



Article

Task-Sensitive Efficient Feature Extraction Network for Oriented Object Detection in Remote Sensing Images

Zhe Liu ¹, Guiqing He ^{1,*}, Liheng Dong ¹, Donglin Jing ² and Haixi Zhang ³¹ School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710129, China; liu_zhe@mail.nwpu.edu.cn (Z.L.); dlh@mail.nwpu.edu.cn (L.D.)² School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China; jingdonglin@bit.edu.cn³ College of Information Engineering, Northwest A&F University, Xianyang 712100, China; zh.haixi@nwfau.edu.cn

* Correspondence: guiqing_he@nwpu.edu.cn

Abstract: The widespread application of convolutional neural networks (CNNs) has led to significant advancements in object detection. However, challenges remain in achieving efficient and precise extraction of critical features when applying typical CNN-based methods to remote sensing detection tasks: (1) The convolutional kernels sliding horizontally in the backbone are misaligned with the features of arbitrarily oriented objects. Additionally, the detector shares the features extracted from the backbone, but the classification task requires orientation-invariant features while the regression task requires orientation-sensitive features. The inconsistency in feature requirements makes it difficult for the detector to extract the critical features required for each task. (2) The use of deeper convolutional structures can improve the detection accuracy, but it also results in substantial convolutional computations and feature redundancy, leading to inefficient feature extraction. To address this issue, we propose a Task-Sensitive Efficient Feature Extraction Network (TFE-Net). Specifically, we propose a special mixed fast convolution module for constructing an efficient network architecture that employs cheap transform operations to replace some of the convolution operations, generating more features with fewer parameters and computation resources. Next, we introduce the task-sensitive detection module, which first aligns the convolutional features with the targets using adaptive dynamic convolution based on the orientation of the targets. The task-sensitive feature decoupling mechanism is further designed to extract orientation-sensitive features and orientation-invariant features from the aligned features and feed them into the regression and classification branches, respectively, which provide the critical features needed for different tasks, thus improving the detection performance comprehensively. In addition, in order to make the training process more stable, we propose a balanced loss function to balance the gradients generated by different samples. Extensive experiments demonstrate that our proposed TFE-Net can achieve superior performance and obtain an effective balance between detection speed and accuracy on DOTA, UCAS-AOD, and HRSC2016.

Keywords: aerial object detection; lightweight network; orientation features; convolutional neural network



Citation: Liu, Z.; He, G.; Dong, L.; Jing, D.; Zhang, H. Task-Sensitive Efficient Feature Extraction Network for Oriented Object Detection in Remote Sensing Images. *Remote Sens.* **2024**, *16*, 2271. <https://doi.org/10.3390/rs16132271>

Academic Editors: Gonzalo Pajares Martinsanz, Mingyang Zhang and Hao Li

Received: 19 May 2024

Revised: 17 June 2024

Accepted: 18 June 2024

Published: 21 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Aerial object detection aims to automatically identify the categories and locations of objects of interest (e.g., aircraft, ships, bridges, etc.) in aerial images using image processing and computer vision techniques, and has been widely used in many fields such as environmental monitoring, smart cities, military reconnaissance, etc. In recent years, the widespread application of deep learning techniques such as convolutional neural networks (CNNs) has led to significant advancements in aerial object detection [1–6]. An important trend is the commitment to address the challenges posed by the limited computation resources of embedded devices and the diverse shape of objects.

The existing CNN-based detection frameworks are primarily classified into two categories: one-stage and two-stage detectors. For most of the one-stage detectors [7–10], the features are first extracted by convolutional operations, then a series of a priori horizontal anchors are preset on the feature maps, and finally, classification and regression are carried out based on these anchors and feature maps in order to obtain the bounding box of the objects. However, targets in aerial images exhibit arbitrary orientations. In such scenarios, employing horizontal anchors may induce misalignment between targets and candidate regions, thereby introducing significant background interference in subsequent tasks. To address the misalignment in horizontal candidate regions during the detection of rotated objects, Gliding Vertex [11] mitigates the interference of these regions by additionally learning the offsets of the four vertices of the detection box. There are also methods [12–15] to localize objects in arbitrary orientations by using predefined anchors of various angles, scales, and aspect ratios; however, these methods usually entail a large number of redundant computations and struggle to extract the key features that are distinctively applicable to the tasks of regression and classification. Compared with the one-stage detectors, the two-stage detectors [16–18] utilize the region proposal network to generate high-quality regions of interest, achieving higher detection performance. For instance, RoI-Transformer [19] converts horizontal candidate regions into rotated candidate regions, effectively avoiding interference from the background. However, these two-stage detectors introduce more complex convolutional structures, resulting in higher computational complexity and slower inference time, which makes it difficult to apply them to hardware platforms with limited resources.

At present, more and more lightweight models are used to improve rotated object detectors to solve the problem of the limited computation resources of remote sensing platforms. SqueezeNet [20] succeeds in reducing the number of parameters in the network using techniques such as smaller convolutional kernels and global average pooling, but this also results in a relatively deep network structure that loses the parallelism of the network, making the inference time longer instead. InceptionV3 [21] introduces the core idea of “factorization” to break up larger kernels into smaller ones. EfficientNet [22] proposes a composite scaling approach using neural architecture search (NAS) that enhances the selection process across the dimensions of width, depth, and resolution. These methods utilize the redundancy of filters to reduce the number of parameters and FLOPs while suffering from increased memory access. Some works reduce redundant features by network pruning, Han et al. [23] proposed pruning connections with small weights in neural networks, and Liu et al. [24] used L1 norm regularization to prune filters and construct more efficient CNN structures.

In summary, the mainstream lightweight network architectures still face issues with low computational efficiency in the feature extraction process and difficulties in precisely extracting high-quality key features when applied to remote sensing detection tasks. Specifically, the construction of deep convolutional neural networks is often accompanied by a considerable amount of convolutional operations and redundant features, which will lead to an increase in memory access frequency and FLOPs. According to Figure 1a, it can be seen that the feature maps between different channels have a high degree of similarity, and we believe that the redundancy in the feature maps may be exactly the key to achieving accurate localization of small objects densely packed in aerial images, because most of these seemingly similar redundant feature maps contain fine-grained information such as the edge texture of the objects. Unlike previous works, instead of avoiding the redundant feature maps, we embrace them in a simple and efficient way to achieve less memory access and convolutional computation. In terms of detection performance, the convolutional kernels sliding horizontally are misaligned with the features of arbitrarily oriented objects, making it difficult to extract high-quality spatial features of these objects. Moreover, the detectors share the features extracted from the backbone, but the classification task requires orientation-invariant features to correctly classify targets distributed in any direction, and the regression task requires orientation-sensitive features in order to achieve direction

prediction for precise target localization [25,26]. If the detection head cannot effectively provide the required key features for each task separately, it may cause inconsistencies between classification accuracy and regression results [27,28], where bounding boxes fail to precisely cover the targets with arbitrary orientation. These issues will degrade the detection accuracy of the detector. Meanwhile, during the training process, the imbalanced gradients generated by different samples will further limit the detection performance. As shown in Figure 1c, compared with natural scenes, objects in remote sensing images are more likely to generate large-gradient samples due to their large aspect ratio. Regressing the blue preset anchor to the red bounding box in the remote sensing image will generate large gradient, and these samples with large gradients do not contain the crucial object features, but instead introduce noise interference, adversely affecting the accurate classification of objects. Therefore, in order to achieve optimal convergence, the different tasks and samples involved need to be properly balanced.

From the above analysis, a challenging issue is how to realize efficient and precise feature extraction with full consideration of remote sensing target characteristics, which is beneficial for the detector to achieve a balance between speed and accuracy. To this end, we propose a Task-Sensitive Efficient Feature Extraction Network (TFE-Net) that consists of three components, including the special mixed fast convolution module, task-sensitive detection module, and balanced loss. The special mixed fast convolution module, used to create an efficient network architecture, applies a series of linear transformation operations to replace some of the convolutional operations based on the redundancy of the feature maps in the middle layer of the convolutional neural network, which can simultaneously reduce the computational complexity and memory access. In the task-sensitive detection module, an adaptive dynamic convolution mechanism is proposed, which can align the feature maps with the rotated object based on its orientation to enhance the representation ability of the features. Then, a feature decoupling mechanism is designed on the aligned features, which first generates orientation-sensitive features for the regression branch through a set of multi-orientation filters. Subsequently, the strongest responses are extracted from the orientation-sensitive features through pooling operations to serve as orientation-invariant features. This design helps to avoid feature interference between different tasks and to effectively extract the critical features needed for specific tasks. Finally, during the training process, the balanced loss function successfully mitigates the potential impact of abnormal gradients arising from significant discrepancies between preset anchors and bounding boxes and it enhances the contribution of high-quality, small-gradient samples in network training. This effectively balances the involved samples and tasks, facilitating the attainment of optimal convergence results. The main contributions of our work are as follows:

- (1) We systematically analyze the issue that existing detectors, when applied to remote sensing detection tasks, suffer from difficulty in extracting features precisely and efficiently, which limits further improvements in overall performance.
- (2) A special mixed fast convolution module is constructed by taking advantage of the feature redundancy in convolutional neural networks. Part of the traditional convolution operations are replaced with a simpler and cheaper way, effectively reducing the computational complexity and memory access.
- (3) A task-sensitive detection module is proposed, which employs adaptive dynamic convolution to align targets with arbitrary orientations, and further, utilizes a task-sensitive feature decoupling mechanism to effectively extract critical features required for classification and regression tasks from the aligned features, thereby enhancing detection performance.
- (4) A balanced loss function is proposed, which rebalances the gradients generated by different quality samples to make the process of network training more stable.

