**c**) vehicle and stadium; (**d**) golf field; (**e**) expressway service area and expressway toll station; (**f**) train station; (**g**) ship; (**h**) ground track field, baseball field, and tennis court; (i) chimney; (**j**) storage tank; (**k**) windmill; (**l**) overpass; (**m**) dam.

6. Conclusions

In this paper, we present a new parameter-free attention mechanism called EuPea, aimed at addressing the semantic information loss in remote sensing target detection during feature extraction, particularly regarding the lack of association between elements. It effectively integrates the Euclidean distances and Pearson correlation coefficients between elements, producing an attention-weighted feature map used to generate more robust features for neural networks. The experimental results show that EuPea outperforms other models, demonstrating a noteworthy 1.0% increase in average precision on the DOTA dataset, a 2.4% enhancement on the NWPU VHR-10 dataset, and a 1.8% boost in average precision on the DIOR dataset. The proposed attention mechanism significantly outperforms its counterparts in accuracy while maintaining a nearly identical inference time to the baseline model. The consistent advancements across various datasets emphasize the generalizability and adaptability of EuPea across diverse scenarios and data distributions. Moreover, it exhibits effective resistance against false-negative and false-positive detections. Furthermore, considering potential applications, EuPea exhibits promise in various real-world scenarios. Its adaptability to different remote sensing tasks and its potential impact on practical applications are noteworthy aspects that merit further exploration. As a parameter-free attention mechanism, the versatility of EuPea suggests that it could easily be integrated into diverse remote sensing applications, providing a valuable tool for researchers and practitioners alike.

Author Contributions: Conceptualization, X.G. and Y.D.; methodology, X.G. and Y.D.; software, X.G. and Y.D.; resources, W.T.; writing—original draft preparation, X.G.; writing—review and editing, Y.D. and W.T.; visualization, X.G. and Y.S.; supervision, P.H.; funding acquisition, Y.D. and P.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Natural Science Foundation of China under Grant U22A2010, in part by the Basic Research Fund Project for Universities Directly Affiliated with Inner Mongolia Autonomous Region under Grant JY20230008, and in part by the Science and Technology Planned Project of Inner Mongolia under Grant 2019GG138 and 2020GG0073.

Data Availability Statement: The data supporting the reported results are publicly available datasets. We utilized the following datasets in our study: DOTA dataset: https://captain-whu.github.io/DOTA/dataset.html; NWPU VHR-10 dataset: https://github.com/Gaoshuaikun/NWPU-VHR-10; DIOR dataset: https://aistudio.baidu.com/datasetdetail/123364.

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

References

1. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Richfeature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
2. Girshick, R. Fast r-cnn. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
3. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
4. Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; Xue, X. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Trans. Multimed.* **2018**, *20*, 3111–3122. [CrossRef]
5. Yang, X.; Yang, J.; Yan, J.; Zhang, Y.; Zhang, T.; Guo, Z.; Sun, X.; Fu, K. Scrdet: Towards more robust detection for small, cluttered and rotated objects. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8232–8241.
6. Zhang, G.; Lu, S.; Zhang, W. CAD-Net: A context-aware detection network for objects in remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 10015–10024. [CrossRef]
7. Zheng, G.; Zhang, F.; Zheng, Z.; Xiang, Y.; Yuan, N.J.; Xie, X.; Li, Z. DRN: A deep reinforcement learning framework for news recommendation. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 167–176.
8. Yang, X.; Yan, J.; Feng, Z.; He, T. R3det: Refined single-stage detector with feature refinement for rotating object. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 3163–3171. [CrossRef]
9. Han, J.; Ding, J.; Xue, N.; Xia, G.S. Redet: A rotation-equivariant detector for aerial object detection. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 2786–2795.
10. Ding, J.; Xue, N.; Long, Y.; Xia, G.S.; Lu, Q. Learning RoI transformer for oriented object detection in aerial images. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 2849–2858.
11. Xie, X.; Cheng, G.; Wang, J.; Yao, X.; Han, J. Oriented R-CNN for object detection. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 19–25 June 2021; pp. 3520–3529.
12. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
13. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
14. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
15. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the 2018 European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
16. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.
17. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
18. Yang, L.; Zhang, R.Y.; Li, L.; Xie, X. Simam: A simple, parameter-free attention module for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 11863–11874.
19. Liu, Y.; Shao, Z.; Teng, Y.; Hoffmann, N. NAM: Normalization-based attention module. *arXiv* **2021**, arXiv:2111.12419.

20. Shugar, D.H.; Jacquemart, M.; Shean, D.; Bhushan, S.; Upadhyay, K.; Sattar, A.; Schwanghart, W.; McBride, S.; De Vries, M.V.W.; Mergili, M.; et al. Massive rock and ice avalanche caused the 2021 disaster at Chamoli, Indian Himalaya. *Science* **2021**, *373*, 300–306. [CrossRef] [PubMed]
21. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
22. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
23. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
24. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*.
25. Liu, Z.; Hu, J.; Weng, L.; Yang, Y. Rotated region based CNN for ship detection. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 900–904.
26. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Honolulu, HI, USA, 21–26 July 2017; pp. 900–904.
27. Yang, Z.; Liu, S.; Hu, H.; Wang, L.; Lin, S. Reppoints: Point set representation for object detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9657–9666.
28. Yang, X.; Yan, J. Arbitrary-oriented object detection with circular smooth label. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Part VIII 16; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 677–694.
29. Chun, M.M. Visual working memory as visual attention sustained internally over time. *Neuropsychologia* **2011**, *49*, 1407–1409. [CrossRef] [PubMed]
30. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning deep features for discriminative localization. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2921–2929.
31. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Honolulu, HI, USA, 21–26 July 2017; pp. 618–626.
32. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
33. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Honolulu, HI, USA, 21–26 July 2017; pp. 2961–2969.
34. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A large-scale dataset for object detection in aerial images. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983.
35. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2009.
36. Muhuri, A.; Gascoin, S.; Menzel, L.; Kostadinov, T.S.; Harpold, A.A.; Sanmiguel-Vallelado, A.; López-Moreno, J.I. Performance Assessment of Optical Satellite-Based Operational Snow Cover Monitoring Algorithms in Forested Landscapes. *IEEE JSTARS* **2022**, *14*, 7159–7178. [CrossRef]
37. Zhou, Y.; Yang, X.; Zhang, G.; Wang, J.; Liu, Y.; Hou, L.; Jiang, X.; Liu, X.; Yan, J.; Lyu, C.; et al. Mmrotate: A rotated object detection benchmark using pytorch. In Proceedings of the 30th ACM International Conference on Multimedia, Lisboa, Portugal, 10–14 October 2022; pp. 7331–7334.
38. Yang, X.; Yan, J. On the arbitrary-oriented object detection: Classification based approaches revisited. *Int. J. Comput. Vis.* **2022**, *130*, 1340–1365. [CrossRef]