Tiny Object Detection in High-Resolution Satellite Imagery via Oriented R-CNN with Dilated Fusion, Balanced Ranking Assignment, and Vector Mapping

by

J.A.H. van Oosten

In partial fulfilment of the requirements for the degree of
Master of Science
at Delft University of Technology,
to be defended publicly on 16 December 2024.

Faculty:	Aerospace Engineering
Department:	Control & Operations
Programme:	Control & Simulation

Mentors / Supervisors:	Dr. A. Jamshidnejad
Graduation committee:	Dr. E.J.J. Smeur Dr. A. Jamshidnejad Dr. S. Khademi

Keywords:

Tiny Object Detection
Machine Learning
Computer Vision
Convolutional Neural Networks
Remote Sensing
Satellite Imagery
Critical Infrastructure Monitoring

Abstract:

Tiny object detection (TOD) in satellite imagery is critical for applications including pipeline monitoring, where the detection of tiny objects, such as excavators near the pipeline networks, can prevent potential incidents. However, TOD faces challenges due to the limited pixel representation of objects, complications with Intersection over Union-based label assignment, and semantic confusion between visually similar classes. This paper introduces OE-Net, a modular extension of Oriented R-CNN, specifically designed for TOD in satellite images. OE-Net integrates three novel components: a Dilated Fusion module to enhance the feature extraction, a Balanced Ranking Assigner to improve the anchor matching, and Vector Mapping for more precise classification of visually similar objects. The modular design of OE-Net allows its novel components to be easily integrated into other detection models. Tested on the TinyDOTA dataset, OE-Net sets a new benchmark in TOD performance, achieving a 6.3 percentage points improvement in mean Average Precision (mAP) over the Oriented R-CNN baseline. On a novel excavator detection dataset, named ExcaSat and developed for pipeline monitoring, OE-Net outperforms the Oriented R-CNN baseline by 11.1 percentage points in mAP. Furthermore, OE-Net surpasses state-of-the-art methods on both TinyDOTA and ExcaSat, establishing a new standard in computational efficiency and detection precision for satellite-based monitoring tasks. The code is available at: https://github.com/OE-jimvanoosten/mmrotate-OE.git.

Highlights

Tiny Object Detection in High-Resolution Satellite Imagery via Oriented R-CNN with Dilated Fusion, Balanced Ranking Assignment, and Vector Mapping.

J.A.H. van Oosten

- Benchmarking twelve common state-of-the-art oriented object detectors on the tiny object detection dataset TinyDOTA and proposing a novel excavator detection dataset named ExcaSat.
- Developing and validating a Dilated Fusion module that extends the Feature Pyramid Network to improve the spatial context by integrating deeper, more comprehensive features from the ResNet-50 backbone.
- Developing and validating a Balanced Ranking Assigner that evaluates anchors using the Normalized Wasserstein Distance and incorporates an objectness score into the ranking metric, ensuring that anchors are assigned based on both their spatial proximity to the ground truth boxes and their likelihood of containing an object.
- Developing and validating a novel vector-based classification approach named Vector Mapping that exploits the adaptive feature extraction capabilities of baseline detectors to mitigate the semantic confusion between object instances and background instances that have highly similar feature representations.
- Introducing OE-Net, which enhances Oriented R-CNN by integrating Dilated Fusion, Balanced Ranking Assigning, and Vector Mapping, comparing its performance with the TinyDOTA benchmark, and validating its generalizability on ExcaNet.

Tiny Object Detection in High-Resolution Satellite Imagery via Oriented R-CNN with Dilated Fusion, Balanced Ranking Assignment, and Vector Mapping.

J.A.H. van Oosten

[a]Technical University of Delft, Kluyverweg 1, 2629 HS, Delft, The Netherlands
[b]Orbital Eye, Olof Palmestraat 14, 2616 LR, Delft, The Netherlands

NOMENCLATURE

AP	Average Precision
BRA	Balanced Ranking Assigner
CNN	Convolutional Neural Network
DF	Dilated Fusion
FPN	Feature Pyramid Network
IoU	Intersection over Union
mAP	mean Average Precision
NMS	Non-Maximum Suppression
NWD	Normalized Wasserstein Distance
RPN	Region Proposal Network
RoI	Region of Interest
TOD	Tiny Object Detection
VM	Vector Mapping

ABSTRACT

Tiny object detection (TOD) in satellite imagery is critical for applications including pipeline monitoring, where the detection of tiny objects, such as excavators near the pipeline networks, can prevent potential incidents. However, TOD faces challenges due to the limited pixel representation of objects, complications with Intersection over Union-based label assignment, and semantic confusion between visually similar classes. This paper introduces OE-Net, a modular extension of Oriented R-CNN, specifically designed for TOD in satellite images. OE-Net integrates three novel components: a Dilated Fusion module to enhance the feature extraction, a Balanced Ranking Assigner to improve the anchor matching, and Vector Mapping for more precise classification of visually similar objects. The modular design of OE-Net allows its novel components to be easily integrated into other detection models. Tested on the TinyDOTA dataset, OE-Net sets a new benchmark in TOD performance, achieving a 6.3 percentage points improvement in mean Average Precision (mAP) over the Oriented R-CNN baseline. On a novel excavator detection dataset, named ExcaSat and developed for pipeline monitoring, OE-Net outperforms the Oriented R-CNN baseline by 11.1 percentage points in mAP. Furthermore, OE-Net surpasses state-of-the-art methods on both TinyDOTA and ExcaSat, establishing a new standard in computational efficiency and detection precision for satellite-based monitoring tasks. The code is available at: `https://github.com/OE-jimvanoosten/mmrotate-OE.git`.

1. Introduction

The deployment and the utilization of earth observation satellites has paved the way for research into remote sensing and its applications. Tiny object detection (TOD) in satellite images has become increasingly relevant for various high-stakes applications, such as monitoring critical infrastructure, disaster response, and urban development (Cheng and Han, 2016). One practical example is in pipeline monitoring, where accurate detection of small objects, particularly vehicles such as excavators, in the vicinity of the pipeline network is essential. Detecting these objects can inform real-time surveillance and prevent incidents.

However, TOD poses significant challenges due to the unique characteristics of satellite imagery. The objects in high-resolution satellite images are often reduced to only a few pixels, making it difficult to capture meaningful feature representations. Unlike traditional object detection methods designed for everyday images, satellite imagery provides a top-down perspective that lacks depth perception and compresses the object features, resulting in highly condensed and less distinguishable details.

This results in three primary issues in TOD: poor feature extraction from tiny objects, as conventional methods are typically designed for larger objects that have richer feature representations (Singh and Davis, 2018); a mismatch between the predefined anchor boxes and the tiny objects, which typically reduces the quality of the label assignment and the region proposals (Zhang et al., 2024); and frequent semantic confusion in the classification stage, where visually similar objects, such as road surface markings and vehicles, become challenging to distinguish due to the minimal differences in the feature representations (Zhou et al., 2022a). These constraints underline the need for specialized TOD techniques for satellite imagery that go beyond the general object detection frameworks.

To address these challenges, this paper proposes OE-Net, an extension of the baseline Oriented R-CNN architecture (Xie et al., 2021). OE-Net incorporates a Dilated Fusion module, a Balanced Ranking Assigner, and a Vector Mapping module to enhance TOD precision in satellite imagery. Without bells and whistles, OE-Net outperforms the novel state-of-the-art benchmark on TinyDOTA, improving the mean Average Precision (mAP) of the baseline Oriented R-CNN detector by 5.9 percentage points. Furthermore, OE-Net surpasses the second-best performing model on TinyDOTA by 1.5 percentage points in mAP, while requiring only half the training time. The generalization capabilities of OE-Net are validated on a novel excavator dataset for pipeline monitoring, named ExcaSat, where OE-Net outperforms other state-of-the-art detection methods. The contributions of this work are summarized as follows:

- Benchmarking twelve common state-of-the-art oriented object detectors on the tiny object detection dataset TinyDOTA and proposing a novel excavator detection dataset named ExcaSat.

- Developing and validating a Dilated Fusion module that extends the Feature Pyramid Network to improve the spatial context by integrating deeper, more comprehensive features from the ResNet-50 backbone.
- Developing and validating a Balanced Ranking Assigner that evaluates anchors using the Normalized Wasserstein Distance and incorporates an objectness score into the ranking metric, ensuring that anchors are assigned based on both their spatial proximity to the ground truth boxes and their likelihood of containing an object.
- Developing and validating a novel vector-based classification approach named Vector Mapping that exploits the adaptive feature extraction capabilities of baseline detectors to mitigate the semantic confusion between object instances and background instances that have highly similar feature representations.
- Introducing OE-Net, which enhances Oriented R-CNN by integrating Dilated Fusion, Balanced Ranking Assigning, and Vector Mapping, comparing its performance with the TinyDOTA benchmark, and validating its generalizability on ExcaNet.

The Related Work section provides an overview of object detection using convolutional neural networks (CNNs). It highlights the challenges and advances in TOD and vehicle detection, and it discusses the three key challenges in TOD that OE-Net addresses. The Methodology section describes the architecture of the novel modules that are integrated into OE-Net. The Experiment section explains the datasets, the evaluation metrics, and the implementation details of the experiments. It presents the results of the comparative study of OE-Net and other state-of-the-art methods on TinyDOTA and ExcaSat. It includes the results of the ablation studies, an analysis of the hyperparameters of OE-Net, and a discussion of the observed outcomes of the experiments. The Conclusion section summarizes the key findings and implications of this paper, Furthermore, it discusses the limitations of OE-Net, and it proposes directions for future research.

2. Related Work

The following section provides an overview of the key advancements in object detection using CNNs. It focuses on TOD in satellite imagery, and explores methods for vehicle detection in satellite imagery. Additionally, the section discusses the challenges of feature fusion techniques, anchor assignment methods, and classification approaches that OE-Net addresses to improve the detection precision for TOD in satellite imagery.

2.1. Object Detection with CNNs

CNNs excel in object detection by learning the spatial hierarchies of the features in images. Convolutional layers, the core of CNNs, apply filters to the input images to produce feature maps that highlight patterns such as edges, textures, and shapes. By sliding the filters across the image and computing the dot products with the overlapping regions, CNNs capture the spatial hierarchies. During the training process, the filter weights are learned, enabling the network to identify the task-relevant features adaptively (LeCun et al., 2015). Stacking multiple convolutional layers allows CNNs to extract increasingly complex features. The initial layers detect simple patterns such as edges, while the deeper layers identify intricate shapes and object components (Krizhevsky et al., 2012; Zeiler and Fergus, 2014).

In object detection pipelines, CNNs are integrated through a multi-stage architecture comprising the backbone, the neck, and the detection heads. The backbone, typically a pre-trained CNN, extracts hierarchical features from the input image using convolutional layers (Dhillon and Verma, 2020). The neck further processes these features, employing convolutions to adjust the dimensions of the features and to enhance the fusion of the features for multiscale detection (Zhiqiang and Jun, 2017). The proposal head generates candidate regions by applying convolutions to predict a set of object proposals. These proposals are areas that are likely to contain objects. The detection head receives these proposals, and passes them through additional convolutions and fully connected layers to classify the objects and refine the predicted bounding boxes (Liu et al., 2021).

Anchor-based detectors use predefined anchor boxes to detect objects within an image (Ren et al., 2016; Lin et al., 2017a; Cai and Vasconcelos, 2018; Li et al., 2019b; Liu et al., 2016; Redmon et al., 2016; Lin et al., 2017b). These anchors are distributed across the image at various scales and aspect ratios, and act as initial guesses for the object locations. Essentially, the anchors approximate objects of various sizes and shapes. Anchor boxes provide a starting point for the detection process. CNNs refine these anchors by predicting the translation offsets and the aspect ratio offsets (Zhong et al., 2020). By comparing the bounding boxes predicted by the model with the ground truth bounding boxes, the model gradually learns to localize and classify the objects. Anchor-based detectors are categorized into two types:

- **Two-Stage Detectors:** Two-stage detectors use two steps to refine the anchors, and to refine the proposals that are generated from the anchors. The original R-CNN (Girshick et al., 2014) uses selective search (Uijlings et al., 2013) to generate the regions of interest (RoIs) that are passed through a CNN to extract a set of regional features. These features are classified using a linear support vector machine (Cortes and Vapnik, 1995). Fast R-CNN (Girshick, 2015) and SPP-net (He et al., 2015) use RoI pooling to enhance the speed and the accuracy of the original R-CNN. Faster R-CNN (Ren et al., 2016) is the first model to introduce the Region Proposal Network (RPN), generating RoIs that are refined and classified in a second detection head downstream. While two-stage detectors generally achieve a higher detection precision than one-stage detectors, the added number of trainable parameters results in a lower inference speed and a higher computational load (Kang et al., 2024).

- **One-Stage Detectors:** One-stage detectors integrate the detection process into a single step. Models such as YOLO (Redmon et al., 2016) and SSD (Liu et al., 2016) predict the class probabilities and the bounding box coordinates directly from the predefined anchor boxes using a single forward pass. This results in faster inference speeds compared to two-stage detectors, but generally a slightly lower detection precision.

2.2. Tiny Object Detection (TOD)

Satellite images have spatial resolutions that vary between 30cm and 10m. Therefore, many normal-sized objects, such as cars, trucks, and excavators, appear as tiny objects. These objects blend into their surroundings because they occupy a very limited number of pixels in the image. Many object detection frameworks struggle with tiny objects (Singh et al., 2018; Wang et al., 2021; Cai and Vasconcelos, 2018; Yu et al., 2020), resulting in a significantly lower detection precision in both low-resolution (Pang et al., 2019) and high-resolution satellite images (Shermeyer and Van Etten, 2019). This challenge of TOD is more noticeable in satellite images than in everyday images. In satellite images, conventional detectors (Ren et al., 2016; Lin et al., 2017b; Liu et al., 2016; Redmon et al., 2016) show a significant drop in detection precision.