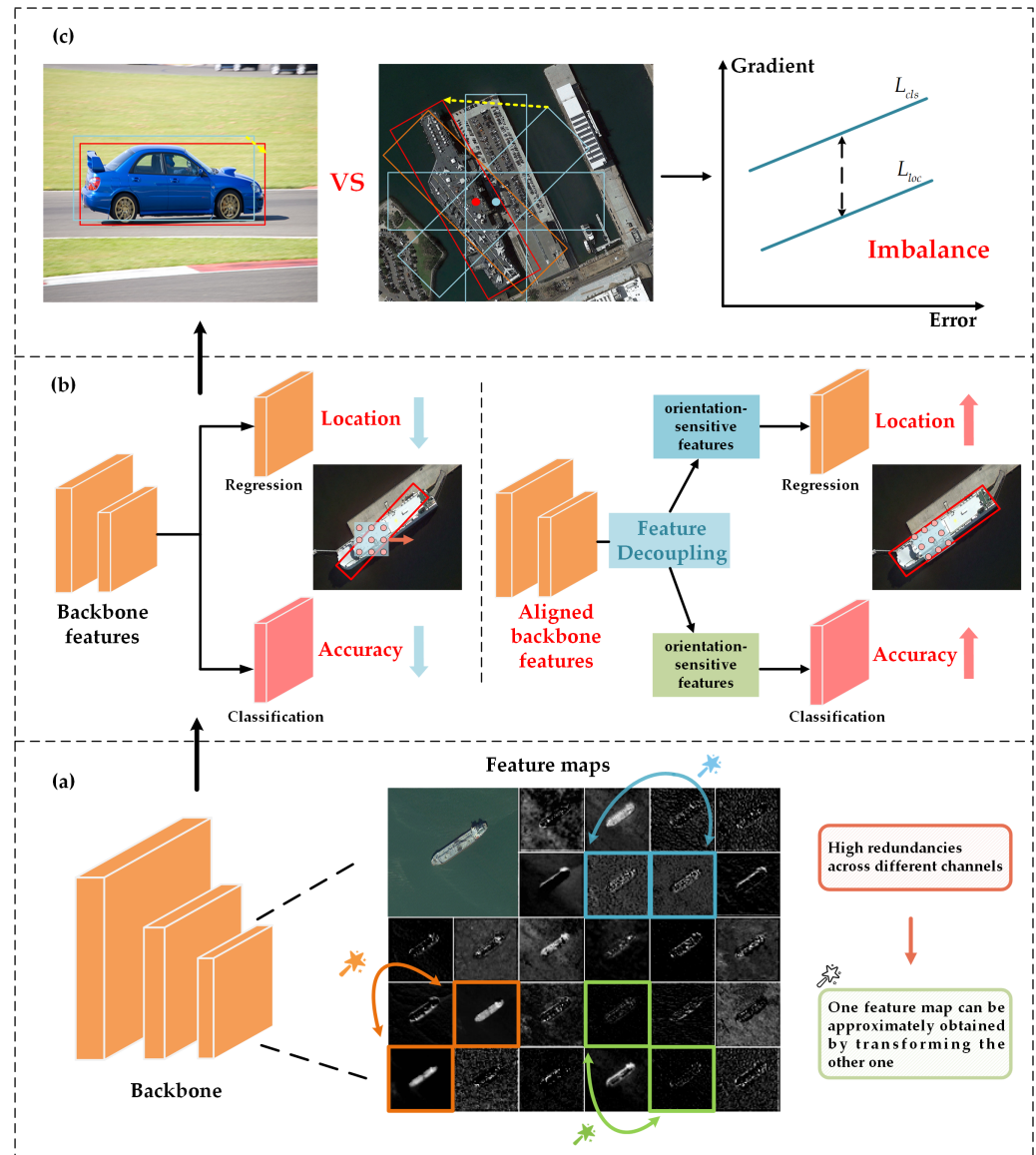


Figure 1. (a) Visualization of feature maps in an intermediate layer of a pretrained ResNet50, where three examples of similar feature map pairs are marked with boxes of the same color. We can see the high redundancies in the pair. (b) The detection head accurately feeds the critical features required for the classification branch and the regression branch, which contributes to the overall improvement in the detection performance. (c) In comparison to natural scenes, remote sensing object detection is more prone to sample imbalance during the training process.

2. Related Works

2.1. Oriented Object Detection

Objects in remote sensing images have characteristics of arbitrary orientations and dense distributions, and it is difficult to compactly and accurately position them based on horizontal bounding boxes. For example, a single horizontal bounding box may often contain multiple densely distributed objects. Moreover, typical object detection networks lack the ability to capture rotation features, which may lead to missed and false detections. To address these issues, the mainstream research primarily concentrates on improving rotation feature extraction and optimization of candidate regions.

FFA [29] extends the region proposal network (RPN) foundation of Faster RCNN by incorporating angle information and introducing angular regression loss into the localization loss function. This enhancement improves the perception of rotational angle information, enabling the generation of bounding boxes that more accurately reflect the actual shape and

orientation of the object. Cheng et al. [30] applied a regularization constraint to the objective function, enforcing that the feature representations of targets from different orientations be mapped to be nearly identical, thereby achieving rotation invariance. DRN [31] introduces a module that adaptively adjusts the receptive fields of neurons in accordance with the object shapes and orientations, effectively mitigating misalignments between the receptive fields and the targets. Additionally, it incorporates a dynamic refinement head to model the uniqueness of each sample, addressing the issue of poor regression prediction flexibility associated with models with fixed parameters. The above methods aim to select additional appropriate information at the object level for extracting rotation features. To achieve a more refined granularity of feature representation, Zhang et al. [32] proposed extending the region-of-interest pooling layer to a rotated region-of-interest pooling layer, which better preserves the orientation angle information of the objects and reduces the loss of rotation features. ReDet [33] incorporates rotation-equivariant networks into the detector to extract rotation-equivariant features and adaptively extract rotation-invariant features from these equivariant features. However, the above methods have complex structures and high computational complexity.

In the optimization of candidate regions, horizontal anchor boxes often lead to misalignments between bounding boxes and objects. DRBox [34] improves the algorithm's robustness to the rotation of target objects by incorporating anchor boxes of various scales, angles, and aspect ratios. Ding et al. [19] proposed the RoI Transformer network, which is capable of converting horizontal regions of interest (RoIs) into rotated RoIs. This approach not only reduces the number of anchor boxes but also effectively alleviates the misalignment issues between RoIs and objects. However, the meticulous design of rotated RoIs often leads to high computational complexity and sensitivity to the hyperparameters of anchor boxes. Xu et al. [11] proposed sliding each vertex on horizontal bounding boxes to more accurately describe multi-oriented objects. Wang et al. [35] proposed GA-RPN, which dynamically generates anchors of arbitrary shapes by jointly predicting the locations and anchor shapes dependent on locations. However, the aforementioned method, when applied to remote sensing object detection with numerous distractors and complex environmental backgrounds, is susceptible to interference and exhibits poor robustness.

2.2. Lightweight Network Design

Remote sensing platforms have limited computational resources and high requirements for hardware deployment. To address these issues, current mainstream efforts primarily focus on optimization through two approaches: model compression methods and the design of lightweight network architectures.

Model compression methods aim to reduce the size of deep learning models, decreasing their computational and storage requirements while maintaining or minimizing performance degradation on specific tasks. Common techniques for model compression include pruning and model quantization. Pruning reduces the size and complexity of a model by selectively eliminating non-essential or redundant parameters (e.g., neurons, channels, or filters). Hu et al. [36] proposed a dynamic pruning method with a multiple sparsity structure, employing a weight evaluation mechanism to determine whether weights are active or inactive, rather than directly removing weights with low scores. This approach not only fully utilizes the complete model capacity but also retains information from the original network, helping to mitigate reductions in accuracy. Lee et al. [37] proposed methods for channel pruning through residual connections which have promising performance despite their simplicity and efficiency. Model quantization is a technique that converts model parameters to a lower precision format. Benoit Jacob et al. [38] propose a quantization scheme that quantizes weights and activations into 8-bit integers and residual vectors into 32-bit integers, thereby improving the trade-off between accuracy and latency on hardware devices. These model compression techniques may lead to model deformation or loss of detail, resulting in a reduction in accuracy, and their performance is constrained by the upper limits of the pretrained model.

Due to the limited computational resources of embedded systems, which cannot accommodate large-scale and computationally intensive deep learning models, a series of lightweight networks have been proposed in recent years. MobileNets [39] is a family of lightweight deep neural networks built on depthwise separable convolutions. MobileNetv2 [40] introduces a new module with an inverted residual and linear bottleneck. MobileNetv3 [41] further utilizes a composite search technique, with module-level search conducted by NAS (network architecture search) and local search executed using the NetAdapt algorithm. ShuffleNet [42] introduces the channel shuffle operation to facilitate information flow across feature channels, thereby addressing the issue of reduced neural network expressive capacity caused by group convolutions. ShuffleNetV2 [43] further takes into account the actual time consumption of network deployment on different hardware to design the network structure. The above methods mainly use efficient convolutions to reduce the number of parameters in the network, but they fail to solve the problem of the model's memory consumption. Wang et al. [44] proposed a lightweight CNN that utilizes a simple convolution combined with pooling structure for ship detection in remote sensing images. This framework is exclusively applicable to detection tasks of a particular category and will encounter difficulties in the complex situations of detecting multiple classes of objects.

3. Methodology

The construction of convolutional neural networks often involves extensive convolutional operations and redundant features, which can lead to an increase in memory access frequency and floating point operations (FLOPs). For this reason, in this paper, an efficient lightweight module was constructed by effectively utilizing redundant features; in addition, a task-sensitive detection module was proposed to enhance the representation capability of target features, and to further extract the critical features for different tasks. Specifically, a series of cheap linear operations and specific residual structures are applied to replace some of the convolution operations, reducing FLOPs and memory access. Then, adaptive dynamic convolution is used to align the axis-aligned features generated by the backbone network with the oriented objects, and the orientation-sensitive and orientation-invariant features are further extracted from the aligned features and fed into the regression and classification branches, respectively. In the end, constraints are applied to samples generating large gradients during training, while the contribution of high-quality samples with small gradients is enhanced to realize a balanced training of each module of TFE-Net. The proposed TFE-Net framework is shown in Figure 2.

3.1. Special Mixed Fast Convolution

The existing lightweight methods often adopt depthwise separable convolutions, which can effectively reduce the number of model parameters. However, in practice, it is common to increase the number of channels in depthwise separable convolutions in order to prevent significant accuracy degradation. This leads to more frequent memory access, resulting in higher latency and a slowdown in overall computation speed [45,46]. It is evident that the contribution of depthwise separable convolutions to reducing the number of parameters cannot directly translate into reduced latency, making the lightweight modules not fast enough in reality.

Given the high similarity between feature maps of different channels, we propose that the redundancy of features could be effectively utilized to reduce the number of convolution filters. Specifically, the convolutional layers in depthwise separable convolutions will be divided into two parts. The first part applies regular convolutions, but their total number will be strictly controlled. Then, a series of linear operations are performed on the feature maps generated by the first part to produce all feature maps, and finally, the feature maps from both parts are fused along the channel dimension.

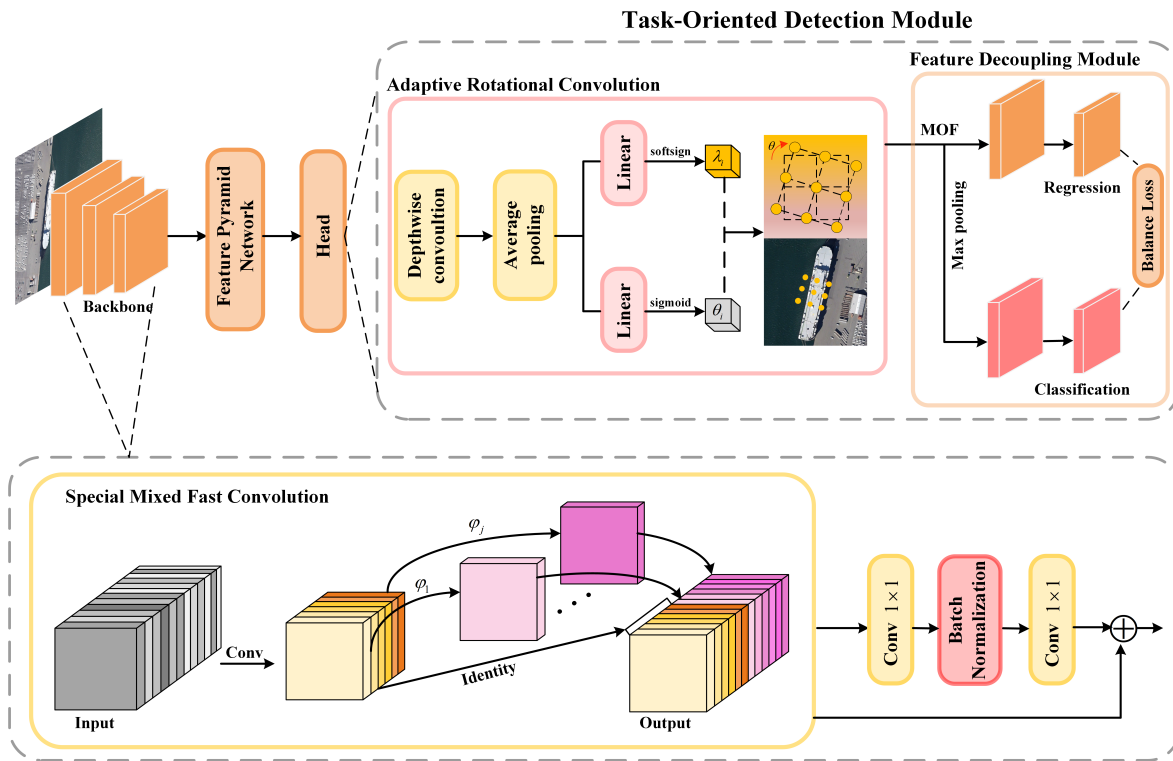


Figure 2. Architecture of the proposed TFE-Net.

Given the input data $\mathbf{X} \in \mathbb{R}^{c \times h \times w}$, where c represents the number of input channels, and w and h denote the width and height of the input, respectively, then, the process of generating feature maps by the depth convolution layer can be simplified as

$$\mathbf{Y} = \mathbf{X} * \mathbf{F} + \gamma \quad (1)$$

where $\mathbf{Y} \in \mathbb{R}^{h' \times w' \times m}$ is the output feature map with m channels, $*$ represents the convolution operation, $\mathbf{F} \in \mathbb{R}^{c \times k \times k \times m}$ denotes the convolution filters, and γ is the bias term. The kernel size of the convolution filters $\mathbf{F}$ is $k \times k$, and the output feature map height and width correspond to h' and w' , respectively. In this convolution process, the required number of FLOPs Q is equal to $m \cdot h' \cdot w' \cdot c \cdot k \cdot k$.

As visualized in Figure 1a, the feature maps share high similarities among different channels. We first generate a portion of feature maps through regular convolutions, which are referred to as intrinsic features. Utilizing the similarity among feature maps, it is assumed that the remaining feature maps can be derived from the intrinsic features through some cheap transformations, referred to as extrinsic features. This approach saves the amount of FLOPs and memory access required to individually generate these redundant feature maps. Specifically, this can be expressed as the following formula:

$$\mathbf{Y}' = \mathbf{X} * \mathbf{F}' + \gamma \quad (2)$$

Using convolution filters to generate n intrinsic feature maps $\mathbf{Y}' \in \mathbb{R}^{h' \times w' \times n}$, $n \leq m$, and $\mathbf{F}' \in \mathbb{R}^{c \times k \times k \times n}$ utilizes filters. To further obtain the required m feature maps, we perform a series of simple linear operations on each feature map in $\mathbf{Y}'$:

$$\mathbf{Y}_{i,j}^* = \varphi_{i,j}(\mathbf{Y}_i'), \quad i = 1, \dots, n \quad j = 1, \dots, q \quad (3)$$

where $\mathbf{Y}_i'$ represents the i -th feature map in $\mathbf{Y}'$, and $\mathbf{Y}_{i,j}^*$ represents the j -th feature map generated from $\mathbf{Y}_i'$. The function $\varphi_{i,j}$ represents a linear operation, where the first $j - 1$ transformations are cheap linear operations, and the last transformation is the identity

mapping of the original feature map $\mathbf{Y}'$. Through the above formula, we can obtain $m = n * q$ feature maps $\hat{\mathbf{Y}} = [\mathbf{Y}_{11}^*, \mathbf{Y}_{12}^*, \dots, \mathbf{Y}_{nq}^*]$. Because linear operations are used instead of original convolution operations to generate redundant feature maps, it effectively reduces computational cost and memory consumption.

For instance, by selecting one identity mapping and $n(q-1) = \frac{m}{q}(q-1)$ linear transformations, with each linear operation having an average kernel size of $l \times l$, it can be calculated that the required number of FLOPs Q' is equal to $\frac{m}{q} \cdot h' \cdot w' \cdot c \cdot k \cdot k + (q-1) \cdot \frac{m}{q} \cdot h' \cdot w' \cdot l \cdot l$. To achieve efficient inference, we propose adopting linear operations of the same size within the module, and through theoretical calculations, it can be determined that our proposed module is accelerated by q times compared to traditional convolution.

$$\begin{aligned} r_s &= \frac{Q}{Q'} \\ &= \frac{m \cdot h' \cdot w' \cdot c \cdot k \cdot k}{\frac{m}{q} \cdot h' \cdot w' \cdot c \cdot k \cdot k + (q-1) \cdot \frac{m}{q} \cdot h' \cdot w' \cdot l \cdot l} \\ &\approx q. \end{aligned} \quad (4)$$

where $l \times l$ is of a similar magnitude to $k \times k$, and $q \ll c$. Likewise, we can obtain the compression ratio by the following formula:

$$\begin{aligned} r_c &= \frac{m \cdot c \cdot k \cdot k}{\frac{m}{q} \cdot c \cdot k \cdot k + (q-1) \cdot \frac{m}{q} \cdot l \cdot l} \\ &\approx \frac{q \cdot c}{q + c - 1} \approx q. \end{aligned} \quad (5)$$

In order to fully utilize the information from all channels, further point-to-point convolution is added; together these form an inverted residual block. Through such operations, associations between different intrinsic and extrinsic features can be established, achieving feature fusion across channels and enhancing the extraction capability of spatial features. The normalization operation is also essential for high-performance neural networks; for faster inference speeds we add a batch normalization operation after the deep separable convolution.

3.2. Task-Sensitive Detection Module

In the majority of remote sensing detection frameworks, the regression and classification tasks depend on shared features extracted from the backbone, and such highly coupled features do not take into account the uniqueness of the two tasks [25], a problem that is magnified under the characteristics of variable rotation and multiple scales of remote sensing objects. For example, classifiers need orientation-invariant features to correctly classify objects with various orientations, and regressors need orientation-sensitive features to achieve accurate orientation prediction, and thus, precise object localization. Meanwhile, the convolution process sliding along a fixed direction has difficulty in accurately modeling the features of targets distributed in arbitrary directions.

To address the issue of the detection head struggling to extract the high-quality key features needed for classification and regression tasks, this paper proposes the task-sensitive detection module. The architecture of the task-sensitive detection module is shown in Figure 3. Firstly, we introduce adaptive dynamic convolution that resamples the spatial sampling positions of the convolution kernel based on the orientation of the object, to solve the misalignment problem between the backbone network features and the object. Further, a task-sensitive feature decoupling mechanism is performed on the aligned features to extract orientation-sensitive and orientation-invariant features to be fed into the regression and classification branches, respectively.

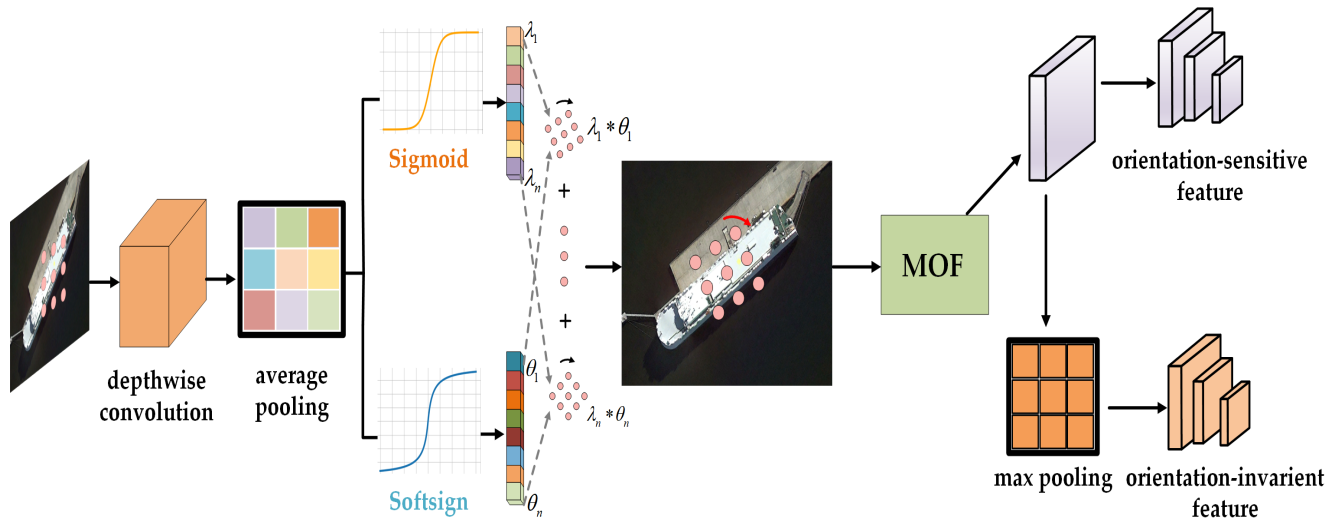


Figure 3. Architecture of the TDM.

Adaptive dynamic convolution. The input feature $\mathbf{X}$ is first fed into a lightweight deep convolution of 3×3 convolutional kernel, after which, a layer normalization and ReLU activation are applied, and then, the activated features are passed to two different branches after average pooling. The first branch is used to predict rotation angle features $(\theta_1, \theta_2, \dots, \theta_n)$, consisting of a linear layer and a softsign activation that sets no bias and expands the rotation range by multiplying the output of the softsign layer by a rotation coefficient. The second branch is in charge of predicting the combination weights $(\lambda_1, \lambda_2, \dots, \lambda_n)$ of the spatial feature. Similar to the rotation angle prediction branch, it is also constructed by a linear layer, but utilizes the Sigmoid activation and with bias. The ADC module has m kernels $(W_1, W_2, \dots, W_m)$, which are first rotated individually according to the predicted rotation angle:

$$W'_i = R(W_i, \theta_i), i = 1, 2, \dots, m, \quad (6)$$

where W'_i is the i -th rotated kernel, θ_i denotes the rotation angle for W_i , and $R(\cdot)$ is defined as the rotation function, which fundamentally can be regarded as resampling. Specifically, as shown in Figure 4, we consider the convolution weight as sampled points from a kernel space, and use linear interpolation techniques to span the original sample points to a 2D kernel space. The sample coordinates are obtained for the new rotated convolution kernel by rotating the original convolution kernel coordinates clockwise around the center point by θ degrees. We then resample the values in the original kernel space according to the new coordinates to obtain the rotated convolution kernel. After obtaining the rotated convolution kernel and the combination weights λ , the output feature maps $\mathbf{Y}$ can be simply defined as

$$\mathbf{Y} = \lambda_1(W'_1 * \mathbf{X}) + \lambda_2(W'_2 * \mathbf{X}) + \dots + \lambda_m(W'_m * \mathbf{X}). \quad (7)$$

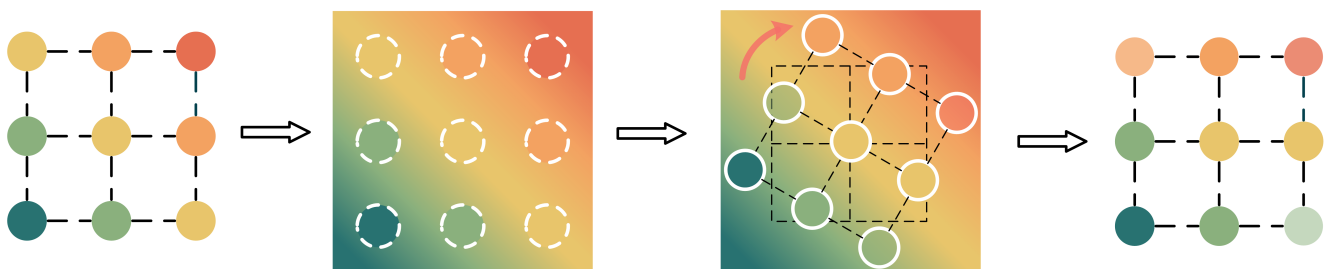


Figure 4. The process of a 3×3 convolution kernel rotation.