Recent studies focus on enhancing the feature discrimination capabilities of the detector through methods such as Generative Adversarial Networks (GANs) (Li et al., 2017; Bai et al., 2018), super-resolution techniques (Noh et al., 2019; Dong et al., 2015; Kim et al., 2016), and feature pyramids (Lin et al., 2017a; Zhao et al., 2019; Qiao et al., 2021). Feature pyramids construct a hierarchy of the image at different scales, combining low-level details with high-level semantic features to detect objects of various sizes. GANs improve TOD by using super-resolution generators to upscale the low-resolution images, recovering fine details and thereby improving the detection precision. Other studies focus on enhancing the resolution by normalizing the scales of the input images, instead of directly improving the feature discrimination capability of the detector itself (Singh and Davis, 2018; Singh et al., 2018).

Several TOD studies introduce modules designed to enhance existing object detectors (Lin et al., 2017a; Ding et al., 2019; Xu et al., 2021). While conventional object detectors excel on everyday object datasets with axis-aligned bounding boxes, satellite imagery often contains multi-oriented objects. As a result, modeling these objects with rotated bounding boxes significantly improves the detection precision. State-of-the-art detectors for satellite imagery predominantly utilize rotated bounding boxes, as these are more effective for capturing objects with arbitrary orientations (Li et al., 2023; Pu et al., 2023; Xu et al., 2023).

Ding et al. (2019) rotate the predefined horizontal anchor boxes using fully connected layers, and then retrieve the features within these boxes for further regression and classification. To improve the modeling of crowded objects and reduce the background noise, Yang et al. (2019) use attention mechanisms that suppress background noise and emphasize the foreground objects. To address the instability caused by the periodicity of the rotation angles of the objects, novel box encoding algorithms are introduced in Oriented R-CNN (Xie et al., 2021) and Gliding Vertex (Xu et al., 2020). Instead of optimizing anchor-based two-stage methods for satellite imagery, one-stage models directly classify and regress a set of oriented bounding boxes from densely sampled grid anchors. The one-stage S2A-Net (Han et al., 2021a) enhances the feature extraction through oriented feature alignment, yielding orientation-invariant features.

Another popular approach is to add a transformer-based detector to the existing CNN-based feature extractors (Dai et al., 2023; Carion et al., 2020; Li et al., 2024b; Lei et al., 2021). The introduction of transformers to computer vision has led to the formation of foundational models that are extensively trained using large and diverse datasets. This makes the trained neural network more robust to variations in the input data. The neural network can then be fine-tuned for specific downstream tasks, such as for TOD in satellite imagery (Wang et al., 2022).

2.3. Vehicle Detection in Satellite Imagery

Various recent studies have focused on the task of vehicle detection in satellite imagery. Ji et al. (2019) use GANs to enable super-resolution CNNs that upsample the images such that the feature representations of the vehicles are enhanced. Tayara et al. (2017) employ a set of CNNs to generate a spatial density map of the input images, and then use a set of fully connected layers to regress the vehicle locations. Yu and Shi (2015) convert the satellite images into a hyperspectral format, and then use a hyperspectral algorithm to detect the vehicles. Li et al. (2019a) detect vehicles with multiple orientations by generating rotatable rectangular bounding boxes. These transformations enhance the features and the feature alignment of the vehicles, and thereby improve the distinction between the vehicles and the background noise.

2.4. Feature Fusion for TOD

In TOD, and particularly in satellite imagery, a major challenge arises from the limited availability of distinctive features in objects. Tiny objects such as cars, excavators, and boats often appear with blurred or indistinct features due to their low resolution. This makes them difficult to differentiate from complex backgrounds. The characteristic elements of these objects, such as their windshields or crane arms, become less defined as the spatial resolution decreases. Additionally, tiny objects span only a few pixels, which causes their features to be easily overshadowed by and merged with the background noise in the image. Figure 1 illustrates the limited feature representation and the pixel density of tiny objects in satellite imagery.

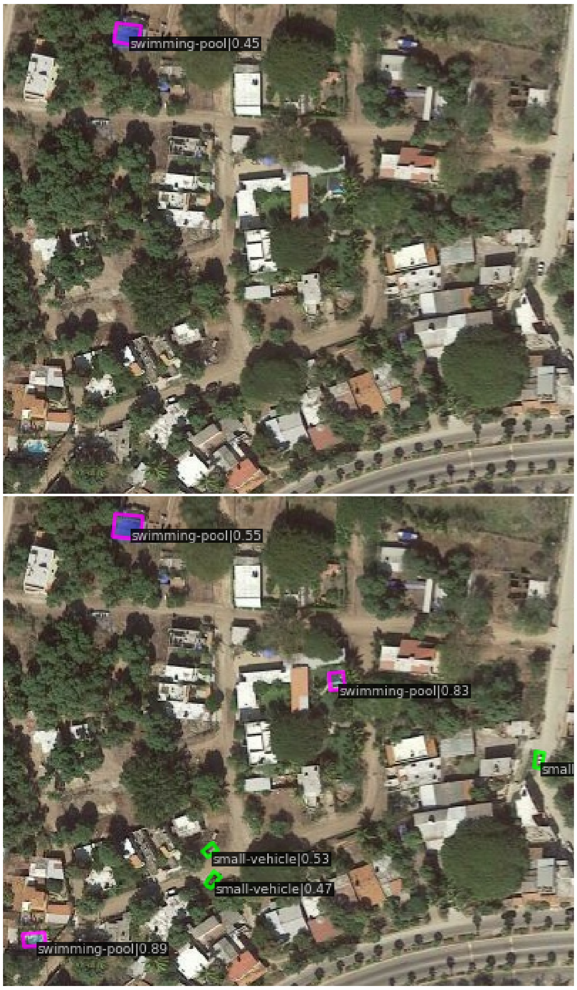

Figure 1: Oriented R-CNN (top) versus OE-Net (bottom). The inference results on an image from the TinyDOTA test dataset. Each bounding box represents a specific class in the dataset and shows the confidence level of the prediction.

In traditional object detectors, feature extraction is usually performed by a pre-trained backbone that captures information across multiple levels of the image. However, these backbones are mainly designed for image classification and are not specifically optimized for detecting small objects at different scales. As a result, they do not adequately capture the fine-grained details necessary for effective TOD. To address this, a Feature Pyramid Network (FPN) is commonly integrated into the detection framework (Lin et al., 2017a). FPNs create a feature pyramid by combining the feature maps from the backbone network. They start at the deepest level and progressively work downward, utilizing lateral connections. This method enables the propagation of semantically rich, high-level features down to lower-resolution layers, where they are combined with higher-resolution, low-level details. The resulting feature maps improve the representation of objects across different scales, and help to capture and preserve both detailed and abstract features.

Typically, the deepest feature map from the backbone undergoes a lateral convolution to produce the highest feature map in the top-down pathway of the FPN. In contrast, the lower feature maps in the FPN hierarchy are produced by combining the lateral convolution of the current feature map with the upsampled version of the previous feature map. For example, the feature map at a given layer is generated in two steps: first, the feature map from the layer above is upsampled to match the resolution of the current layer; then, this upsampled feature map is added to the output of a lateral convolution applied to the current feature map. This pattern is repeated at each successive layer, combining higher-level semantic information from the previous layer with spatial details from the current layer. This approach enhances the ability of the feature map to represent objects at different scales. However, the top-level feature map of the FPN is generated solely from the deepest feature map from the backbone without such contextual enhancement.

Since the top-level feature map within the FPN undergoes minimal enhancement and lacks additional refinement, FPNs encounter challenges in TOD within satellite imagery. This leads to a loss of localized, fine-grained contextual information that is essential for identifying tiny objects (Wen et al., 2023). In satellite imagery, the absence of such detailed cues results in a noticeable drop in detection accuracy, as crucial spatial information is not adequately captured. While FPNs are effective in enhancing multiscale representations, the limited pixel footprint of tiny objects means that without further refinement, the FPN alone may not achieve the necessary level of detail for reliable TOD in complex, cluttered scenes.

To mitigate this issue, various adaptations of the FPN architecture have been proposed. AugFPN (Guo et al., 2020) improves the representation of the top-level feature map by incorporating a Residual Feature Augmentation module. This module enriches the spatial context through a residual branch to reduce the information loss in the channels of the top-level feature map. LR-FPN (Li et al., 2024a) enhances the low-level positional information by extracting precise positional and saliency information from the shallow layers, and by integrating this information throughout the FPN via spatial and channel interaction modules. PANet (Liu et al., 2018) augments the FPN with a bottom-up pathway to strengthen the lower-layer localization, and employs adaptive feature pooling to connect all the feature levels directly. NAS-FPN (Ghiasi et al., 2019) leverages Neural Architecture Search to identify an optimal configuration of cross-scale connections, and creates a flexible feature pyramid structure with both top-down and bottom-up fusion paths. Xiao et al. (2023) introduce a Context Enhancement

Module within the FPN, and use multiscale dilated convolutions to fuse rich context information.

While recent methods have introduced various modules to enhance the FPN, the resulting increase in architecture complexity often outweighs the modest gains in detection precision. Additionally, it remains unclear whether these precision gains result from enhancements to the top-level feature map or from additional feature fusion methods across the entire set of FPN feature maps (see, e.g., Guo et al. (2020); Xiao et al. (2023)). Therefore, this paper proposes a simple Dilated Fusion (DF) module that is incorporated into the FPN to enhance the top-level feature map with enriched contextual information.

The DF module introduces an additional residual connection to the FPN that takes the deepest feature map from the backbone, enriches it through multiple dilated convolutions, and adds the enhanced feature map to the lateral convolution of the original feature map. Inspired by Xiao et al. (2023), who employ multi-rate dilated convolutions to capture context information across varying receptive fields, and by Guo et al. (2020), who use adaptive spatial fusion for the fusion of feature maps, the DF module combines these methods in a unified approach. The module generates three new feature maps from the deepest feature map through dilated convolutions with different strides. The new feature maps are fused via adaptive spatial fusion, enhancing the context and the spatial detail of the top-level feature map.

The deepest feature map in the FPN has the most channels and the smallest spatial resolution, capturing high-level, abstract information but with limited spatial detail. Dilated convolutions that increase the receptive field without downsampling are applied to this map to gather richer, multiscale contextual cues. Adaptive spatial fusion then combines these features without sacrificing spatial precision. By adding this enriched output to the top-level feature map, the DF module enhances the ability of the FPN to create feature representations in which tiny objects are more easily detectable, hence enhancing the precision. Figure 1 shows an example where OE-Net is more capable of detecting tiny objects that have limited feature representations than the Oriented R-CNN baseline.

2.5. Anchor Matching for TOD

In two-stage object detectors, the RPN identifies potential RoIs that are likely to contain objects, which are then refined and classified by a separate detection head. This two-stage process streamlines the overall computation by reducing the burden on the final detection head that focuses on a smaller subset of high-quality proposals instead of evaluating all the possible regions.

A core challenge in using an RPN lies in the label assignment process. This process matches each proposed region, also known as anchor, with a corresponding ground truth bounding box. Anchors are bounding boxes that are generated for each image at various scales and aspect ratios, aiming to capture objects of different sizes. Traditionally, RPNs use a MaxIoUAssigner that relies on the Intersection over Union (IoU) metric to calculate the overlap between each anchor and each ground truth bounding box. Based on predefined IoU thresholds, the MaxIoUAssigner classifies anchors as either positive or negative, determining which should proceed as RoIs and which should be discarded as redundant. However, for tiny objects, this IoU-based approach is problematic. Due to their small size, minor shifts in the anchor location cause large drops in the IoU. These drops flip the anchor label from positive to negative, causing tiny objects to have an insufficient number of positive training samples. Figure 2 illustrates the dense clustering of tiny objects in satellite images. In such scenarios, an IoU-based assigner favors larger objects, leading to an underrepresentation of positive samples for tiny objects. This imbalance significantly decreases the detection precision, since the RPN does not generate a sufficient number of proposals for the tiny objects that are present.

To address this issue, recent research has proposed to replace the IoU metric with alternative distance metrics, such as the Normalized Wasserstein Distance (NWD) (Peyré et al., 2019). Initial experiments with NWD focused on integrating the metric into the bounding box regression loss function of the final detection head, improving the detection precision for tiny objects (Yang et al., 2021b; Wang et al., 2024). Unlike IoU, NWD treats both the predicted bounding boxes and the ground truth bounding boxes as Gaussian distributions, allowing for a continuous distance metric rather than relying solely on the geometric overlap between the bounding boxes. This approach provides a more meaningful distance metric for tiny objects and remains informative even when there is no overlap between the predicted bounding box and the ground truth bounding box, which IoU cannot handle.

While integrating NWD in the bounding box regression loss of the final detection head improves TOD precision, its impact is limited because the detection head typically already receives refined, high-quality proposals from the RPN. In contrast, the RPN benefits significantly more from a precise distance metric such as NWD, as the initial distances between the anchors and the ground truth bounding boxes are much larger in this stage. Therefore, recent research focuses on incorporating NWD into ranking-based assigners for the RPN (Xu et al., 2022a,b). Unlike the MaxIoUAssigner that relies on fixed IoU thresholds, ranking-based assigners prioritize anchors based on their proximity to ground truth bounding boxes, converting the highest-ranked anchors into RoIs. However, this approach struggles when the top-ranked anchors still have poor NWD values, as selecting them based on a uniform ranking alone does not adequately reflect their quality in these cases.