Inspired by the conditional parameterization technique [47], the above formula can be written as

$$\mathbf{Y} = (\lambda_1 W'_1 + \lambda_2 W'_2 + \cdots + \lambda_m W'_m) * \mathbf{X}, \quad (8)$$

where $\mathbf{Y}$ denotes the feature map after alignment by adaptive dynamic convolution, $\lambda = [\lambda_1, \lambda_2, \cdots, \lambda_m]$ is the combination weights, and $*$ is the convolution operation. This method aligns features according to the orientation of the object, increasing the representation capability of the network for objects with various orientations and facilitating the further encoding of orientation features. Moreover, Equation (8) conducts only one convolution operation compared to Equation (7), which makes the method highly efficient.

Task-sensitive feature decoupling mechanism. This method is used to solve the problem of inconsistency in the features required by classifiers and regressors. We first use multi-orientation filters (MOFs) [48] on the aligned features to extract orientation-sensitive features for precise object localization. Specifically, a MOF is a $k \times k \times N$ convolution filter which actively rotates $N-1$ times during the convolution operation to produce a feature map with N orientation channels. The output feature map of the i -th orientation channel for a feature map $\mathbf{Y}$ and an MOF $\mathbf{F}$ can be calculated as

$$\mathbf{Y}'_{(i)} = \sum_{n=0}^{N-1} \mathbf{F}_{\theta_i(n)} * \mathbf{Y}_{(n)}, \quad \theta_i = i \frac{2\pi}{N}, \quad i = 0, 1, \cdots, N-1, \quad (9)$$

where $\mathbf{F}_{\theta_i}$ is the version of $\mathbf{F}$ rotated by θ degrees in a clockwise direction, the n -th orientation channels of $\mathbf{F}_{\theta_i}$ and $\mathbf{Y}$ are denoted as $\mathbf{F}_{\theta_i(n)}$ and $\mathbf{Y}_{(n)}$, respectively. When an MOF is applied to the convolutional layer, it can encode information from each orientation channel into a feature map, thereby obtaining outputs from multiple orientation channels as orientation-sensitive features. Unlike the regression of the bounding box, which requires orientation-sensitive features, the classification task benefits from orientation-invariant features. And we aim to obtain orientation-invariant features by pooling the orientation-sensitive features [48], which can be simply achieved by selecting the orientation channel with the strongest response as the output $\hat{\mathbf{Y}}$.

$$\hat{\mathbf{Y}} = \max \mathbf{Y}_{(n)}. \quad (10)$$

In this way, the detector can effectively focus on the critical features of the object in any orientation and reduce the interference of orientation information on object classification, which not only decreases the number of parameters but also enhances the robustness of the classification task. The experiments verified that feeding orientation-invariant features into the classifier resulted in higher classification accuracy for oriented objects, while feeding orientation-sensitive features into the regressor improved the precision of object position localization.

3.3. Balanced Loss

In existing remote sensing detectors, the multi-task loss [49] is usually defined as

$$L_{p,t,s^k,v} = L_{cls}(p,t) + \lambda L_{loc}(s^k,v), \quad (11)$$

where L_{cls} and L_{loc} are the objective functions for object recognition and localization, respectively. p and t denote the predictions and targets in L_{cls} . λ is utilized in multi-task learning to tune the loss weight. s^k is the results of the corresponding regression for class k , and v is the regression target. In order to balance the tasks involved it is often necessary to tune the loss weights; however, for the regression task, since the regression objects are unbounded, a direct increase in the weight of the localization loss will make the model more sensitive to outliers. We refer to samples with a loss greater than or equal to 1.0 as outliers, and other samples as inliers. These outliers can be considered as hard samples that will produce excessively large gradients. During the training process, samples with large gradients often receive more attention, while those with small gradients are not adequately

trained. This leads to an imbalance in the training samples, which in turn impairs the performance of subsequent tasks.

Therefore, we propose a balanced loss function to suppress the contribution of samples that generate large gradients in training due to a significant deviation between the anchor boxes and ground truth boxes. For the training process, these large-gradient samples are harmful. At the same time, it promotes the gradients from high-quality samples that are accurate but contribute little gradient to the overall gradients, achieving a proper balance between large- and small-gradient samples during the training process.

The ground truth box's position is denoted by $(x', y', w', h', \theta')$, while the anchor box's position is denoted by $(x^*, y^*, w^*, h^*, \theta^*)$. (x, y) denotes the center coordinates of the box, (w, h) represents the length and width of the box, and θ is the angle of the box's longer side in the horizontal direction. The parameterization of the ground truth regression offset $v = (v_x, v_y, v_w, v_h, v_\theta)$ is defined as follows:

$$\begin{aligned} v_x &= (x' - x^*)/w^*, v_y = (y' - y^*)/h^* \\ v_w &= \log(w'/w^*), v_h = \log(h'/h^*) \\ v_\theta &= \theta' - \theta^*. \end{aligned} \quad (12)$$

Balanced loss is derived from the conventional smooth L1 loss, which is denoted as L_b . The key idea of balanced loss is to set an inflection point to separate outliers from the other samples, and to clip the large gradients generated by outliers, setting the maximum value of the gradient to 1.0. Localization loss L_{loc} using balanced loss is defined as

$$L_{loc} = \sum_{i \in (x, y, h, w, \theta)} L_b(s_i^k - v_i). \quad (13)$$

Based on the above equation, we design an enhanced gradient equation as

$$\frac{\partial L_b}{\partial x} = \begin{cases} \alpha \ln(\gamma |x| + 1) & \text{if } |x| < 1 \\ \beta & \text{otherwise,} \end{cases} \quad (14)$$

The proposed balanced loss increases the gradients of inliers controlled by a factor called α , as shown in Figure 5. And a small α results in a greater increase in small gradients, while the gradients of outliers remain unaffected. In addition, β is a constant used to control the promotion magnification, which tunes the upper bound of regression gradients. γ is used to ensure balanced loss has the same value when $L_b(x = 1)$ for both formulations in Equations (15). Through such operations, we can increase the contribution of small-gradient samples to the training process and suppress the harmful effects caused by large gradients produced by outliers on the training process, thereby achieving a more balanced training within accurate classification and precise localization.

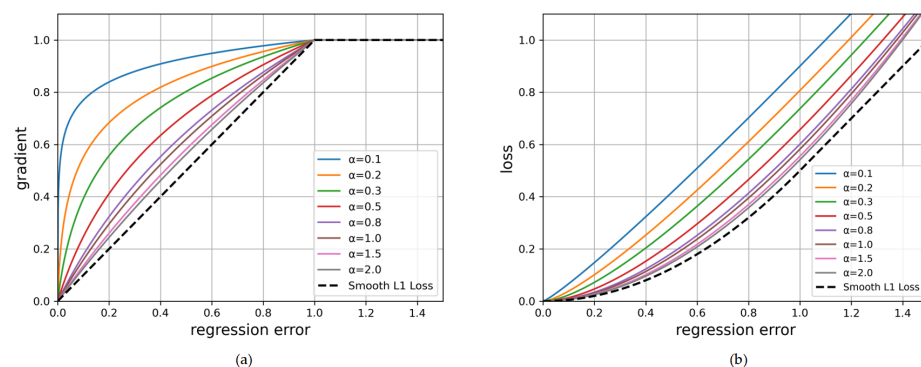


Figure 5. β in balanced loss is set by default as 1.0. (a) The gradient curves of our balanced loss with different α and the smooth L1 loss. (b) The loss curves of our balanced loss with different α and smooth L1 loss.

By integrating Equation (14), we can obtain the balanced loss:

$$L_b(x) = \begin{cases} \frac{\alpha}{\gamma}(\gamma |x| + 1) \ln(\gamma |x| + 1) - \alpha |x| & \text{if } |x| < 1 \\ \beta |x| + C & \text{otherwise.} \end{cases} \quad (15)$$

4. Experiments

4.1. Datasets

The UCAS-AOD [50] dataset is extensively used for remote sensing object detection, and the objects in the dataset are aircraft and vehicles under aerial images. Specifically, the dataset focuses on aircraft, including 600 images featuring 3210 aircraft, and the dataset for vehicles comprises 310 images, with 2819 vehicles included. Given the absence of an officially segmented version of the dataset, the entire dataset was randomly split into training, testing, and validation sets in a 6:2:2 ratio.

HRSC2016 [51] is a ship detection dataset released in 2016, comprising 1061 high-resolution images extracted from Google Earth. It covers six significant port scenes and is annotated using rotated bounding boxes. We randomly divided it into a training set (636 images), a testing set (212 images), and a validation set (213 images), including ships of various scales, shapes, and angles, both on sea and close inshore.

The DOTA [52] dataset is considered to be the largest public dataset for aerial target detection currently available, containing a total of 2806 aerial images from different sensors and platforms. The pixel sizes of these images range from 800×800 to 4000×4000 , and the objects in the images are characterized by rotation in multiple directions, significant scale differences, and dense distribution. It is noteworthy that most of the images in DOTA are too large. We resized all the original images to 1024×1024 for training and testing, and used random horizontal flipping during training to prevent overfitting.

4.2. Experimental Evaluation Metrics

It is necessary to comprehensively evaluate the performance of target localization and classification in the aerial object detection task. Due to the complexities of aerial images, which often contain multiple object categories and multiple instances of objects at various scales, the standard metrics for image classification are not directly applicable for evaluating the performance of aerial object detection. Consequently, mean average precision (mAP) has been proposed as a comprehensive and objective standard to measure the performance of detectors. Specifically, the precision and recall are defined as follows:

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TN + FN}, \quad (16)$$

where TP, FP, TN, FN are used to represent true positive, false positive, true negative, and false negative, respectively. For a high-performing detector, we aim for both high precision and high recall, so we need to consider both factors. By constructing a precision–recall curve and calculating the area under the curve, we obtain the AP as follows:

$$AP = \int_0^1 P(r) dr, \quad (17)$$

where $P(r)$ denotes the accuracy value corresponding to when the recall is r . AP ranges from 0 to 1, where higher values suggest superior detector performance. When there are N categories of targets, mAP is obtained by taking the average of the AP values of all categories:

$$mAP = \frac{\sum_{i=1}^N AP_i}{N}. \quad (18)$$

At the same time, we introduce FLOPs for a comprehensive evaluation of detector performance. The number of FLOPs (floating point operations) serves as a metric to assess the computational complexity in a model. The larger the amount of computation of FLOPs,

the higher the complexity of the model, and thus, more computational resources and time are required to execute it.

4.3. Implementation Details

The backbone of the proposed TFE-Net is ResNet. The threshold for matching positive samples was established at 0.5, and the detection head's confidence was determined to be 0.6. The feature pyramid of P3–P7 was utilized for detecting across multiple scales. The proposed TFE-Net is optimized by the Adam optimizer with momentum set to 0.9.

We trained the models for 100 epochs, starting with a learning rate of 0.01, which was then decreased by 0.1 after 40 epochs. In addition, the total batch size was set to 8 for training. All experiments were carried out on a server equipped with RTX 2080Ti based on the PyTorch framework. The versions of python and cuda were 3.8.10 and 11.7, respectively. The mAP and FLOPs were used as the experimental evaluation metrics. Throughout the training, testing, and validation phases of the experiment, we maintained consistent hyperparameter settings. We adopted RetinaNet as a baseline because it is a simple but effective one-stage detector, and the proposed modules and optimization designs are experimented on the basis of the baseline.

4.4. Comparisons with Other Methods on Different Datasets

4.4.1. Results on DOTA

We compared the proposed TFE-Net with other state-of-the-art methods on the DOTA dataset. The experimental results are shown in Table 1. For plane (PL), ship (SH), tennis court (TC), ground track field (GTF), harbor (HA), bridge (BR), and swimming pool (SP), our proposed TFE-Net achieved the best AP values: 90.32, 88.67, 90.93, 78.85, 78.61, 62.33, 82.91, respectively. It is also worth noting that we achieved the optimal results in terms of average accuracy across all categories with 79.25% mAP. The visualization of some detection results on the DOTA dataset are shown in Figure 6.

Table 1. Performance evaluation on the DOTA dataset.

Method	PL	SH	ST	BD	TC	BC	GTF	HA	BR	LV	SV	HC	RA	SBF	SP	mAP
SCRDet [53]	89.98	72.41	86.86	80.65	90.85	87.94	68.36	66.25	52.09	60.32	68.36	65.21	66.68	65.02	68.24	72.67
SCRDet++ [14]	90.01	78.83	85.10	82.32	90.86	86.32	68.62	77.69	61.94	81.17	69.92	64.25	61.12	65.10	80.68	76.24
RRPN [54]	88.52	57.25	67.38	71.20	90.81	72.84	59.30	53.08	31.66	56.19	51.85	53.85	52.84	56.69	51.94	61.01
CSL [55]	90.12	77.62	86.69	85.53	90.84	86.15	75.31	73.83	54.64	73.51	70.44	68.93	68.04	69.60	71.10	76.17
FR-O [16]	79.42	37.16	59.28	77.13	89.41	69.64	64.05	47.89	17.70	38.02	35.30	46.30	52.91	50.30	47.40	54.13
R ³ Det [56]	89.54	77.54	83.54	81.99	90.80	81.39	62.52	65.44	48.46	74.29	70.48	60.05	89.82	61.97	67.46	71.69
RoI-Trans [19]	88.64	83.59	81.46	78.52	90.74	77.27	75.92	62.83	43.44	73.68	68.81	47.67	53.54	58.39	58.93	69.56
RSDet [57]	89.80	70.20	83.40	82.90	90.50	85.60	65.20	65.60	48.60	70.10	69.50	68.00	63.90	62.50	67.20	72.20
CADNet [58]	87.80	76.60	73.30	82.40	90.90	79.20	73.50	62.00	49.40	63.50	71.10	62.20	60.90	48.40	67.00	69.90
RIDet-O [59]	88.94	88.18	83.61	78.45	90.55	81.33	72.63	73.09	46.87	80.68	77.63	56.87	63.72	64.85	73.13	74.70
Gliding Vertex [11]	89.64	86.82	86.81	85.00	90.74	79.02	77.34	72.94	52.26	73.14	73.01	57.32	70.91	59.55	70.86	75.02
R ² CNN [18]	80.94	55.81	72.39	65.67	90.67	66.92	67.44	55.14	35.34	50.91	59.92	48.22	52.23	55.06	53.35	60.67
TFE-Net (ours)	90.32	88.67	74.90	83.01	90.93	85.58	78.85	78.61	62.33	80.25	72.36	66.54	85.84	67.70	82.91	79.25
Explanation of each category:																
Full Name	plane	baseball diamond	bridge	ground track field	small vehicle	large vehicle	ship	tennis court	basketball court	storage tank	soccerball field	roundabout	harbor	swimming pool	helicopter	–

As can be seen from Table 1, the one-stage method R3Det only achieved 71.69% mAP on the DOTA dataset. While R3Det effectively achieves feature alignment through feature refinement module, it struggles to precisely extract the key features required for classification and regression tasks, thereby limiting the improvement in detection performance. Additionally, the two-stage method Gliding Vertex introduces more regression parameters

to enhance the representation of orientation information, resulting in better detection performance in multi-orientation object detection tasks, achieving 75.02% mAP on the DOTA dataset. However, this also increases the computational complexity of the model. Our TFE-Net can achieve feature alignment and extract orientation-invariant features for classification and orientation-sensitive features for regression, while also reducing computational complexity, thus achieving optimal detection performance.

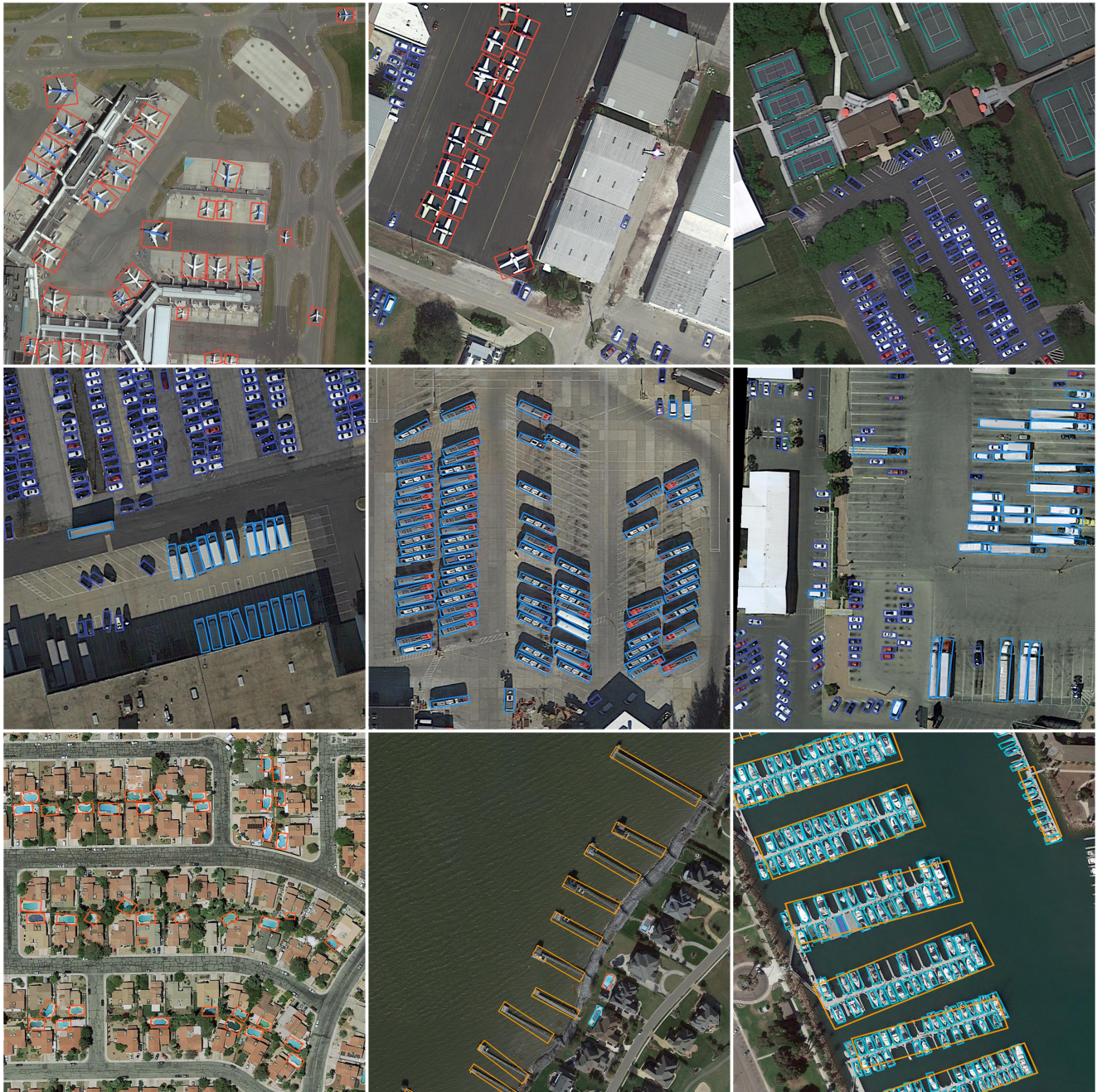


Figure 6. Visualization of results on DOTA dataset with TFE-Net.

For objects with arbitrary orientations (e.g., airplanes, ships), TFE-Net is able to obtain the orientation-sensitive features of the objects, and then, accurately align the orientations of these objects to achieve the adaptation to rotations. TFE-Net retains and effectively utilizes the edge texture features, which allows accurate detection of densely arranged small objects (e.g., small vehicles). Moreover, our proposed network can also achieve a

precise detection performance that is closer to the ground truth boxes of the objects with high aspect ratios (e.g., ships, bridges).

These experimental results show that our proposed TFE-Net achieves excellent detection performance, and also reflects that the key to achieving precise detection of remote sensing objects is that orientation-invariant features need to be extracted for the classification task in order to correctly classify targets in any direction, while orientation-sensitive features are extracted for the regression task in order to precisely localize the objects, and should be adapted to the characteristics of remote sensing objects for balanced training. Adaptive dynamic convolution is utilized in the task-sensitive detection module to align the features extracted by the backbone according to the orientation of the object to ensure the quality of the features, and further extract orientation-sensitive and orientation-invariant features from the high-quality aligned features to be applied to the regression and classification branches, and the balanced loss module ensures the balance of samples during the training process, which facilitates stable training of the network. The proposed three components work synergistically with each other to ensure detection performance.

4.4.2. Results on HRSC2016

As shown in Table 2, we conducted comparative experiments of TFE-Net with other detectors on the HRSC dataset. RRPN and R²CNN have relatively poor detection results when applied to oriented object detection, with mAP values of only 79.1% and 73.1%. RoI-Trans converts the horizontal RoI to a rotated one and achieves 86.2% mAP, but at the same time brings higher computational complexity. In comparison to specific ship detectors such as SDet and R³Det, our TFE-Net improves the mAP values by 1.0% and 0.9%, respectively. By adopting our proposed module, TFE-Net achieves outstanding detection performance in the experiments, and finally, realizes the highest 90.2% mAP.

Table 2. Performance evaluation on HRSC2016 dataset.

Method	Backbone	Size	mAP
RRPN [54]	ResNet101	800 × 800	79.1
R ² CNN [18]	ResNet101	800 × 800	73.1
RoI-Trans [19]	ResNet101	512 × 800	86.2
RRD [28]	VGG16	384 × 384	84.3
Gliding Vertex [11]	ResNet101	512 × 800	88.2
SDet [60]	ResNet101	800 × 800	89.2
R ³ Det [56]	ResNet101	800 × 800	89.3
TFE-Net (ours)	ResNet101	800 × 800	90.2

Some visual detection results of the proposed TFE-Net on HRSC2016 are shown in Figure 7. It can be observed that our TFE-Net effectively locates the boundaries of objects with high aspect ratios and achieves precise detection results. Specifically, in the first column of the figure, the ships are closely positioned next to each other, which often leads to missed detection or significant deviations between the bounding boxes and objects. Our TFE-Net is capable of accurately identifying the edges of each ship for more precise object localization. In the second column of the figure, ships are docked at the shore and piers, where these ships and the background share similar colors and textures. This similarity introduces severe background interference to object detection. TFE-Net is able to distinguish between the background and the ships effectively. In addition, the ships in the figure exhibit characteristics of arbitrary orientations. For objects with multiple rotation directions, our proposed network demonstrates strong rotational adaptability for multi-directional objects.