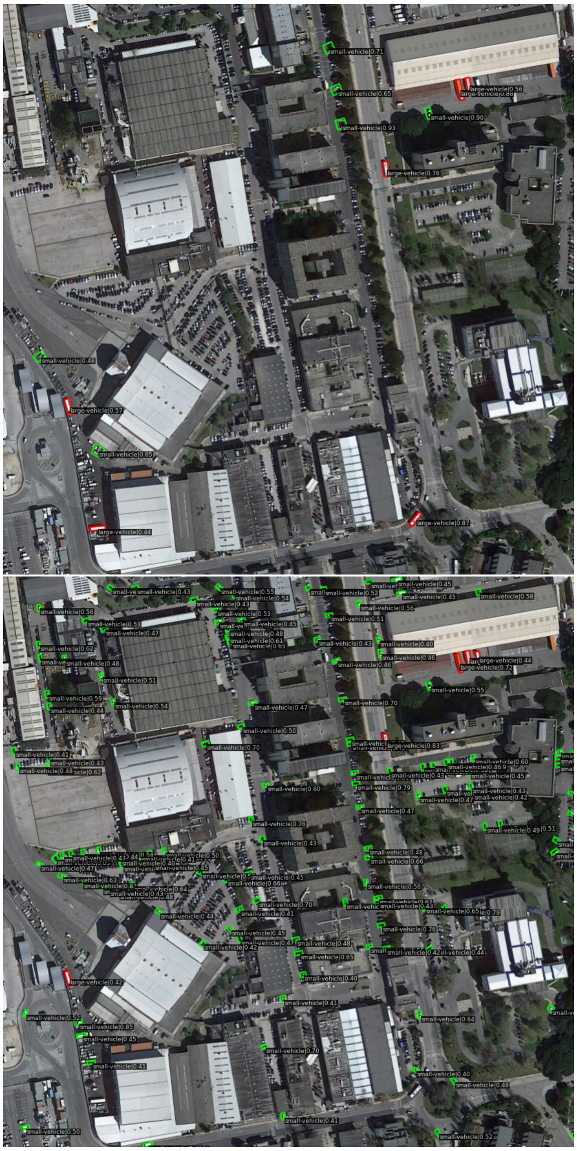

Figure 2: Oriented R-CNN (top) versus OE-Net (bottom). The inference results on an image from the TinyDOTA test dataset.

In response to this limitation, this paper proposes a Balanced Ranking Assigner (BRA) that combines NWD-based ranking with an objectness score. The BRA is inspired by Libra-SOD (Zhou and Zhu, 2024), which takes into account both classification confidence and localization quality during the assignment process. The classification confidence or objectness score, generated by the RPN, estimates the likelihood that an anchor contains an object based on the feature maps that are received from the FPN. By integrating the objectness score with NWD-based ranking, the BRA is able to resolve cases where the NWD values are inconclusive, allowing the objectness score to guide the ranking assignment. This balanced approach not only improves the assignment accuracy for tiny objects but also strengthens the ability of the RPN to discern meaningful RoIs, enhancing the overall detection precision in satellite imagery. Figure 2 shows an example where OE-Net is more accurate in detecting cluttered tiny objects than Oriented R-CNN. In such images, the BRA ensures that the RPN passes a sufficient number of positive samples for tiny objects to the detection head.

2.6. Classification for TOD

Semantic confusion due to background-foreground similarity presents a significant challenge in TOD. In satellite images, objects such as road surface markings and trailers, air-conditioning units and camper vans, or climbing frames and airplanes exhibit highly similar visual features. Due to the limited feature space and the uniform, top-down perspective of satellite images, certain objects frequently appear as indistinguishable spots with similar shapes or reflections. This leads to misclassifications and to a reduction in the detection precision. Figure 3 illustrates how a background object (i.e., a climbing frame in a playground), can have a similar feature representation to one of the classes in the dataset (i.e., a plane), leading to a false positive detection result in the top image.

Semantic confusion has driven research to refine the contextual information and emphasize the subtle distinctions within the feature space to better separate similar classes. LSKNet (Li et al., 2024c) leverages spatial and channel attention mechanisms inside the backbone to enhance the feature representations by refining the contextual information. Devaki et al. (2023) introduce an attention-based Group Enhance Module that groups the feature channels to improve the ability of the detector to extract discriminative features from subtle object differences. Second, their sub-saliency feature learning mechanism directs attention to secondary feature details that are often essential for distinguishing between closely related classes, rather than only the most salient features. PCLDet (Ouyang et al., 2023) introduces a prototype-based approach, and constructs a feature prototype for each class to serve as a reference for differentiating the fine-grained objects. By leveraging contrastive learning, PCLDet enhances the feature discrimination by maximizing the distance between different classes while minimizing the distance within each class.

While these methods improve the feature discrimination in complex scenes, they often introduce higher computational costs and complexity that reduce the interpretability and the adaptability in resource-constrained environments. The challenge remains to achieve a balance between refined feature discrimination and computational efficiency, while maintaining model transparency. Furthermore, recent methods focus on mitigating the semantic confusion between the classes in the dataset, rather than between the classes and visually similar background objects.

In response to these limitations, this paper proposes a computationally efficient, transparent module that mitigates semantic confusion using Vector Mapping (VM). This novel classification approach, inspired by the orthogonal mapping

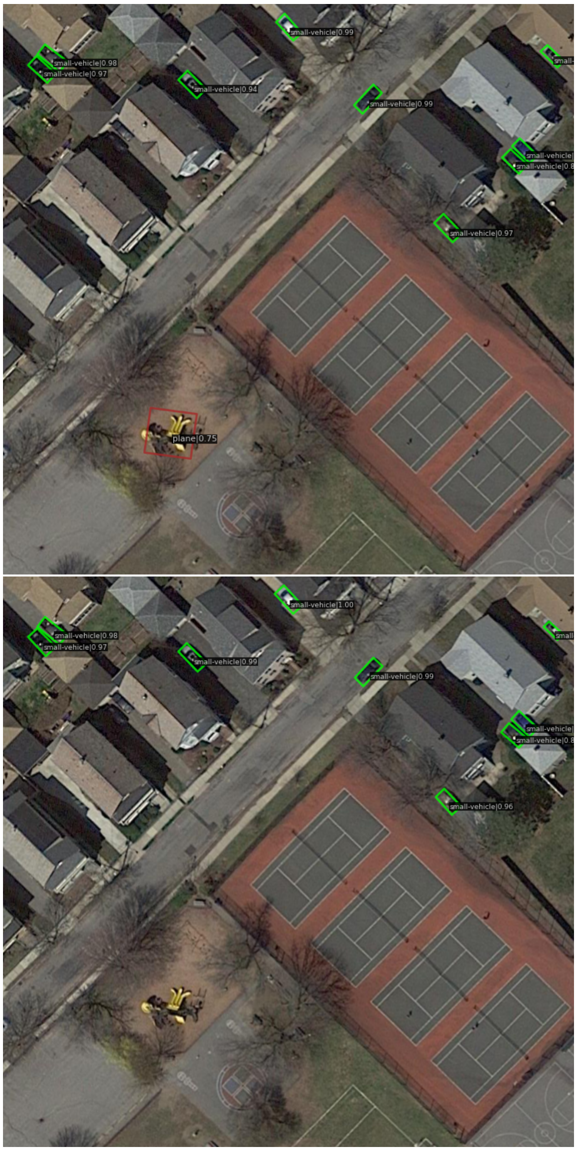

Figure 3: Oriented R-CNN (top) versus OE-Net (bottom). The inference results on an image from the TinyDOTA test dataset.

module in Zhu et al. (2024), reduces semantic confusion without adding computational overhead, and allows easy integration with several baseline object detectors. Instead of using linear mapping layers in the final classification stage of the detector, which is the common approach in object detection architectures, VM employs a predefined set of normalized vectors that represent a set of unique directions in the feature space. By computing the cosine similarity between these fixed vectors and the feature representations of the proposed RoIs, VM produces classification scores based on the feature alignment.

Unlike traditional object detectors that rely on adjustable weights in the classification stage, VM uses fixed classification vectors, removing the need for weight optimization in this stage. This way, VM forces upstream layers to generate semantically distinct feature maps for each class, as classification relies solely on feature alignment with these fixed vectors. Through hard constraints, VM assigns inherently dissimilar category prototypes to each class. This maps different classes to distinct directions on the unit hypersphere that represents the extracted feature space, inherently mitigating the background-foreground confusion. VM offers a simple, effective way to enhance classification performance in TOD tasks, providing a transparent alternative to compute-heavy models that improves accuracy while preserving interpretability, adaptability and applicability in baseline detectors. Figure 3 gives an example of how OE-Net is more capable of distinguishing a background object from a class instance than Oriented R-CNN.

3. Methodology

The methodology section explains how a two-stage object detector works and describes the additional modules that OE-Net incorporates.

3.1. Anchor-based Two-stage Object Detectors

In general, two-stage object detectors begin with loading and preprocessing the input images. Preprocessing steps include random flipping, normalizing, and padding. OE-Net also adds random rotations to increase the robustness of the model to different object orientations. This is particularly important in satellite imagery, where object orientations vary significantly. The images are then passed to the ResNet-50 backbone. In satellite images, objects often have scarce feature representations due to their limited pixel resolution and the presence of background noise. This makes the contextual information of the image crucial for TOD. OE-Net addresses this by incorporating context blocks (Cao et al., 2019) into the ResNet-50 backbone, which enhance the global context modeling inside the residual layers.

The backbone extracts hierarchical feature maps from the input images. These maps are multidimensional representations of the image. At each consecutive map, the spatial resolution decreases while the channel depth increases. Thus, the deeper the feature map, the more abstract and high-level the extracted features become. The FPN refines these feature maps across different scales, allowing for the detection of objects at varying sizes. OE-Net extends the FPN with a residual module that leverages dilated convolutions and adaptive fusion to improve the spatial awareness in the deepest feature maps. The RPN then generates a set of anchor boxes that are evaluated against the ground truth bounding boxes to identify the RoIs. OE-Net enhances this process with the BRA, replacing the traditional IoU metric with NWD and incorporating an objectness score to resolve the ambiguous assignments where all the anchors have highly similar NWD values.

Finally, the RoIs are processed by the detection head via the RoI Extractor. This module pools the feature map regions from the FPN that correspond to the proposals from the RPN. The extracted RoIs are then passed to and

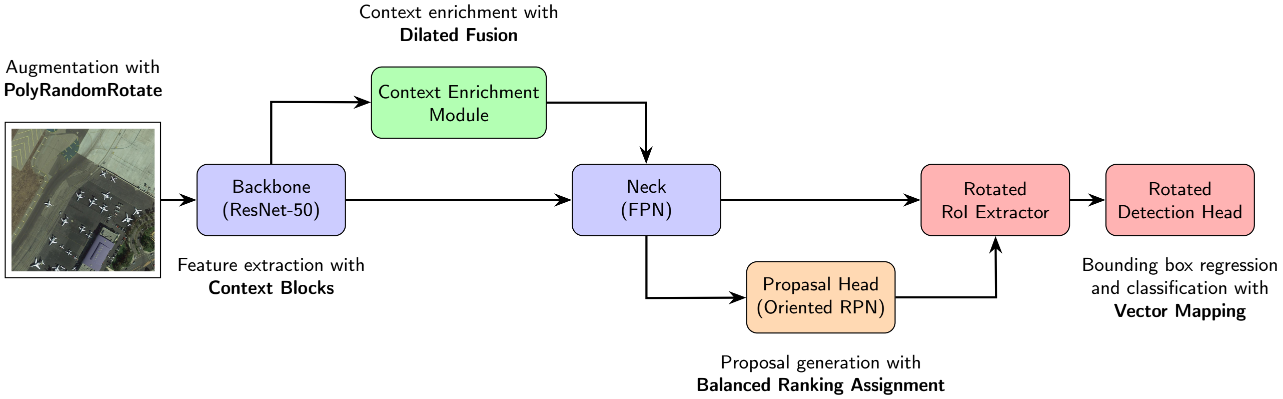

Figure 4: Schematic of OE-Net, an extension of Oriented R-CNN, incorporating several additional modules. OE-Net uses PolyRandomRotate for the data augmentation, context blocks for the feature extraction in the ResNet-50 backbone, and Dilated Fusion (DF) to better preserve the contextual information in the feature maps. Additionally, OE-Net integrates a Balanced Ranking Assigner (BRA) in the Region Proposal Network (RPN) and Vector Mapping (VM) for the classification of the predicted RoIs (RoIs). The schematic illustrates the data flow from the input image through to the final detection head.

processed by the detection head. OE-Net introduces VM, a novel classification method, that calculates the similarity between the extracted feature vectors and a predefined set of feature vectors. This way, VM compels upstream modules to produce distinct feature representations for each class in the dataset. VM mitigates semantic confusion between visually similar instances, which is a significant challenge in satellite imagery due to the small size and the limited detail of the objects. The data flow and the different modules of the OE-Net architecture are schematically illustrated in Figure 4.

3.2. ResNet-50 Backbone with Context Blocks

ResNet-50 is a deep CNN, pre-trained on ImageNet (He et al., 2016; Deng et al., 2009). Designed for image classification, its strength lies in its ability to learn features at various levels of abstraction. The early layers capture simple features such as edges and textures, while the deeper layers capture complex features, such as contextual relationships. Object detectors rely on multilevel feature maps to detect objects of varying scales and complexities. Hence, ResNet-50 is utilized by object detectors as an upstream feature extractor. Table 1 details the specific layers of the ResNet-50 backbone used by OE-Net. ResNet-50 begins with a stem layer that uses strided convolutions with large kernel sizes (i.e., 7×7) to downsample the input image, reducing the computational load on the subsequent layers. This is followed by four stages, each composed of multiple residual blocks. The backbone generates four feature maps that are subsequently processed by the FPN. At each consecutive feature map, the number of channels doubles, while the resolution of the image halves.

Training deep CNNs presents significant challenges, particularly the vanishing gradient problem (Bengio et al., 1994). As gradients are propagated backward, they are repeatedly scaled by weight matrices at each layer. This causes the gradients to become exponentially smaller and impede the learning process in the early layers. ResNet-50 addresses this issue through residual connections that allow the gradients to propagate directly by adding the input of a residual block to its output. This preserves the gradient magnitude across the layers. Figure 5 illustrates the structure of a residual layer, highlighting its two branches: the main branch (left), which processes the input through convolutions, and the residual branch (right), which either downsamples the input using a convolution or passes it unchanged. In the first residual layer of each stage, the residual branch downsamples the input to match the dimensionality of the main branch. In the subsequent layers, the residual branch passes the input directly, as downsampling is no longer necessary.