Additionally, we conducted comparative experiments between our proposed TFE-Net and the classical detector RetinaNet. As illustrated in Figure 8, our TFE-Net has superior detection performance for ships of various scales, and also achieves more compact localization for ships of any orientation. However, RetinaNet suffers from missed detections,

and it is difficult to precisely locate the boundaries of the ships, making the orientation of bounding boxes deviate from the object's actual orientation. These experimental results indicate that TFE-Net effectively captures the orientation information of objects, provides high-quality spatial features, and balances the gradients of different samples. This further improves the accuracy of bounding box regression, resulting in more precise and tight regression of object boundaries.

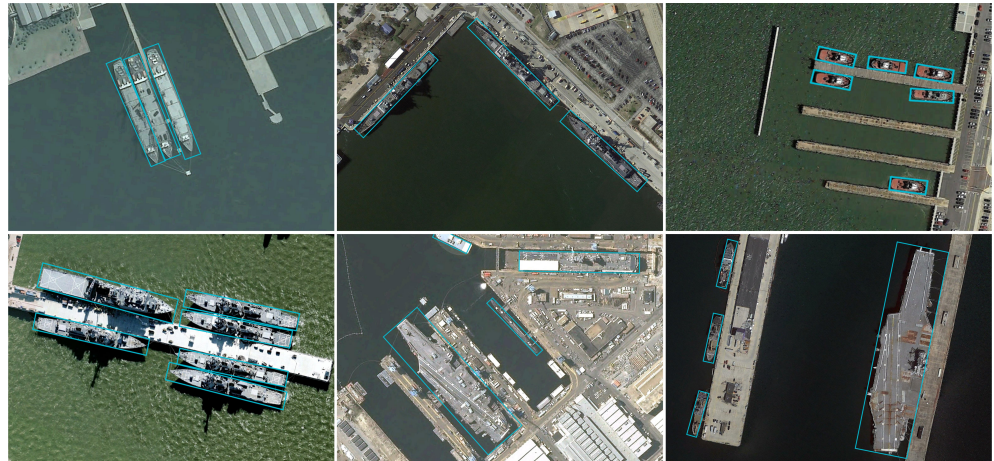


Figure 7. Performance on HRSC2016.

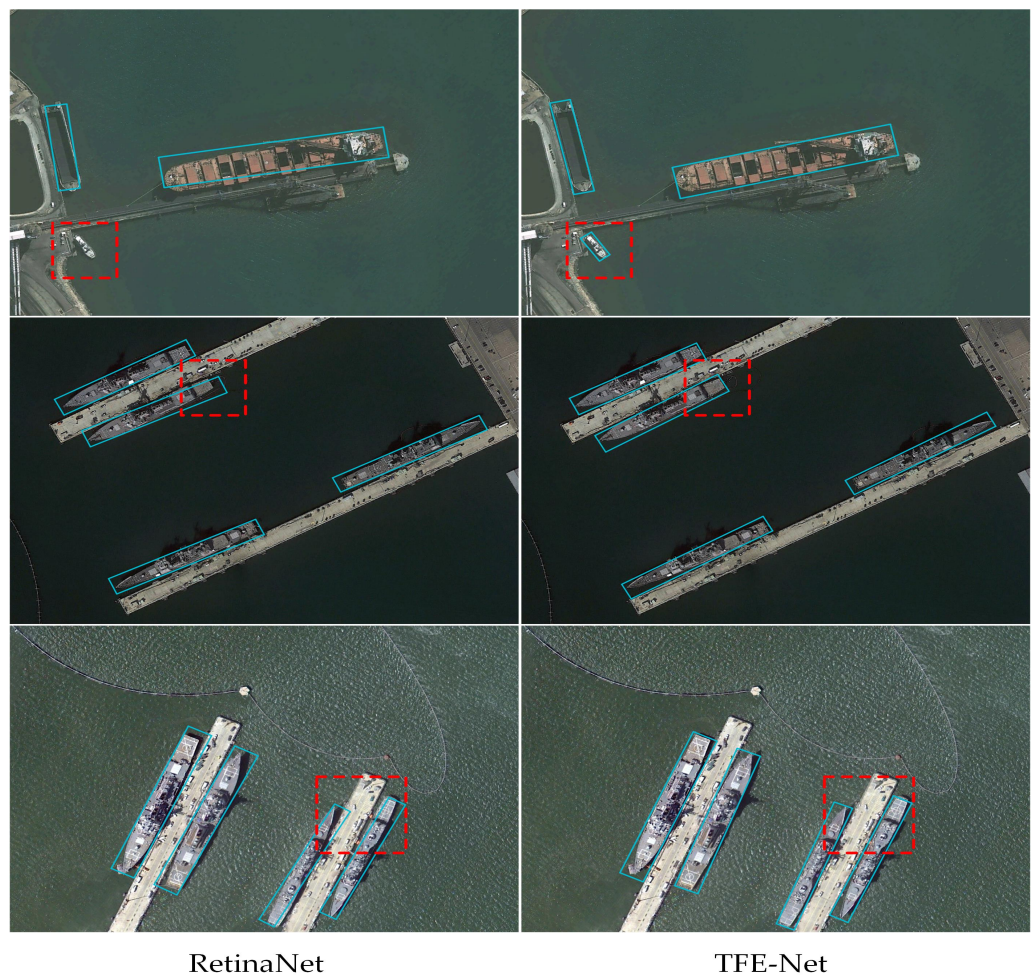


Figure 8. Comparison between TFE-Net and RetinaNet. The area marked by the red dashed box shows that TFE-Net outperforms RetinaNet.

4.4.3. Results on UCAS-AOD

We compared TFE-Net with other existing detectors on the UCAS-AOD dataset, and as can be seen from the results in Table 3, TFE-Net shows the best detection performance, achieving 90.86% mAP. This highlights the superior performance of our proposed network. As shown in Figures 9 and 10, for densely arranged aircraft and cars with arbitrary orientations in the UCAS-AOD dataset, TFE-Net is able to effectively distinguish different targets and achieve high-quality detection. We believe this is due to the task-sensitive detection module significantly enhancing the spatial representation capability of the features and effectively decoupling the features for regression and classification tasks. This allows for the extraction of critical features specific to each task, avoiding interference between features, and thus, further improving detection performance. However, as can be seen from Table 3, the proposed TFE-Net has lower detection accuracy for cars compared to airplanes, with 87.04% mAP and 94.68% mAP, respectively. This may be due to the fact that the construction of the backbone did not consider the impact of background context on detection performance, resulting in some occluded cars not being detected. In future work, we would aim to introduce a global attention mechanism or large-scale convolution kernel into the network to further enhance detection performance.

Table 3. Performance evaluation on UCAS-AOD dataset.

Method	Car	Airplane	mAP
R ² PN [32]	76.74	88.66	78.63
RoI-Trans [19]	88.02	90.02	89.02
S2ANet [61]	89.56	90.42	89.99
DRBox [34]	85.00	94.90	89.95
RIDet-O [62]	88.88	90.35	89.62
R-RetinaNet [63]	84.65	85.46	78.19
TFE-Net (ours)	87.04	94.68	90.86

4.5. Feature Visualization

We visualized the intermediate feature maps of the classification and regression branches within the task-sensitive detection module. As shown in Figure 11, the classifier accurately detects aircraft targets from the background and primarily extracts key features of the aircraft. Moreover, when the target undergoes rotation, the areas of strong response in the feature map equivalently rotate as well, demonstrating the ability to effectively extract key features for correct classification of the same type of target in any orientation. Compared to the classification branch, the area of strong response in the feature maps of the regression branch expands, and its direction is almost aligned with the orientation of the aircraft. It also focuses more on the edge texture information of the target to achieve compact positioning of the target boundaries.

4.6. Ablation Studies

4.6.1. Evaluation on Different Components

To verify the performance of the different modules proposed in the TFE-Net, we conducted relevant ablation experiments on the UCAS-AOD and HRSC2016 datasets. The experimental results on the UCAS-AOD dataset are shown in Table 4. '✓' and '✗' represent the use and non-use of the module, respectively.

Table 4. Experimental results of each component of TFE-Net on UCAS-AOD.

With SMFC?	With TDM?	With BLoss	FLOPs (G)	mAP
✗	✗	✗	137.8	84.2%
✓	✗	✗	71.2	85.5%
✗	✓	✗	158.0	89.1%
✓	✓	✗	109.6	88.4%
✗	✓	✓	163.5	91.3%
✓	✓	✓	97.6	90.9%



Figure 9. Visual detection results of airplanes on UCAS-AOD.

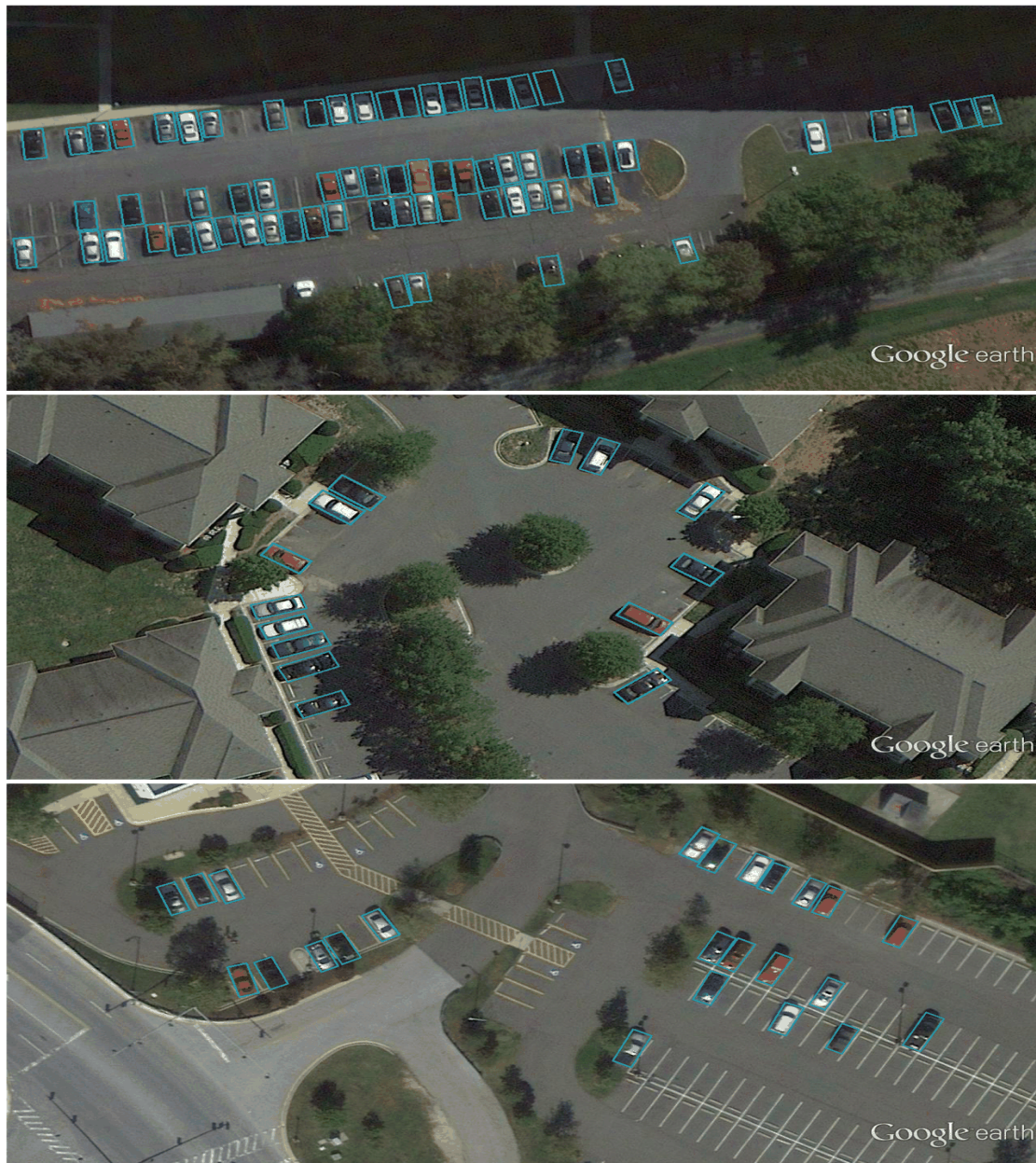


Figure 10. Visual detection results of vehicles on UCAS-AOD.