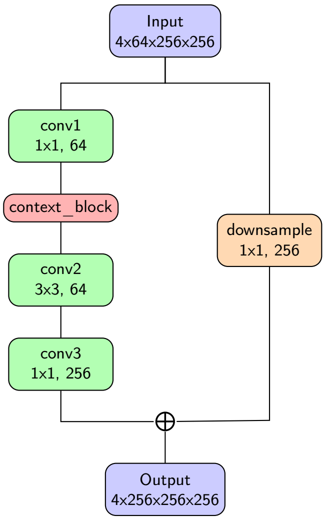

Figure 5: A schematic of the first of three residual layers within the initial stage of the four-stage ResNet-50 architecture. In the second and the third residual layer, the residual branch (right) does not comprise a downsampling layer, and the original input is added to the output of the main branch (left).

Table 1
The different layers of the ResNet-50 backbone. OE-Net integrates a set of context blocks into ResNet-50. The context blocks have been added between the first and the second convolution of each residual layer.

Layer	Output	Kernel, Channels, Stride
Input Image	-	-
conv1	512x512	7x7, 64, 2
maxpool	256x256	3x3 max pooling, stride 2
layer1	256x256	$\left\{\begin{array}{l} 1\times1, 64, 1 \\ \text{context\_block} \\ 3\times3, 64, 1 \\ 1\times1, 256, 1 \end{array}\right\}\times 3$
layer2	128x128	$\left\{\begin{array}{l} 1\times1, 128, 1 \\ \text{context\_block} \\ 3\times3, 128, 2 \\ 1\times1, 512, 1 \end{array}\right\}\times 4$
layer3	64x64	$\left\{\begin{array}{l} 1\times1, 256, 1 \\ \text{context\_block} \\ 3\times3, 256, 2 \\ 1\times1, 1024, 1 \end{array}\right\}\times 6$
layer4	32x32	$\left\{\begin{array}{l} 1\times1, 512, 1 \\ \text{context\_block} \\ 3\times3, 512, 2 \\ 1\times1, 2048 \end{array}\right\}\times 3$
Total Parameters	-	23.915 M

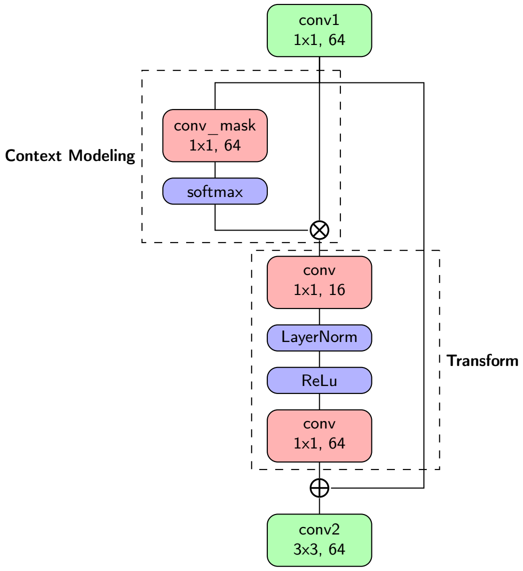

Figure 6: A schematic of the context block. The context block is added after the first convolution in each residual layer across all four stages. Matrix multiplication is used by the global attention pooling mechanism (upper dashed rectangle) to model the context of the feature map. Element-wise addition is used to fuse the processed feature map from the transform block (lower dashed rectangle) with the original feature map.

ResNet-50 captures only a limited range of contextual information, as its hierarchical structure primarily focuses on extracting features through local receptive fields. For tiny objects that suffer from sparse feature representations, increasing the contextual information enhances the quality of the feature maps. To extract and preserve global context, context blocks (Cao et al., 2019) are introduced after the first convolutional layer of each residual block inside the backbone. As illustrated in Figure 6, a context block generates a feature attention map via a softmax function applied to the output of a convolution. This attention map is then multiplied with the input feature map to highlight the important regions. The resulting features are refined in a transform block to capture the relationships between different feature channels. The processed features are then fused with the original feature map through an element-wise addition. By integrating context blocks across all four stages of ResNet-50, the network captures both local and global contextual information, enhancing the spatial awareness of the feature maps.

3.3. FPN with Dilated Fusion

FPNs address the challenge of detecting objects at varying scales, which is essential for TOD in satellite imagery. OE-Net employs an FPN that extracts the feature maps from different layers of the ResNet-50 backbone. Figure 7 shows that this results in a set of hierarchical feature maps, $\{C2, C3, C4, C5\}$, where the spatial resolution decreases and the channel depth increases with each subsequent level. The feature map at the highest level, C5, is first transformed using a lateral convolution. The resulting feature map, M5, is progressively propagated downward and fused with the lower-level feature maps, $\{C4, C3, C2\}$, through lateral convolutions and through upsampling. The resulting feature maps, $\{M4, M4, M4\}$, are enriched with spatial and contextual information from the higher-level feature maps. In traditional FPNs, the top-level feature map, M5, suffers from significant information loss due to the reduced feature channels. Therefore, M5 lacks the multiscale context needed to align with the other levels in the pyramid. This limitation weakens the representation of M5 and adversely affects the overall performance of the resulting feature pyramid.

To address this, OE-Net uses the DF module to refine M5. DF introduces a residual branch that enriches M5 with spatial and contextual information. By augmenting the spatial context, the DF module compensates for the information loss in M5, thereby improving its compatibility with the feature maps at other levels and enhancing the overall representation of the feature pyramid. DF uses dilated convolutions, of which an example is shown in Figure 8, that expand the receptive field of the convolutional filters without increasing the number of parameters or downsampling the feature map. By spacing the filter weights apart, dilated convolutions capture broader context and preserve spatial details (Yu and Koltun, 2016). Figure 9 shows that in the DF module, three dilated convolutions with dilation rates of 1, 3, and 5 are applied to C5. The three resulting feature maps capture local, intermediate, and broad contextual information. These feature maps are normalized using batch normalization, and non-linearities are introduced through ReLU activation.

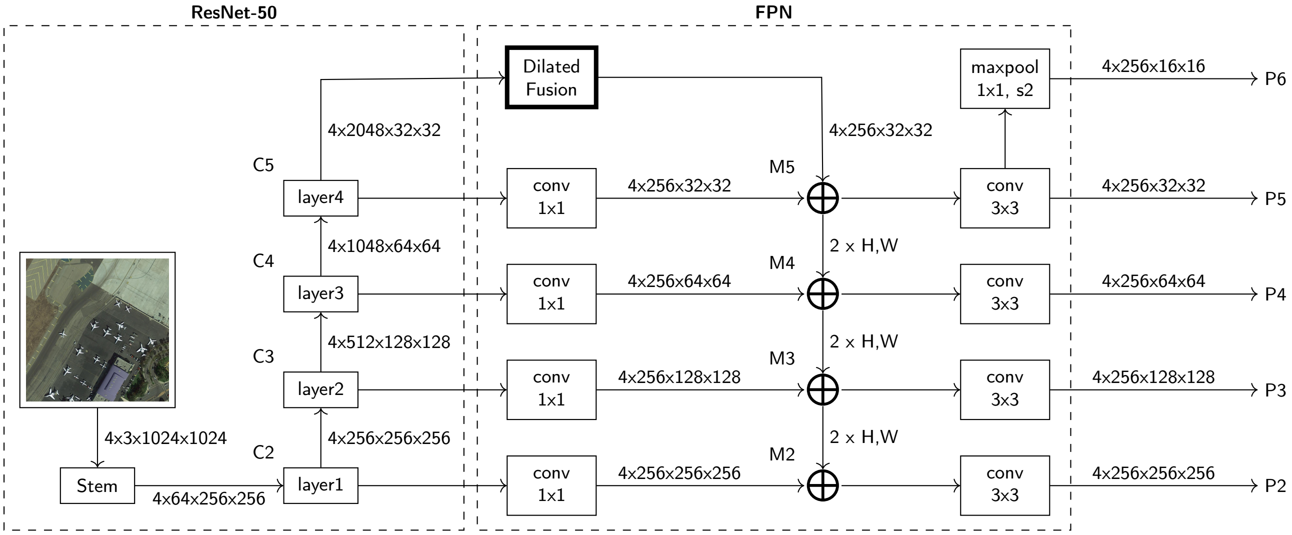

Figure 7: A schematic of the Feature Pyramid Network (FPN) with an integrated Dilated Fusion (DF) module that processes the deepest feature map (C5) from the ResNet-50 backbone. For the convolutional layers, denoted by conv, 1x1 and 3x3 refer to kernel sizes of 1 and 3, respectively. In the top-down path of the FPN, the resolution of the upper tensor is doubled before it is summed with the lower tensor. To accommodate Oriented R-CNN, an extra feature map (P6) is created using max pooling.

To combine the multiscale feature maps, $\{D1, D3, D5\}$, the DF module employs an attention-based adaptive spatial fusion mechanism. First, the three feature maps are interpolated along the resolution dimension and concatenated along the channel dimension. An attention layer generates fusion weights by sequentially applying convolution, activation, a second convolution, and a softmax function to the multiscale feature maps. These weights are applied to the original three feature maps through element-wise multiplication. The resulting weighted feature maps are then summed to produce a fused output that matches the dimension of M5. The fused feature map is then added to M5 through the residual connection, creating an enhanced top-level feature map with an improved multiscale feature representation.

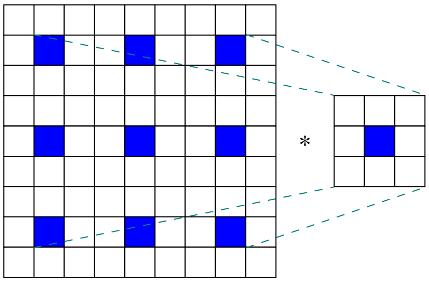

Figure 8: A dilated convolution with a kernel size and a dilation of 3. The receptive field is increased without downsampling.

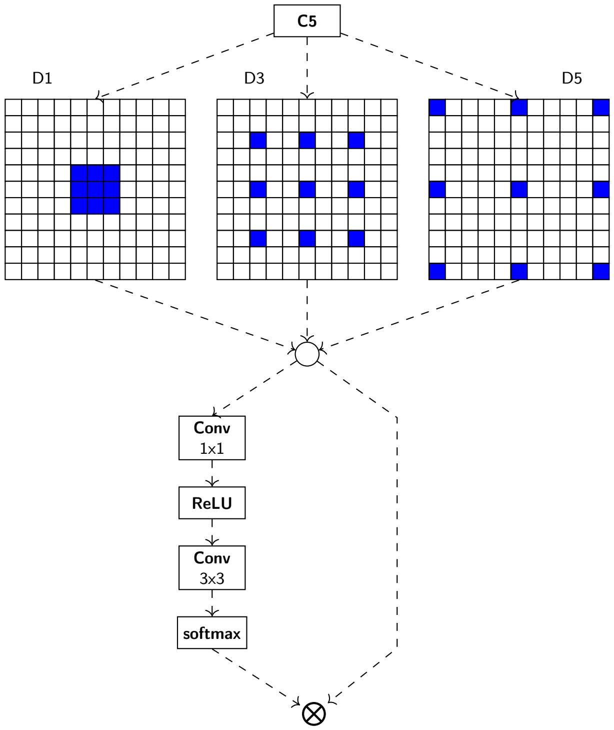

Figure 9: A schematic of the Dilated Fusion (module). The module consists of two parts: the first part comprises three dilated convolutions with dilations of 1, 3, and 5, while the second part consists of the adaptive spatial fusion branch. In the FPN, the output of the DF module is added to the lateral convolution of the deepest feature map from the backbone.

3.4. RPN with Balanced Ranking Assigner

The Oriented RPN (Xie et al., 2021) generates a set of oriented proposals using a number of CNNs. It operates on the five feature maps $\{P2, P3, P4, P5, P6\}$, propagated by the FPN. Each feature map is processed by a head that consists of a 3×3 convolutional layer and two sibling 1×1 convolutional layers. At each spatial location of the feature map, the network assigns three horizontal anchors with aspect ratios of 0.5, 1, and 2. The anchors are generated for each feature map, with the sizes scaling according to the feature map stride: 32 × 32 at P2, 64 × 64 at P3, and so on, up to 512 × 512 at P6. This scaling corresponds to the strides of the feature maps and the scaling factor of the anchor generator. For example, P2 has a resolution of 256 × 256, indicating that the original 1024 × 1024 image has been downsampled with a stride of 4. The pixel area of the anchors is then determined by squaring the product of the scaling factor and the stride, ensuring correctly sized anchors at each feature map level. Each anchor is represented as a 4-dimensional vector (a_x, a_y, a_w, a_h), where a_x and a_y define the center, and a_w and a_h denote the width and height of the anchor.

The regression branch of the RPN predicts the offsets $(\delta_x, \delta_y, \delta_w, \delta_h, \delta_\alpha, \delta_\beta)$ that represent the deviations between the anchors and the ground truth bounding boxes. These offsets are used to convert the horizontal anchors into oriented anchors with six parameters $(x, y, w, h, \Delta\alpha, \Delta\beta)$. As illustrated in Figure 10, x, y define the center, w, h the dimensions, and $\Delta\alpha, \Delta\beta$ the offsets relative to the midpoints of the top and the right sides of the external rectangle. Essentially, the horizontal anchor is used as the external rectangle of the oriented anchor. This method, known as midpoint offset representation (Xie et al., 2021), calculates the coordinates of the four vertices (i.e., $\mathbf{v}_1 \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$) of the oriented anchor based on the six aforementioned parameters. Hence, the midpoint offsets $\Delta\alpha$ and $\Delta\beta$ shift the vertices from the midpoints of the horizontal anchors along its top and right edge, rotating the anchor.

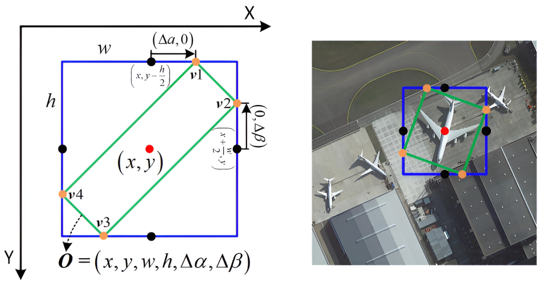
X
Y
w
h
$(\Delta a, 0)$
$\left(x, y-\frac{h}{2}\right)$
$v1$
$v2$
$v3$
$v4$
$(0, \Delta\beta)$
$\left(x+\frac{w}{2}, y\right)$
(x, y)
$O = (x, y, w, h, \Delta\alpha, \Delta\beta)$

Figure 10: The regression branch of the Region Proposal Network (RPN) uses the midpoint offset representation to turn horizontal anchors into oriented proposals (Xie et al., 2021).

The classification branch computes the objectness score, estimating the likelihood that each anchor contains an object. In traditional label assignment, where the anchors that will be passed to the detection head are selected, the anchors are classified as positive or negative based on their IoU with the ground truth bounding boxes. IoU-based assignment exhibits significant sensitivity to object scale, with tiny objects showing large variations in IoU due to minor positional shifts. Furthermore, IoU fails to capture the spatial relationships between bounding boxes, particularly when there is no overlap or when the boxes are fully contained within one another. These limitations result in inaccurate labeling in the case of TOD.

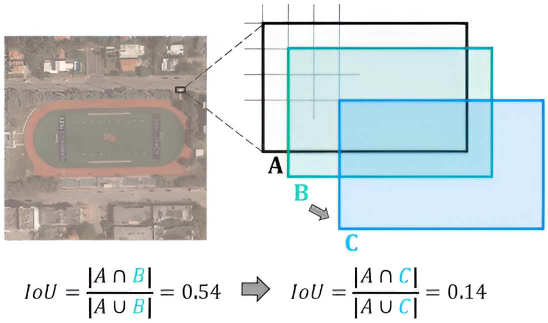
A
B
C
$IoU = \frac{|A \cap B|}{|A \cup B|} = 0.54$
$IoU = \frac{|A \cap C|}{|A \cup C|} = 0.14$

Figure 11: A visualization of how the Intersection over Union (IoU) metric is extremely sensitive to positional shifts in the anchors for tiny objects (Xu et al., 2022a).

Addressing these limitations, OE-Net uses NWD to assess the spatial accuracy of the anchors. Specifically, the anchors and the ground truth bounding boxes are modeled as two-dimensional Gaussian distributions, and the Wasserstein Distance is used to measure the similarity between these distributions. To ensure that the value range is normalized such that ranges between 0 and 1, an exponential nonlinear transformation is applied to the Wasserstein distance, resulting in the NWD. This method works particularly well for tiny objects (see e.g., Xu et al. (2022a)), because the ground truth bounding boxes of tiny object often have minimal or no overlap with most anchors. The Wasserstein distance is still able to measure the distribution similarity even when the overlap is negligible. Moreover, NWD is less sensitive to location deviations and exhibits scale-invariant properties, making it a more reliable metric for accurate label assignment in TOD than IoU.

Traditional threshold-based label assignment, poses challenges for TOD because tiny objects have minimal or no overlap with anchors, hindering them to meet the IoU threshold. This imbalance skews the assignment process, disproportionately favoring larger objects and leaving tiny objects underrepresented during the training. To address this, ranking-based assignment is used. This method replaces the threshold-based approach with a rank-based mechanism that prioritizes anchors with the highest similarity scores to ground truth bounding boxes. Specifically, an NWD score matrix that ranks the anchors based on their scores for each ground truth bounding box is computed. Rather than applying a static threshold, ranking-based assignment assigns positive labels to anchors within the top-K ranked scores for a given ground truth bounding box. This strategy ensures that even tiny objects have sufficient positive samples, leveraging the robustness and scale-invariance of NWD. By focusing on top-ranked anchors, ranking-based assignment generates a more balanced distribution of objects in terms of scale.

In case the NWD scores for multiple anchors are highly similar, such as when the anchors have uniformly low similarity scores to the ground truth bounding boxes, the ranking process becomes ambiguous. To address this, OE-Net introduces the BRA, combining the NWD with the objectness scores generated by the classification branch of the RPN. By incorporating objectness, BRA leverages additional information about the likelihood that an anchor contains an object to resolve ambiguities and guide the ranking process in non-trivial ranking processes.

The BRA first scales and standardizes the NWD scores and the objectness scores such that their proportionalities are similar. Then, the BRA computes a combined ranking score by weighting the NWD score with the objectness score. This combined ranking score is given by:

$$\xi^{\mathrm{C}} = \left(\frac{\log(\xi^{\mathrm{W}} + \epsilon) - \mu_{\xi^{\mathrm{W}}}}{\sigma_{\xi^{\mathrm{W}}}} \right)^{\alpha} \cdot \left(\frac{\xi^{\mathrm{O}} - \mu_{\xi^{\mathrm{O}}}}{\sigma_{\xi^{\mathrm{O}}}} \right), \qquad (1)$$

where ξ^{W} represents the NWD score, ξ^{O} represents the objectness score, ϵ is a positive scaling factor used to normalize the scores to a similar proportionality, μ denotes the mean, σ is the standard deviation of the respective scores, and α is the weighting factor that amplifies the NWD score.

Following this calculation, the anchors are ranked on the combined ranking score. The top k ranked anchors for each ground truth box are assigned as positive samples, while the remaining anchors are labeled negative. This approach enhances the robustness of the label assignment, particularly in challenging cases where tiny objects or non-overlapping anchors make NWD scores alone insufficient. By integrating both spatial accuracy and object likelihood, the BRA ensures a well balanced and more accurate label assignment.

3.5. Detection Head with Vector Mapping

The detection head, which is the final component of the object detection pipeline, receives a set of RoIs and their corresponding feature maps from the Rotated RoIAlign module. This module extracts rotation-invariant features from the oriented region proposals that have been generated by the RPN. The oriented proposals that are passed by the RPN are defined as parallelograms. These parallelograms are represented by their four vertex coordinates, as shown in Figure 10. The parallelogram is transformed into an oriented rectangle parameterized by (x, y, w, h, θ), where x, y represent the center coordinates of the rectangle, w, h the width and height, and θ the rotation angle. This transformation is achieved by extending the shorter diagonal of the parallelogram to match the length of the longer diagonal. The resulting rectangle maintains the orientation of the original parallelogram.

The oriented rectangle is projected onto its corresponding feature map $\mathbf{F}$. This way, a rotated RoI is created. Each rotated RoI is divided into a fixed grid of $m \times m$ cells. A rotation transformation is applied to align the features within each cell to the orientation of each RoI in that cell. The resulting feature map $\mathbf{F}^*$ has a dimension of $m \times m \times C$, where C is the number of channels in the original feature map. Hence, the output of the Rotated RoIAlign module is a fixed-size feature tensor of dimensions (RoIs, C, m, m). In TOD, the number of RoIs and the dimensionality of the extracted features reach orders as high as 10^3.

As shown in Figure 12, the feature tensor is then passed to the detection head that refines and predicts the bounding boxes and computes the classification scores for each RoI. The feature tensor is flattened and passed through two fully connected layers with ReLU activations to reduce its dimensionality. The feature tensor is then passed to two separate branches: a classification branch and a regression branch.

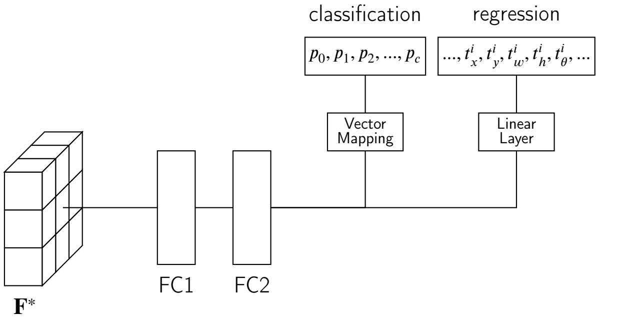
classification
regression
$p_0, p_1, p_2, ..., p_c$
$..., t^i_x, t^i_y, t^i_w, t^i_h, t^i_\theta, ...$
Vector Mapping
Linear Layer
FC1
FC2
$\mathbf{F}^*$

Figure 12: A schematic of the detection head. The Region Proposal Network (RPN) passes the feature tensor $\mathbf{F}^*$ to the detection head. This head processes the feature tensor using two fully connected layers and two separate branches for the regression and the classification of the bounding boxes.

In the baseline Oriented R-CNN architecture (Xie et al., 2021), the classification branch comprises a linear layer that produces a tensor that indicates the probability of each RoI belonging to a particular class. In other words, the linear layer is responsible for mapping the extracted features to a tensor that indicates to which class the features belong. Figure 13 shows how the extracted feature maps are mapped to a set of classification scores by the linear layer, otherwise known as the fully connected layer. This approach leverages the learning capabilities of the weights inside the linear layer. When classes and background objects have highly similar feature representations, however, the linear layer struggles with differentiating between these instances, resulting in the misclassification of objects. Since linear layers treat each class independently, they fail to capture the semantic similarities between classes and background objects.

In contrast, the Vector Mapping (VM) method compares the extracted feature maps with a predefined set of feature maps, each corresponding to a specific class in the dataset. VM employs a random set of predefined feature vectors with a dimensionality that is proportional to the dimension of the propagated feature map tensor. These vectors are normalized to represent a set of distinctly different directions on a unit hypersphere that spans the propagated feature space. In essence, the vectors mimic a set of semantically

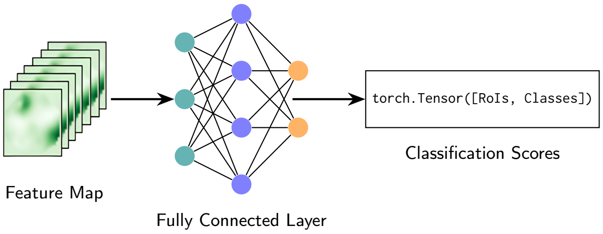

Figure 13: A schematic illustrating the traditional process for classifying the predicted bounding boxes: a fully connected layer transforms the feature representation of each RoI into a set of probabilities, indicating the likelihood that the region belongs to any of the classes in the dataset.

different feature representations in the same feature space as that of the input images. Figure 14 shows how the extracted feature maps (left), with a dimensionality proportional to the number of RoIs and the dimension of the propagated feature space, are multiplied with the predefined set of feature vectors (middle), with a dimensionality proportional to the number of classes and the dimension of the feature space.

Rather than relying on a linear layer for the classification, VM calculates the cosine similarity between the propagated feature tensor and the predefined vector tensor. Using cosine similarity, the classification score is calculated as:

$$\psi^{K\times c} = (X^{K\times F} \cdot (V^{c\times F})^{\top}) \times 100, \tag{2}$$

where ψ represents classification score, X the extracted feature tensor and V the predefined feature tensor. Additionally, K represents the number of RoIs, F the dimensionality of the feature space, and c the number of classes in the dataset. Cosine similarity measures the angular relationship between the tensors, ensuring that classes are distinguished by distinct directions in the feature space. This operation yields a classification score tensor with a dimensionality that is proportional to the number of RoIs and the number of classes, wherein each score represents the degree of similarity between the feature representations of the RoIs and the predefined vectors. The resulting classification scores are then passed to the loss function, and the loss is propagated backward through the upstream layers of the detector.

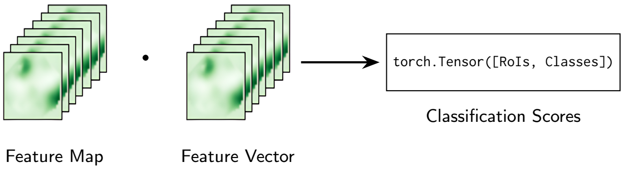

Figure 14: A schematic of Vector Mapping (VM): for each RoI, the cosine similarity (denoted by the dot product symbol in the figure) is calculated between the extracted feature representation and a predefined set of feature vectors. These similarities represent the probabilities that the region belongs to specific classes.

While vector mapping is inspired by the orthogonal mapping module by Zhu et al. (2024), it diverges in its approach by not strictly orthogonalizing the vectors. Instead, it leverages the randomness of the vectors on the unit hypersphere, which has the same dimension as the propagated feature space. This enables a more flexible and dynamic interaction with the feature representations. Nevertheless, to guarantee a minimum of directional difference among the random vectors, a pre-training step using a cosine similarity loss is employed. This ensures that if any of the randomly initialized vectors happen to have similar orientations, they are adjusted slightly to diverge from one another. Unlike Zhu et al. (2024), where the vectors are strictly orthogonalized, VM retains the randomness of the vector initialization, proving that strict orthogonality is unnecessary. Instead, the pre-training process subtly encourages the vectors to spread out while maintaining their fundamentally random nature. The cosine similarity loss used for this pre-training is defined as follows:

$$\mathcal{L}^{\text{CE}} = \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{V}_i \cdot \mathbf{V}_i^{\top} - \mathbf{I}_i \right)^2, \tag{3}$$

where $\mathbf{V}$ is the set of N predefined feature vectors, and $\mathbf{I}$ the identity matrix used to ignore self-similarity. This process ensures that the predefined feature vectors serve as distinct class prototypes, encouraging the extracted feature maps to align well with these prototypes during the training. Algorithm 1 shows how the feature vectors are initialized, normalized, and pre-trained based on the cosine similarity loss function. Furthermore, it shows how the classification scores are calculated as the cosine similarity scores between the extracted features and the predefined vectors.