As can be seen from Table 4, due to the large amount of computation and redundant features in conventional CNN structures, which make it challenging to extract key features for precise detection of remote sensing targets, the baseline model only achieved 137.8 GFLOPs and an mAP of 84.2%. With the addition of SMFC, the detector is able to generate features in a simple and cheap way based on the redundancy of the feature map, which effectively reduces the computational complexity of the model. Specifically, the FLOPs change from 137.8 GFLOPs to 71.2 GFLOPs, which is nearly two times less. When TDM is added, there is a 4.9% optimization in the performance of the detector. This indicates that adaptive dynamic convolution effectively extracts the orientation information of the targets and enhances the network's adaptability to the orientation change of the target. And the orientation-sensitive features and orientation-invariant features extracted by using the task-sensitive feature decoupling mechanism are beneficial for regression and classification tasks. However, TDM also leads to an increase in the model's computational complexity. When TDM and BLoss are added, the optimization of the loss function solves the problem of sample gradient imbalance, making the training process more stable and

conductive to further improvements in detection performance, but also leads to an increase in the model's computational complexity. After further adding SMFC, the performance of the detector only slightly decreases, with the mAP decreasing from 91.3% to 90.9%. While maintaining high detection performance, a significant reduction in the computational complexity of the model is achieved, with FLOPs of only 97.6 GFLOPs lower than that of the baseline model, which is 137.8 GFLOPs. Therefore, this indicates that our proposed TFE-Net achieves an effective balance between speed and accuracy.

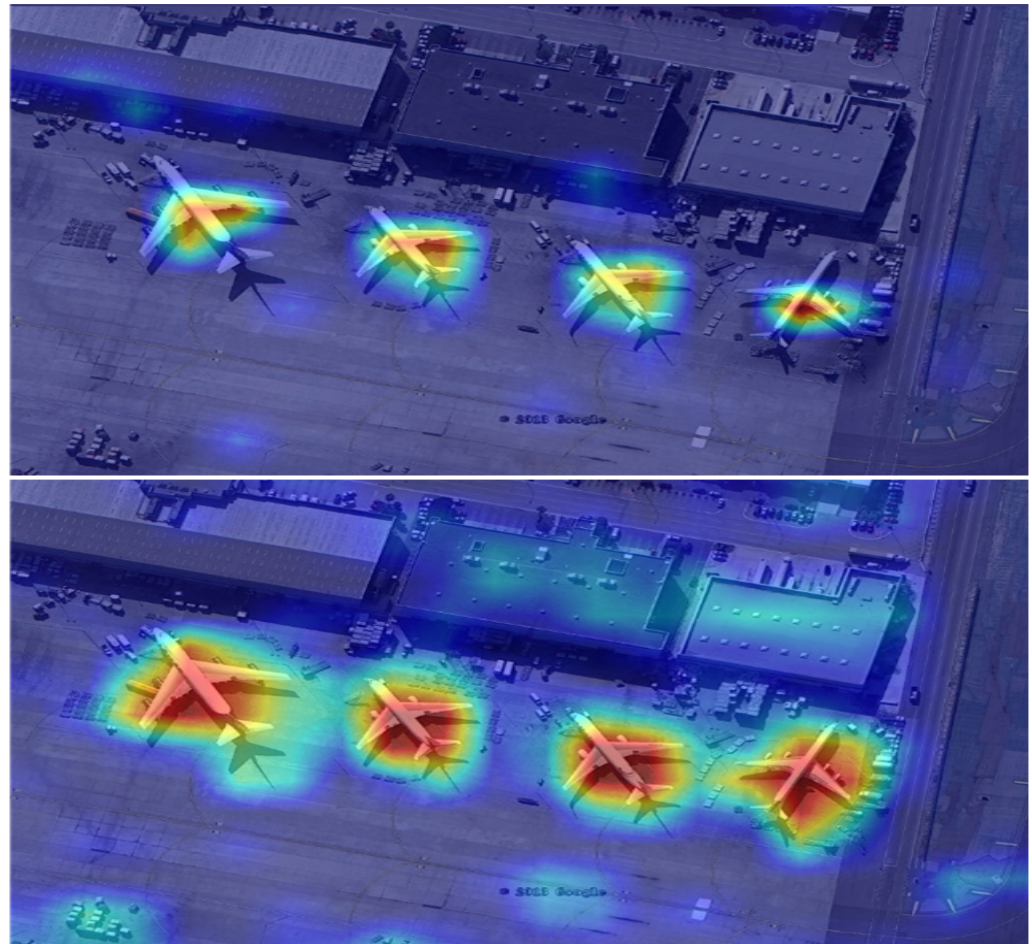


Figure 11. Visualization results of intermediate features.

Similar experimental results can be obtained on the HRSC2016 dataset, as shown in Table 5. The incorporation of the proposed modules allows the network to achieve better performance compared to the baseline model. The addition of SMFC is able to reduce both redundant computations and memory accesses for more efficient extraction of spatial features. Further, combining TDM and BLoss, more precise feature selection for specific tasks is conducted after feature alignment, while ensuring the balance of samples during the training process. This helps to further enhance the detection performance and achieve better classification and regression results. The experimental results also show that there is no conflict between the different modules proposed, and the detector exhibits the optimal performance of 90.2% mAP when all proposed modules are used.

Table 5. Experimental results of each component of TFE-Net on HRSC2016.

With SMFC?	With TDM?	With BLoss	FLOPs (G)	mAP
✗	✗	✗	137.6	82.0%
✓	✗	✗	71.2	83.1%
✗	✓	✗	157.8	89.6%
✓	✓	✗	109.5	89.0%
✗	✓	✓	163.2	90.8%
✓	✓	✓	97.3	90.2%

4.6.2. Evaluation on Special Mixed Fast Convolution

There are two hyperparameters in the special mixed fast convolution module, which are q for generating $n = m/q$ intrinsic features and the kernel size $l \times l$ of linear operations for calculating the extrinsic features. Currently, convolution operations can be efficiently implemented on hardware, and they can cover some widely used linear operations, such as sharpening, smoothing, and blurring, etc. At the same time, different linear operations can cause the module to have an irregular structure, which may lead to a decrease in computational efficiency. Therefore, we set l to a fixed value and use deep convolution to construct the SMFC module. As shown in Table 6, we performed the experiments on the HRSC2016 dataset, fixing $q = 2$ to test the impact of the hyperparameter l on detection performance.

Table 6. Evaluation of the SMFC module with different l on HRSC2016.

l	FLOPs (G)	mAP
1	96.8	88.4%
3	97.3	90.2%
5	97.7	89.6%
7	98.2	88.0%

From the results in Table 6, the detector achieves optimal detection performance when l takes the value of 3. This may be due to the fact that a convolutional kernel of size $l \times l$ cannot extract spatial features efficiently, resulting in the loss of some critical features; while a larger size convolutional kernel ($l = 5$ or $l = 7$) will introduce more computations, leading to increased computational complexity, and may also lead to overfitting. Therefore, taking into account both the accuracy and efficiency of the detector, we use $l = 3$ in all experiments in this paper.

After exploring the effect of the hyperparameter l on the detection performance in the SMFC module, we fixed $l = 3$ to tune the parameter q in $[2, 5]$. We tested the effect of the hyperparameter q on the detection performance and the experimental results are shown in Table 7.

It has been derived from Equations (4) and (5) that the computational efficiency of the network is closely linked to the value of q , meaning that a larger q can achieve a greater acceleration ratio and model compression ratio. According to the results in Table 7, the FLOPs decrease significantly with the increase in q , which is consistent with the results as expected. However, this is also accompanied by a decrease in detection accuracy, especially, when a larger q value is chosen, the detection accuracy is lower. In order to effectively reduce the FLOPs while avoiding excessive degradation of the detection accuracy, we choose $q = 3$, which shows notably better detection performance than the baseline model. These experiments demonstrate that our proposed SMFC module can successfully reduce the computational complexity of the model.

Table 7. Evaluation of the SMFC module with different q on HRSC2016.

q	FLOPs (G)	mAP
2	114.3	91.2%
3	97.3	90.2%
4	88.5	85.4%
5	78.9	81.2%
6	66.0	77.9%

4.6.3. Evaluation of Task-Sensitive Detection Module

We carried out some comparative experiments on the HRSC dataset in order to further validate the effectiveness of the TDM, as shown in Table 8. After feature alignment through adaptive dynamic convolution (ADC), the performance of the detector improved by 1.9%, indicating that feature alignment enhances the feature representation capability. The performance of the model is further improved by 3.3% when the task-sensitive feature decoupling mechanism (TFDM) is added, which indicates that extracting the key features required for classification and regression tasks from the shared features extracted from the backbone is very effective in improving the performance of the remote sensing object detector. The combination of ADC and TFDM aligns the features of the backbone according to the orientation of the objects, optimizes the feature representation, and further, extracts orientation-sensitive features and orientation-invariant features from the aligned high-quality features for specific tasks, which is beneficial for accurately identifying and locating objects. The combination of ADC and TFDM significantly improves the performance of the detector, with the mAP value rising from 84.5% to 90.2%, which affirms our perspective. These experimental results indicate that our proposed TDM can significantly improve the detection performance.

Table 8. Effects of each component of TDM on HRSC.

With ADC?	With TFDM?	mAP
✗	✗	84.5%
✓	✗	86.4%
✗	✓	87.8%
✓	✓	90.2%

4.6.4. Evaluation on Balanced Loss

To confirm the effectiveness of the proposed balanced loss, a comparative experiment is conducted with L1 loss on the HRSC2016 dataset and the optimal values of the two hyperparameters α and β in the balanced loss are explored. The experiment is shown in Table 9.

Table 9. Ablation studies of the proposed balanced loss on HRSC2016.