By using fixed vectors as semantic targets, vector mapping introduces an implicit clustering objective that encourages the feature maps to self-organize around predefined class directions. In other words, the fact that the predefined vectors are fixed during training forces the upstream modules in the detector to create distinctly different feature maps for each class in the dataset. In traditional object detectors, the linear classification layer acts as a weight matrix, with the gradients flowing through this layer and into the upstream modules. Since the classification branch has trainable weights, a portion of the loss gradient is used to update these weights, which reduces the extent of learning in the upstream feature extraction modules. With vector mapping, there are no weights to be adjusted in the classification head. Instead, the network must learn to produce feature vectors that align well with one of the predefined vectors. Since the directions are fixed, the network needs to produce feature vectors that naturally cluster in the directions of these vectors, ensuring that the features themselves encode the necessary information for the classification stage. When background objects share semantic features with one of the classes, VM is more capable of differentiating between them, since the class-specific encoding of the feature maps emphasizes unique characteristics that separate the class instances from the similarly looking background instances.

This shift in weight adjustments towards the upstream modules has a significant impact on the learning dynamics. In neural networks, stacking modules dilutes the focus on the feature extraction, as the gradients are progressively weakened across the layers. By removing the need for weight updates in the classification branch, vector mapping consolidates the learning process in the feature extraction layers, ensuring that the gradients contribute more heavily to shaping class-distinct feature maps. This approach encourages the feature extraction modules to encode class-specific information directly within the feature maps. As a result, the network develops feature maps that are inherently class-distinct and foreground-specific, and are not dependent on a set of trainable weights in the final classification layer.

Algorithm 1 Pseudocode for Vector Mapping (VM)

```
# Generate normalized random vectors
vectors = GenerateRandomVectors(classes, feats_dim)
vectors = vectors / Normalize(vectors)

# pre-train vectors
optimizer = InitializeAdamOptimizer(vectors)
scheduler = InitializeReduceLROnPlateau(optimizer)

for epoch in epochs:
    optimizer.zero_grad()
    loss = CosineSimilarityLoss(vectors)
    loss.backward()
    optimizer.step()
    scheduler.step(loss)

% Calculate classification scores in forward function
for features in dataloader:
    cls_scores = 100.0 * (features @ Transpose(vectors))
    loss = CrossEntropyLoss(cls_score, labels)
    loss.backward()
```

4. Experiment

This section presents the quantitative and the qualitative results of two case studies that use OE-Net to enhance the precision of TOD in satellite images. This section also includes a detailed description of the datasets, the evaluation metrics, and the experimental settings.

4.1. Datasets

(1) Tiny-DOTA: This TOD dataset was proposed by Lee et al. (2022). TinyDOTA is a refined version of the DOTA v2.0 dataset (Ding et al., 2021). DOTA v2.0 comprises satellite images with a wide variety of classes, from small objects such as vehicles and ships to larger objects such as soccer fields and basketball courts. DOTA v2.0 sources images from Google Earth, the Gaofen-2 and Jilin-1 satellites, and CycloMedia. The spatial resolutions of these image range from 0.1 to 4.5 meters. To focus specifically on TOD, TinyDOTA discards the larger objects, creating a dataset that emphasizes 8 tiny object classes. Since the original DOTA v2.0 test dataset does not have publicly available labels, a custom split of 70%, 10%, 20% for the training, validation, and test subsets is used for the experiments, respectively. The original images are divided into 1024 × 1024 pixel patches with a stride of 512 pixels. The ground truth bounding boxes are oriented bounding boxes. The final Tiny-DOTA dataset includes the following 8 object categories: plane (pl), bridge (br), small-vehicle (sv), large-vehicle (lv), ship (sh), storage-tank (st), swimming-pool (sp), and helicopter (hc). Table 2 shows the number of instances per class for each of the three splits, as well as the total number of images.

Table 2
Further statistics on the TinyDOTA dataset.

Class	Train	Val	Test
Storage Tank	12,892	1,400	4,558
Bridge	3,733	803	804
Large Vehicle	46,067	6,169	8,729
Small Vehicle	276,689	31,415	71,059
Ship	78,822	8,943	24,994
Plane	16,377	1,056	5,464
Swimming Pool	4,132	360	1,323
Helicopter	1,244	0	139
Total Images	22,545	2,642	6,086

(2) ExcaSat: This excavator detection dataset consists of two parts. First, ExcaSat comprises 232 high-resolution satellite images acquired from Airbus' Pleiades Neo and Planet's SkySat satellites, with spatial resolutions of 0.3 and 0.5 meters, respectively. These images were originally acquired to perform change detection for pipeline monitoring. Hence, the images serve for excavator and vehicle detection in areas that surround pipeline corridors. Second, ExcaSat consists of 557 high-resolution satellite images that were originally in the FAIR1M2.0 (Sun et al., 2022) dataset.

FAIR1M2.0 contains more than 40,000 images for TOD in satellite imagery, and it includes an excavator class. From FAIR1M2.0, the 557 images that contain one or more excavators were extracted and incorporated into ExcaSat. The trainval set was created by fusing the 557 FAIR1M2.0 excavator images with 162 (i.e., 70%) of the acquired images. The test set consists of the 70 (i.e., 30%) acquired images that remained. In order to augment the trainval set, a multiscale version of the original dataset was created using a sliding window that divides the images into smaller patches, keeping the original image and the patches that contain objects. The resulting trainval and test sets are detailed in Table 3. The ground truth bounding boxes are oriented bounding boxes. ExcaSat includes three classes: excavator (ex), small-vehicle (sv), and large-vehicle (lv).

Table 3
Further statistics on the ExcaSat dataset.

Class	Trainval	Test
Small Vehicle	81,795	6,260
Large Vehicle	14,445	360
Excavator	4,021	149
Total Images	1,976	70

In satellite imagery, variations in the sensor resolution and the object scale create significant challenges for object detection models. The objects appear at vastly different sizes due to the distance, the perspective, or the inherent size differences within the same classes. This often leads detectors to rely too heavily on the object size, resulting in misclassifications when the objects appear at unexpected scales (Xia et al., 2018). To address this issue, TinyDOTA and ExcaSAT incorporate images from a variety of sensors with multiple spatial resolutions. Providing diverse resolutions allows the models to learn object features that are not solely dependent on the size, making the models more robust across a range of object scales and spatial resolutions.

The training process incorporates a series of preprocessing and augmentation techniques to enhance the robustness and the generalization of the models. The process begins by loading the images and the annotations that contain the bounding boxes. To increase the variation, the `RRandomFlip` (Zhou et al., 2022b) algorithm introduces horizontal, vertical, and diagonal flips to the images. This creates the diversity in the object orientation. Additionally, inspired by Li et al. (2024c), the `PolyRandomRotate` (Zhou et al., 2022b) algorithm randomly rotates 50% of the images by angles up to 180 degrees. Finally, the images are normalized using the mean and the standard deviation values, and padded for efficient batch processing. Finally, the `DefaultFormatBundle` and `Collect` steps structure the data by gathering the images, the bounding boxes, and the labels, converting the information into the correct format for model input.

4.2. Evaluation Metrics

Similar to Zhou et al. (2022b), several metrics are used to assess the detection accuracy and the inference speed of the models. These include Intersection over Union (IoU), Precision, Recall, Average Precision (AP), and mean Average Precision (mAP). The relevant formulas and definitions for the key metrics are as follows:

- **IoU:** This metric measures the overlap between the predicted and the ground truth bounding boxes. IoU is used by the MaxIoUAssigner in the detection head to assign the proposals from the RPN to the ground truth bounding boxes based on a predefined IoU threshold. Additionally, IoU is applied by the Non-Maximum Suppression (NMS) algorithm that uses an IoU threshold to eliminate the overlapping bounding boxes and retain only the highest-confidence detections for each object. IoU is calculated as:

$$\text{IoU} = \frac{A^{\text{GT}} \cap A^{\text{BB}}}{A^{\text{GT}} \cup A^{\text{BB}}} \tag{4}$$

where A^{GT} and A^{BB} are the ground truth and the predicted bounding box areas, respectively.

- **Precision:** This metric represents the accuracy of the model in identifying the true positives. It is calculated as the ratio of the true positives to the sum of the true positives and the false positives:

$$\text{Precision} = \frac{\#\text{True Positives}}{\#\text{True Positives} + \#\text{False Positives}} \tag{5}$$

- **Recall:** This metric measures the ability of the model to identify all the relevant instances. It is computed as the ratio of the true positives to the sum of the true positives and the false negatives:

$$\text{Recall} = \frac{\#\text{True Positives}}{\#\text{True Positives} + \#\text{False Negatives}} \tag{6}$$

- **AP:** This metric measures the average precision over different levels of recall. AP provides a summary of the ability of the model to detect objects across various thresholds of recall. AP is computed as:

$$\text{AP} = \int_0^1 P(R)dR \tag{7}$$

where R is the recall, and $P(R)$ is the precision-recall curve. This curve illustrates the trade-off between precision and recall by plotting their values as the classification threshold, which determines what the model considers a positive prediction, is adjusted.

- **mAP:** The mean Average Precision aggregates the AP scores across all the classes in the dataset to provide a comprehensive performance indicator:

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i \tag{8}$$

where N is the number of classes.

Furthermore, the Frames Per Second (FPS) metric is used to evaluate the detection speed of the models during inference. Floating Point Operations per Second (FLOPS), and the number of model parameters are used to analyze the computational complexity of the model. The higher the FPS, the faster the processing speed of the model, while lower FLOPS and fewer parameters are indicators of a more lightweight model.

4.3. Implementation Details

The models are implemented using the MMRotate (Zhou et al., 2022b) toolkit and PyTorch. The models are trained on a single NVIDIA RTX 4000 SFF Ada Generation GPU with a batch size of 4. The detection pipeline uses ResNet-50 (He et al., 2016) that has been pre-trained on ImageNet (Deng et al., 2009) as the backbone feature extractor. This backbone is enhanced with context blocks, applied after the first convolutional layer of each residual stage. The context block uses a channel reduction ratio of 0.25.

The models are trained for 12 epochs with an initial learning rate of 0.0001 on both TinyDOTA and ExcaSat. Following Li et al. (2024c), the AdamW (Loshchilov, 2017) optimizer is used. AdamW applies weight decay decoupled from the gradient-based update steps, which prevents overfitting by penalizing the large weight values. The optimizer uses betas of 0.9 and 0.999 and a weight decay of 0.05. Gradient clipping is configured with a maximum L2 norm of 35. The L2 norm calculates the Euclidean length of the gradient vector to ensure it stays within the maximum norm. Gradient clipping is used to limit the gradient values during the training, which prevents gradient from exploding. The learning rate schedule follows a step policy, with a linear warm-up during the first 500 iterations, and starting from a warm-up ratio of 0.333. After epochs 8 and 11, the learning rate is decreased by a factor of 10.

During testing and inference, the detection pipeline applies a two-stage filtering process to handle cluttered predictions. In the first stage, oriented RPN proposals, often highly overlapping, are filtered using NMS with an IoU threshold of 0.8, retaining a maximum of 2000 proposals per FPN level. These proposals are merged, and the top 2000 proposals, ranked by their classification scores, are passed to the second stage. In the second stage, NMS is applied for each class on the oriented bounding boxes with class probabilities above 0.05, using an IoU threshold of 0.1. Up to 2000 bounding boxes are retained following the MMRotate toolkit guidelines, and a maximum of 2000 detections per image are allowed. All the ablation studies are conducted on the TinyDOTA dataset, while the analyses on the hyperparameters of OE-Net are conducted on the ExcaSat dataset.

4.4. Comparison with State-of-the-art Methods

OE-Net has been trained and tested using both TinyDOTA and ExcaSat. In the case of TinyDOTA, OE-Net is compared to twelve state-of-the-art object detectors that have been trained and tested on TinyDOTA using the same hyperparameter settings as OE-Net. With ExcaSat, OE-Net is compared with Oriented R-CNN, which is the baseline method that OE-Net builds upon. The comparison on TinyDOTA is used to validate the added performance in terms of precision from the extra modules that OE-Net integrates. The comparison on ExcaSat is used to validate the applicability of OE-Net in the real-world use case of pipeline monitoring using satellite images.

4.4.1. Results on TinyDOTA

Table 4 illustrates the test results of twelve state-of-the art models and OE-Net, each of them trained on TinyDOTA. Benefiting from the novel approaches in the feature pyramid, the proposal network, and the detection head, OE-net outperforms the state-of-the-art detectors, scoring an mAP of 0.687. OE-Net not only achieves the highest mAP but also sets the highest AP score for four classes (bridge, small-vehicle, ship, and storage-tank), the second-highest AP score for three classes (large-vehicle, swimming-pool, and helicopter), and the third-highest AP score for one class (plane). The mAP consistency of OE-Net across all eight classes indicates its adaptability to varying object scales. Furthermore, the 0.569 mAP score on the small-vehicle class, which is both the smallest and the most abundant class in the dataset, indicates that OE-Net outperforms the state-of-the-art when it comes to TOD. Thus, the additional modules in OE-Net are particularly beneficial for TOD.

OE-Net demonstrates a significant improvement over the baseline Oriented R-CNN (Xie et al., 2021), outperforming the model by 6.3 percentage points in mAP. This highlights the impact of the additional modules integrated in OE-Net on enhancing the feature discrimination capabilities, the anchor matching process, and the classification approach. The substantial leap of 14.3 percentage points in the AP for the small-vehicle underscores the efficacy of OE-Net in the task of TOD in satellite imagery.

Notably, OE-Net outperforms the recently published LSKNet (Li et al., 2024c), which achieves the second-highest mAP of 0.672. While both OE-Net and LSKNet build upon Oriented R-CNN, the two models address the challenge of semantic confusion at different stages of the detection pipeline. LSKNet focuses on mitigating semantic confusion in the feature extraction phase by replacing the backbone, and incorporating dynamic receptive fields to enhance the contextual understanding. In contrast, OE-Net addresses semantic confusion in the detection phase through the vector mapping method. This modular approach allows OE-Net to build upon Oriented R-CNN without replacing the backbone. While OE-Net requires half the training time of LSKNet, OE-Net outperforms LSKNet by 1.5 percentage points in mAP.

ReDet (Han et al., 2021b) and STD (Yu et al., 2024) yielded the third and the fourth best mAP scores, respectively. ReDet uses a rotation-invariant RoI alignment mechanism, and a rotation-equivariant backbone. The RoI alignment module performs two key operations: spatial alignment to warp the RoIs from the feature maps, and orientation alignment to ensure that the features are invariant to rotation. This capability to extract invariant features enhances the detection precision in TOD, where unseen satellite images contain objects with arbitrary orientations.

STD is the only state-of-the-art model in Table 4 that employs a transformer backbone instead of a CNN backbone. Transformer backbones use self-attention mechanisms to model the long-range dependencies and the global context

Table 4
The AP scores of twelve models on the eight different object classes in TinyDOTA and the corresponding mAP for all the classes. The blue score denotes the highest score, and the red score denotes the second-highest score. The abbreviated classes are: plane (pl), bridge (br), small-vehicle (sv), large-vehicle (lv), ship (sh), storage-tank (st), swimming-pool (sp), and helicopter (hc).

Model	pl	br	sv	lv	sh	st	sp	hc	mAP
R3Det (Yang et al., 2021a)	0.785	0.348	0.414	0.689	0.796	0.607	0.518	0.474	0.579
LSKNet (Li et al., 2024c)	0.810	0.488	0.505	0.801	0.812	0.611	0.659	0.690	0.672
RoI Transformer (Ding et al., 2019)	0.808	0.426	0.428	0.743	0.805	0.615	0.578	0.593	0.625
Oriented R-CNN (Xie et al., 2021)	0.804	0.424	0.426	0.752	0.805	0.610	0.571	0.600	0.624
Gliding Vertex (Xu et al., 2020)	0.799	0.366	0.422	0.701	0.790	0.611	0.531	0.551	0.597
DCFL (Xu et al., 2023)	0.800	0.351	0.423	0.700	0.764	0.634	0.549	0.494	0.590
S2A-Net (Han et al., 2021a)	0.790	0.343	0.450	0.718	0.801	0.679	0.562	0.479	0.603
Oriented RepPoints (Li et al., 2022)	0.801	0.382	0.512	0.717	0.822	0.674	0.559	0.446	0.614
Rotated RetinaNet (Lin et al., 2017b)	0.790	0.236	0.295	0.597	0.693	0.582	0.503	0.374	0.509
Rotated FCOS (Tian et al., 2019)	0.799	0.325	0.467	0.745	0.805	0.650	0.574	0.466	0.604
ReDet (Han et al., 2021b)	0.810	0.425	0.503	0.747	0.809	0.615	0.624	0.599	0.642
STD (Yu et al., 2024)	0.806	0.436	0.489	0.758	0.809	0.614	0.631	0.557	0.637
OE-Net	0.806	0.499	0.569	0.764	0.890	0.700	0.654	0.613	0.687

within images. Standard transformer backbones struggle in rotation-sensitive tasks due to their lack of explicit mechanisms for spatial transformations. STD addresses this limitation by introducing a decoupled approach for predicting oriented bounding boxes, leveraging the hierarchical nature of transformer backbones to handle the rotation and the scale variations. The drawback of STD is its high computational cost. STD takes four times as long to train compared to OE-Net, which limits its applicability to use cases such as pipeline monitoring.

Rotated RetinaNet (Lin et al., 2017b) and R3Det (Yang et al., 2021a) yield the lowest detection precision with an mAP of 0.509 and 0.579, respectively. Both models do not utilize an RPN. Instead, the anchors are directly assigned to the ground-truth bounding boxes based on their IoU scores. Each anchor is matched to at most one object. In scenes with densely packed tiny objects, this assignment approach leads to the underrepresentation of positive labels for tiny objects, which in turn leads to a reduction in detection precision on datasets such as TinyDOTA.

4.4.2. *Results on ExcaSat*

Table 5 shows the test results of eight state-of-the-art models and OE-Net, each of them trained on ExcaSat. OE-Net outperforms the other models, yielding an mAP score of 0.593. OE-Net achieves the highest AP score for the small-vehicle class and the excavator class, and it achieves the second-highest AP score for the large-vehicle class. OE-Net improves the mAP score of the Oriented R-CNN baseline by 11.1 percentage points. Furthermore, it improves the AP score for the excavator class, which is the most import class in pipeline monitoring, by 14 percentage points. The ExcaSat dataset excludes certain object classes that are in TinyDOTA, such as the bridge class, the storage-tank class, and the swimming-pool class. As a result, the average object size in ExcaSat is smaller than in TinyDOTA. Interestingly, OE-Net shows an even greater improvement over both the Oriented R-CNN baseline and LSKNet on ExcaSat than on TinyDOTA. This suggests that the additional modules in OE-Net are particularly effective in addressing the challenges that are posed by the smallest instances in the dataset. Specifically, the worse the feature representations of the objects, the smaller the overlap with the predefined anchors, and the greater the semantic confusion with similar looking background instances, the higher the relative increase in performance by OE-Net as compared to the other state-of-the-art methods.

Table 5
The AP scores of Oriented R-CNN (Xie et al., 2021) and the proposed OE-Net on the three different object classes in ExcaSat and the corresponding mAP for all the classes. The blue score denotes the highest score, and the red score denotes the second-highest score. The abbreviated classes are: small-vehicle (sv), large-vehicle (lv), and excavator (ex).

Model	sv	lv	ex	mAP
Oriented R-CNN	0.514	0.571	0.361	0.482
Rotated FCOS	0.550	0.478	0.387	0.472
S2A-Net	0.518	0.365	0.264	0.392
LSKNet	0.606	0.596	0.479	0.560
RoI Transformer	0.511	0.463	0.403	0.459
ReDet	0.518	0.545	0.401	0.488
Oriented RepPoints	0.631	0.538	0.369	0.513
STD	0.478	0.444	0.385	0.436
OE-Net	0.691	0.586	0.501	0.593

4.5. Ablation Study

To evaluate the individual contributions of the additional modules in OE-Net, an ablation study was conducted using the TinyDOTA dataset. Each module was integrated independently into the baseline Oriented R-CNN architecture to assess its impact on the detection precision. This approach

Table 6
The mAP scores of the individual additional modules when integrated with the baseline Oriented R-CNN architecture. The utmost right column represents OE-Net, integrating each of the individual modules into one TOD detector. The blue score denotes the highest score, and the red score denotes the second-highest score.

Oriented R-CNN (Xie et al., 2021)	✓	✓	✓	✓	✓	✓	✓	✓
Dilated Fusion		✓						✓
Balanced Ranking Assignment			✓					✓
Vector Mapping				✓				✓
Context Blocks (Cao et al., 2019)					✓			✓
PolyRandomRotate (Zhou et al., 2022b)						✓		✓
AdamW (Loshchilov, 2017)							✓	✓
mAP	0.624	0.627	0.636	0.628	0.631	0.640	0.637	0.687

enables the evaluation of each component individually, ensuring that the performance improvement is attributed to the module itself rather than to its presence within a combination of modules. Table 6 shows the results of the ablation study.

4.5.1. *Effect of Dilated Fusion*

Integrating the Dilated Fusion module into the baseline Oriented R-CNN detector improved the mAP from 0.624 to 0.627, demonstrating a positive impact on detection precision. This result validates that incorporating a residual connection to the deepest feature map produced by the backbone, refining it, and merging it with the lateral convolution of the original feature map enhances attention to spatial details. This enhancement proves particularly beneficial in TOD scenarios, where objects are tiny and feature representations are sparse. In essence, extending the FPN provides a performance advantage over the standard FPN in TOD.

4.5.2. *Effect of Balanced Ranking Assignment*

The BRA module yields a significant improvement, raising the mAP to 0.636 and demonstrating its effectiveness in enhancing anchor assignment for tiny objects. This result underscores its superiority over IoU-based anchor matching. For tiny objects, even small positional shifts in predefined anchors lead to disproportionately large changes in the IoU metric. This results in IoU-based methods favoring positive label assignment to larger objects and causing tiny objects to be underrepresented in the detection head downstream. In contrast, NWD-based ranking assignment mitigates this sensitivity to overlap changes, offering a more robust method for anchor matching in TOD.

BRA incorporates both objectness and NWD into the ranking metric to address scenarios where NWD scores are very similar, making it difficult to confidently select the top-k anchors. By combining NWD and objectness, the module ensures more reliable anchor assignment in ambiguous cases. To validate the added benefit of objectness to NWD-based ranking, a comparative study was conducted between IoU-based, NWD-based, and NWD with objectness-based assignment. As shown in Table 7, the NWD-based Ranking Assigner outperforms the Maximum IoU Assigner, and the inclusion of objectness in BRA further enhances the precision beyond purely NWD-based assignment.

Table 7
The AP and the mAP scores of OE-Net with three different assigners on the ExcaSat dataset. The blue score denotes the highest score, and the red score denotes the second-highest score. The abbreviated classes are: small-vehicle (sv), large-vehicle (lv), and excavator (ex).

Assigner	**sv**	**lv**	**ex**	**mAP**
Maximum IoU Assigner	0.584	0.623	0.475	0.561
Ranking Assigner	0.690	0.575	0.451	0.572
Balanced Rank Assigner	0.691	0.586	0.501	0.593

4.5.3. *Effect of Vector Mapping*

The addition of Vector Mapping improved the mAP to 0.628. This enhancement demonstrates that the traditional fully connected layer in the classification branch of the baseline Oriented R-CNN detector, which uses trainable weights to map the feature space to a set of class probabilities for each RoI, can be replaced with a fixed set of class vectors. These vectors, with a dimensionality that matches the feature space and the number of classes, allow class probabilities to be computed simply by calculating the cosine similarity between the extracted features and the fixed vectors. By eliminating the learning component from the classification branch, the upstream modules are forced to produce semantically distinct feature maps for each class. This approach is particularly beneficial for TOD, where different classes are often visually similar.

4.5.4. *Effect of Context Blocks*

Integrating Context Blocks increased the mAP to 0.631. Context Blocks are incorporated into the residual layers of the ResNet-50 backbone to model the long-range dependencies and capture the global context within a scene. This method, proposed by Cao et al. (2019), was designed to enhance global understanding rather than focusing on local features. Although not specifically aimed at TOD in satellite imagery, the performance improvement demonstrates that, in TOD, where object feature representations are limited, capturing a global understanding of the scene is beneficial. For example, in the case of excavators, global context such as the surrounding dirt or brown land provides valuable a association cue, even when the individual features of the excavator are sparse or ambiguous.

4.5.5. *Effect of PolyRandomRotate*

Incorporating PolyRandomRotate provided the greatest standalone improvement, increasing the mAP to 0.640. In the context of oriented object detection in satellite images, where objects appear at various orientations, augmenting the dataset with random rotations is particularly valuable. By introducing random rotations, the model is forced to learn features that are invariant to orientation, ensuring that the object detector accurately detects and classifies objects regardless of their orientation. PolyRandomRotate enables the model to map features to classes and bounding boxes in a way that is not biased toward a specific orientation. As a result, this augmentation leads to better generalization, improving the detection precision on unseen images where the orientations of objects will vary significantly.

4.5.6. *Effect of AdamW*

Switching the optimizer from the traditionally used SGD to AdamW improved the mAP to 0.637. While SGD relies on a fixed learning rate schedule and uses momentum to accelerate convergence, AdamW incorporates adaptive learning rates for individual parameters and decouples the weight decay from the gradient updates. In object detection, models need to optimize not only for the classification task but also for the regression of the bounding boxes. This involves balancing multiple loss components. The adaptive learning rates in AdamW allow it to adjust the step size for each parameter based on the magnitude of its gradient, ensuring faster and more stable convergence. Additionally, by decoupling the weight decay, AdamW prevents the over-regularization of parameters that have already been optimized. Furthermore, AdamW is better equipped to handle sparse gradients that are common when detectors are trained on high-dimensional feature maps, such as in TOD.

The final OE-Net architecture, integrating all the modules, achieved the highest mAP of 0.687, surpassing the Oriented R-CNN baseline by 6.3 percentage points. This result validates the complementary nature of the modules and their combined effectiveness in addressing the aforementioned challenges of TOD.

4.6. Model Analysis

4.6.1. *Anchor Scales*

Table 8 shows the results of the comparative study of different anchor scales that are used by the anchor generator as part of the RPN. On the ExcaSat dataset, OE-Net achieves the highest mAP score of 0.603 with an anchor scale of 6. For the excavator class, OE-Net achieves the highest AP score of 0.501 with an anchor scale of 8.

Table 8
The AP and the mAP scores of OE-Net with five different anchor size settings on the ExcaSat dataset. The blue score denotes the highest score, and the red score denotes the second-highest score. The abbreviated classes are: small-vehicle (sv), large-vehicle (lv), and excavator (ex).

Anchor Scale	sv	lv	ex	mAP
2	0.685	0.596	0.444	0.575
4	0.755	0.615	0.394	0.588
6	0.750	0.622	0.435	0.603
8	0.691	0.586	0.501	0.593
10	0.686	0.571	0.460	0.573

The area of an anchor in a particular feature map is equal to the product of the anchor scale and the stride of that feature map. For the DOTA dataset, an anchor scale of 8 is typically selected. For the comparison on the TinyDOTA dataset, an anchor scale of 8 was used. For the comparison on the ExcaSat dataset, which has a relatively smaller average object size, the results in Table 8 indicate that choosing an anchor scale of 6 is beneficial for the mAP score of OE-Net. However, the excavator detection precision is critical in pipeline monitoring, and therefore the anchor scale of 8 is selected. Furthermore, it is worth to be noted that OE-Net achieves higher mAP scores than the other state-of-the-art methods regardless of which anchor scale is being used.