Settings	mAP
L1 loss	89.1%
$\alpha = 0.2, \beta = 1.0$	89.5%
$\alpha = 0.5, \beta = 1.0$	89.7%
$\alpha = 0.5, \beta = 1.5$	90.2%
$\alpha = 0.5, \beta = 2.0$	89.8%

The experiments demonstrate that the results using the proposed balanced loss are significantly better than using L1 loss. When β is set to a relatively large value, a slight decrease in mAP is observed. We speculate that this may be attributed to insufficient suppression of samples with large gradients. The optimal results are achieved when $\alpha = 1.0$ and $\beta = 1.5$.

5. Conclusions

In this paper, we propose a task-sensitive efficient feature extraction network to achieve faster and more precise lightweight remote sensing object detection, comprising three principal components: the special mixed fast convolution module (SMFC), task-sensitive detection module (TDM), and balanced loss function (BLoss). Specifically, the SMFC utilizes the redundancy of feature maps, replacing some convolution operations with cheap transformation operations, thereby reducing computational complexity while fully preserving spatial features. Subsequently, we use the TDM to firstly align the target with the features to enhance the spatial representation of the features, and further, extract orientation-invariant features and orientation-sensitive features from the high-quality aligned features to be fed into the classification and regression branches, respectively. Lastly, BLoss is introduced to effectively address the gradient imbalance issue caused by varying samples, thus stabilizing the network's training process. The ablation experiments and feature map visualization experiments verified the effectiveness of our proposed modules. Extensive comparative experiments on three commonly utilized public aerial image datasets indicate that our proposed TFE-Net can achieve excellent performance and obtain an effective balance between detection speed and accuracy. The proposed TFE-Net still has some limitations, such as not considering the impact of object scale and background context in the construction of the backbone. In future work, we would aim to introduce large kernel convolution or a global attention mechanism to enhance the network's ability to capture global dependencies, thereby further improving detection performance.

Author Contributions: Methodology, Z.L. and G.H.; Software, Z.L. and L.D.; Validation, D.J. and H.Z.; Investigation, L.D.; Writing—original draft, Z.L. and L.D.; Writing—review & editing, G.H. and H.Z.; Supervision, G.H. and D.J. All authors have read and agreed to the published version of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by two Aviation Science Foundations grant numbers 2023Z073053008 and D5120220246.

Data Availability Statement: The DOTA and UCAS-AOD are available at the following URLs: <https://captain-whu.github.io/DOTA/dataset.html> (accessed on 19 December 2023) and <https://github.com/Lbx2020/UCAS-AOD-dataset> (accessed on 20 December 2023), respectively.

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

References

- Huang, Z.; Li, W.; Xia, X.G.; Wang, H.; Jie, F.; Tao, R. LO-Det: Lightweight Oriented Object Detection in Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–15. [\[CrossRef\]](#)
- Zhao, B.; Zhao, B.; Tang, L.; Han, Y.; Wang, W. Deep spatial-temporal joint feature representation for video object detection. *Sensors* **2018**, *18*, 774. [\[CrossRef\]](#)
- Haq, M.A.; Rahim Khan, M.A. DNNBoT: Deep neural network-based botnet detection and classification. *Comput. Mater. Contin.* **2022**, *71*, 1729–1750.
- Merugu, S.; Tiwari, A.; Sharma, S.K. Spatial-spectral image classification with edge preserving method. *J. Indian Soc. Remote Sens.* **2021**, *49*, 703–711. [\[CrossRef\]](#)
- Haq, M. DBoTPM: A Deep Neural Network-Based Botnet Prediction Model. *Electronics* **2023**, *12*, 1159. [\[CrossRef\]](#)
- Merugu, S.; Jain, K.; Mittal, A.; Raman, B. Sub-scene target detection and recognition using deep learning convolution neural networks. In *Proceedings of the ICDSMLA 2019: Proceedings of the 1st International Conference on Data Science, Machine Learning and Applications*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 1082–1101.
- Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
- Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016*; pp. 779–788.
- Zhao, B.; Wang, Q.; Wu, Y.; Cao, Q.; Ran, Q. Target detection model distillation using feature transition and label registration for remote sensing imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 5416–5426. [\[CrossRef\]](#)

10. Wang, K.; Liew, J.H.; Zou, Y.; Zhou, D.; Feng, J. Panet: Few-shot image semantic segmentation with prototype alignment. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9197–9206.
11. Xu, Y.; Fu, M.; Wang, Q.; Wang, Y.; Chen, K.; Xia, G.S.; Bai, X. Gliding vertex on the horizontal bounding box for multi-oriented object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 1452–1459. [\[CrossRef\]](#)
12. Wang, J.; Yang, W.; Li, H.C.; Zhang, H.; Xia, G.S. Learning center probability map for detecting objects in aerial images. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 4307–4323. [\[CrossRef\]](#)
13. Yang, X.; Hou, L.; Zhou, Y.; Wang, W.; Yan, J. Dense label encoding for boundary discontinuity free rotation detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15819–15829.
14. Yang, X.; Yan, J.; Liao, W.; Yang, X.; Tang, J.; He, T. Scrdet++: Detecting small, cluttered and rotated objects via instance-level feature denoising and rotation loss smoothing. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 2384–2399. [\[CrossRef\]](#)
15. Qiu, H.; Li, H.; Wu, Q.; Meng, F.; Ngan, K.N.; Shi, H. A2RMNet: Adaptively aspect ratio multi-scale network for object detection in remote sensing images. *Remote Sens.* **2019**, *11*, 1594. [\[CrossRef\]](#)
16. 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*. [\[CrossRef\]](#)
17. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
18. Jiang, Y.; Zhu, X.; Wang, X.; Yang, S.; Li, W.; Wang, H.; Fu, P.; Luo, Z. R2CNN: Rotational region CNN for orientation robust scene text detection. *arXiv* **2017**, arXiv:1706.09579.
19. 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2849–2858.
20. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
21. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2818–2826.
22. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
23. Han, S.; Pool, J.; Tran, J.; Dally, W. Learning both weights and connections for efficient neural network. *Adv. Neural Inf. Process. Syst.* **2015**, *28*.
24. Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; Zhang, C. Learning efficient convolutional networks through network slimming. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2736–2744.
25. Wu, Y.; Chen, Y.; Yuan, L.; Liu, Z.; Wang, L.; Li, H.; Fu, Y. Rethinking classification and localization for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10186–10195.
26. Jiang, B.; Luo, R.; Mao, J.; Xiao, T.; Jiang, Y. Acquisition of localization confidence for accurate object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 784–799.
27. Ming, Q.; Miao, L.; Zhou, Z.; Dong, Y. CFC-Net: A critical feature capturing network for arbitrary-oriented object detection in remote-sensing images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–14. [\[CrossRef\]](#)
28. Liao, M.; Zhu, Z.; Shi, B.; Xia, G.S.; Bai, X. Rotation-sensitive regression for oriented scene text detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5909–5918.
29. Fu, K.; Chang, Z.; Zhang, Y.; Xu, G.; Zhang, K.; Sun, X. Rotation-aware and multi-scale convolutional neural network for object detection in remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2020**, *161*, 294–308. [\[CrossRef\]](#)
30. Cheng, G.; Zhou, P.; Han, J. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 7405–7415. [\[CrossRef\]](#)
31. Pan, X.; Ren, Y.; Sheng, K.; Dong, W.; Yuan, H.; Guo, X.; Ma, C.; Xu, C. Dynamic refinement network for oriented and densely packed object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11207–11216.
32. Zhang, Z.; Guo, W.; Zhu, S.; Yu, W. Toward arbitrary-oriented ship detection with rotated region proposal and discrimination networks. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1745–1749. [\[CrossRef\]](#)
33. Han, J.; Ding, J.; Xue, N.; Xia, G.S. Redet: A rotation-equivariant detector for aerial object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2786–2795.
34. Liu, L.; Pan, Z.; Lei, B. Learning a rotation invariant detector with rotatable bounding box. *arXiv* **2017**, arXiv:1711.09405.
35. Wang, J.; Chen, K.; Yang, S.; Loy, C.C.; Lin, D. Region proposal by guided anchoring. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2965–2974.
36. Hu, J.; Lin, P.; Zhang, H.; Lan, Z.; Chen, W.; Xie, K.; Chen, S.; Wang, H.; Chang, S. A Dynamic Pruning Method on Multiple Sparse Structures in Deep Neural Networks. *IEEE Access* **2023**, *11*, 38448–38457. [\[CrossRef\]](#)

37. Lee, J.R.; Moon, Y.H. An Empirical Study on Channel Pruning through Residual Connections. In Proceedings of the 2021 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 20–22 October 2021; pp. 1380–1382.
38. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2704–2713.
39. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
40. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
41. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
42. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
43. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
44. Wang, N.; Li, B.; Wei, X.; Wang, Y.; Yan, H. Ship detection in spaceborne infrared image based on lightweight CNN and multisource feature cascade decision. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 4324–4339. [[CrossRef](#)]
45. Mehta, S.; Rastegari, M. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv* **2021**, arXiv:2110.02178.
46. Chen, S.; Xie, E.; Ge, C.; Chen, R.; Liang, D.; Luo, P. CycleMLP: A MLP-Like Architecture for Dense Visual Predictions. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 14284–14300. [[CrossRef](#)] [[PubMed](#)]
47. Yang, B.; Bender, G.; Le, Q.V.; Ngiam, J. Condconv: Conditionally parameterized convolutions for efficient inference. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
48. Zhou, Y.; Ye, Q.; Qiu, Q.; Jiao, J. Oriented response networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 519–528.
49. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
50. Zhu, H.; Chen, X.; Dai, W.; Fu, K.; Ye, Q.; Jiao, J. Orientation robust object detection in aerial images using deep convolutional neural network. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 3735–3739.
51. Liu, Z.; Yuan, L.; Weng, L.; Yang, Y. A high resolution optical satellite image dataset for ship recognition and some new baselines. In Proceedings of the International Conference on Pattern Recognition Applications and Methods, SciTePress, Porto, Portugal, 24–26 February 2017; Volume 2, pp. 324–331.
52. 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 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983.
53. Yang, X.; Yang, J.; Yan, J.; Zhang, Y.; Zhang, T.; Guo, Z.; Sun, X.; Fu, K. Scrnet: Towards more robust detection for small, cluttered and rotated objects. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8232–8241.
54. 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](#)]
55. Yang, X.; Yan, J. On the arbitrary-oriented object detection: Classification based approaches revisited. *Int. J. Comput. Vis.* **2022**, *130*, 1340–1365. [[CrossRef](#)]
56. Yang, X.; Yan, J.; Feng, Z.; He, T. R3det: Refined single-stage detector with feature refinement for rotating object. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; Volume 35, pp. 3163–3171.
57. Qian, W.; Yang, X.; Peng, S.; Yan, J.; Guo, Y. Learning modulated loss for rotated object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; Volume 35, pp. 2458–2466.
58. 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](#)]
59. Ming, Q.; Miao, L.; Zhou, Z.; Yang, X.; Dong, Y. Optimization for arbitrary-oriented object detection via representation invariance loss. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [[CrossRef](#)]
60. Ren, Z.; Tang, Y.; He, Z.; Tian, L.; Yang, Y.; Zhang, W. Ship detection in high-resolution optical remote sensing images aided by saliency information. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–16. [[CrossRef](#)]
61. Han, J.; Ding, J.; Li, J.; Xia, G.S. Align deep features for oriented object detection. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–11. [[CrossRef](#)]

62. Shu, Z.; Hu, X.; Sun, J. Center-point-guided proposal generation for detection of small and dense buildings in aerial imagery. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1100–1104. [[CrossRef](#)]
63. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.

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.