4.6.2. *Hyperparameters*

Table 9 summarizes the results of the comparative study on the dilation combinations that are used by the Dilated Fusion module as part of the FPN. The choice of dilation rates influences the balance between local and global context captured by the model. For example, the combination of dilations (1, 2, 3) results in a significant overlap between the three receptive fields, emphasizing the fine-grained local features but limiting the capture of broader spatial relationships. In contrast, a combination such as (1, 5, 9) prioritizes capturing the global context at the expense of the local context.

The results in Table 9 indicate that the dilation combination of (1, 3, 5) yields the highest mAP score as well as the highest AP score for the excavator class. This dilation combination provides a trade-off by covering both the local and the global context in the deepest feature map. Moreover, all the dilation combinations outperform the other state-of-the-art models in terms of mAP.

In the Balanced Ranking Assigner, there are two hyperparameters that can be tuned. As can be seen in Equation 1, α determines the extent to which the NWD metric influences the ranking metric as opposed to the objectness metric. Moreover, the top-K value determines how many of the top ranked proposals will receive positive labels. In Table 10, the results of the comparative study on the selection of α and top-K are shown. An α of 7 and a top-K of 2 yield the

Table 9
The AP and the mAP scores of OE-Net with four different dilation combinations on the ExcaSat dataset. The blue score denotes the highest score, and the red score denotes the second-highest score. The abbreviated classes are: small-vehicle (sv), large-vehicle (lv), and excavator (ex).

Dilation Combination	sv	lv	ex	mAP
1, 2, 3	0.688	0.579	0.449	0.572
1, 3, 5	0.691	0.586	0.501	0.593
1, 4, 7	0.685	0.602	0.438	0.575
1, 5, 9	0.689	0.591	0.472	0.584

Table 10
The AP and the mAP scores of OE-Net with different values for α and top-K on the ExcaSat dataset. The blue score denotes the highest score, and the red score denotes the second-highest score. The abbreviated classes are: small-vehicle (sv), large-vehicle (lv), and excavator (ex).

α	top-K	sv	lv	ex	mAP
4	2	0.690	0.613	0.472	0.592
5		0.691	0.608	0.438	0.579
6		0.691	0.586	0.501	0.593
7		0.690	0.625	0.468	0.594
8		0.694	0.582	0.408	0.562
6	4	0.686	0.580	0.450	0.572
	6	0.684	0.592	0.495	0.590
	8	0.678	0.591	0.451	0.573

highest mAP score, while an α of 6 and a top-K of 2 yield the highest AP score for the excavator class.

Table 10 shows that with a top-K of 2, the AP for the small-vehicle class remains relatively stable across different values of α. This suggests that the objectness score plays a consistent role in the assignment process, regardless of whether its influence is weighted heavily or lightly relative to the NWD metric. In the case of the excavator class and the large-vehicle class, however, the AP varies across different values of α. These two classes are less abundant compared to the small-vehicle class and are less frequently found in densely packed scenarios. As a result, the predefined anchors struggle to align with these objects, leading to a more significant influence of the objectness score during the assignment process. Furthermore, Table 10 indicates that passing more than the top-2 ranked anchors to the detection head does not increase the detection precision.

Table 11
The AP and the mAP scores of OE-Net with four different epoch counts used by the Vector Mapping module for the pre-training of the feature vectors on the ExcaSat dataset. The blue score denotes the highest score, and the red score denotes the second-highest score. The abbreviated classes are: small-vehicle (sv), large-vehicle (lv), and excavator (ex).

Epochs	sv	lv	ex	mAP
10	0.743	0.569	0.467	0.593
20	0.691	0.586	0.501	0.593
30	0.692	0.577	0.459	0.576
40	0.693	0.618	0.461	0.590
50	0.736	0.553	0.500	0.596

Table 11 summarizes the comparative study of the number of epochs that are used to pre-train the feature vectors for the Vector Mapping module. The higher the number of epochs, the more cosine dissimilarity is imposed upon the feature vectors. The results in Table 11 suggest that there is no consistent correlation between a higher cosine dissimilarity and improved detection precision. Specifically, increasing the number of pre-training epochs from 20 to 30 and 40 results in a decline in mAP, whereas extending the training to 50 epochs leads to an improvement. This indicates that the primary value of the feature vectors lies in their random initialization. This provides a sufficiently diverse set of vectors, each representing distinct directions in the feature space. While pre-training helps to mitigate the risk of two vectors being directionally similar, increasing the number of epochs, and thereby imposing a greater cosine dissimilarity, does not consistently enhance the detection precision.

4.7. Discussion

4.7.1. Qualitative Analysis

Figure 15 shows a selection of the detection results of OE-Net on the TinyDOTA dataset. The images in TinyDOTA comprise a number of complicating aspects, such as the occlusion of objects, variations in object sizes, cluttered object scenes, and complex backgrounds. Similar to the quantitative results, the detection results indicate that OE-Net is capable of accurately detecting objects at various scales and orientations. The bottom-right image shows that OE-Net captures densely packed objects, such as densely parked cars. Furthermore, the top-left image shows that OE-Net successfully distinguishes between semantically similar classes, such as boats and cars. In the top-right image, the objects blend into the complex background, causing OE-Net to falsely detect a number of background instances.

Figure 16 shows the inference results of OE-Net on four images from the ExcaSat test dataset. The images demonstrate that OE-Net is capable of accurately detecting the excavators across various orientations, and against different backgrounds. A distinguishing feature of excavators is their crane, which sets them apart from other vehicle types. In the bottom-left image, OE-Net accurately detects both an excavator with an extended crane and an excavator with its crane retracted. This suggests that the learned feature representation for excavators extends beyond the crane, and captures a more comprehensive understanding of the overall shape, color, size, and global context of the object. The bottom-right image in Figure 16 demonstrates how OE-Net accurately detects an excavator that is obscured by shadows, and surrounded by many other tiny object instances.

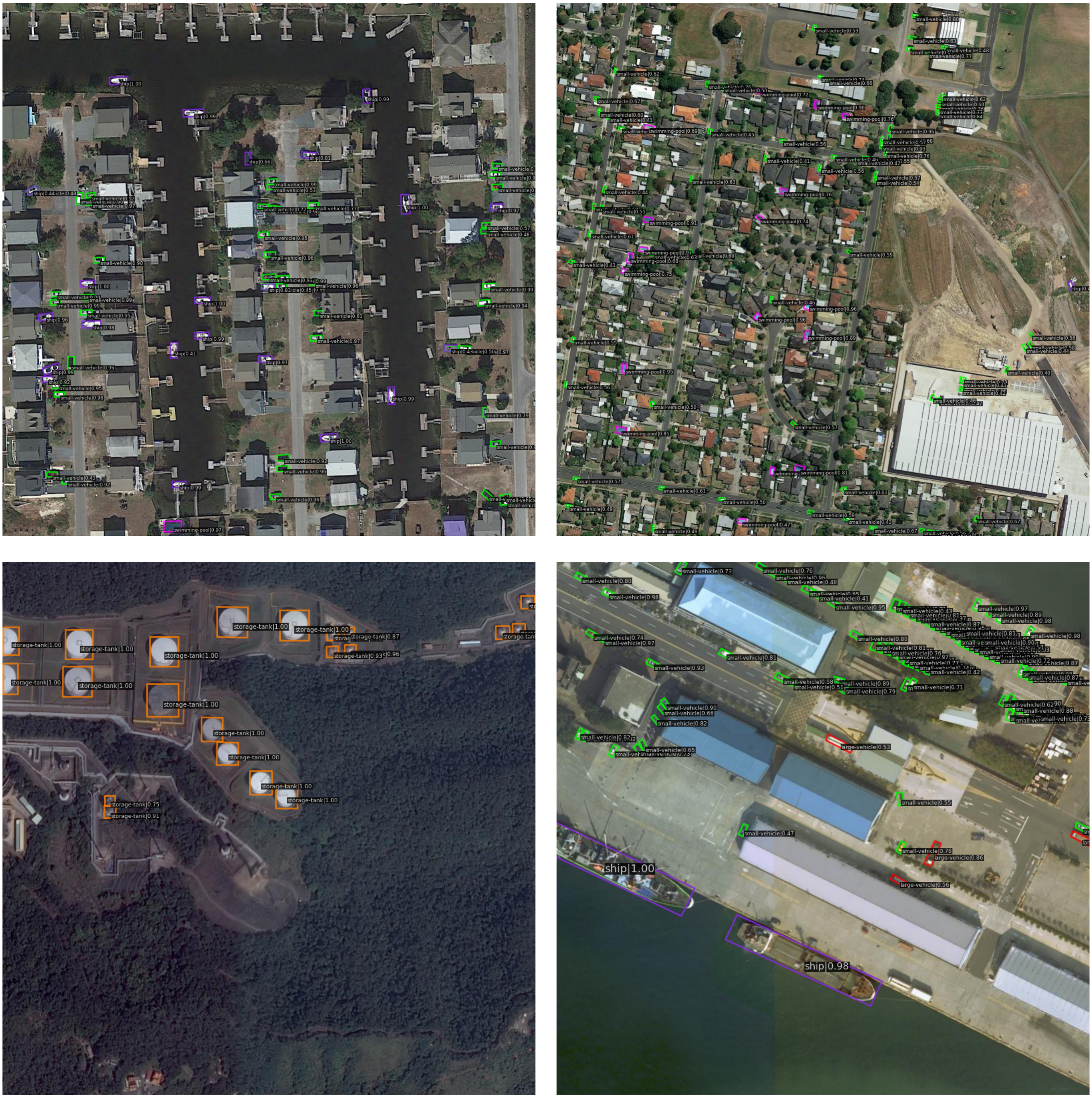

Figure 15: A collection of inference results of OE-Net on the TinyDOTA dataset. The green bounding boxes denote the small-vehicle class, the red bounding boxes denote the large-vehicle class, the orange bounding boxes denote the storage-tank class, the purple bounding boxes denote the ship class, and the pink bounding boxes denote the swimming-pool class.

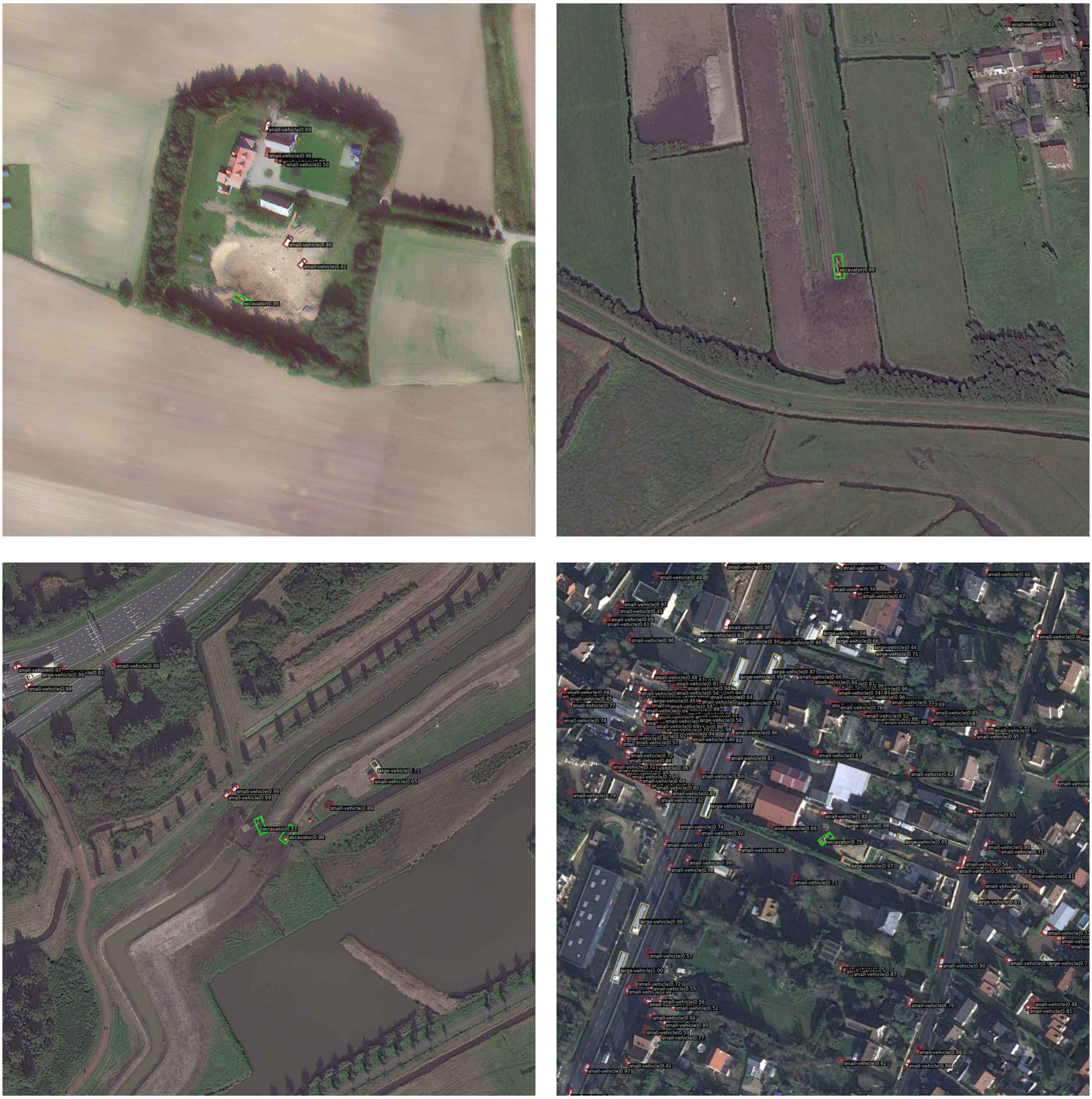

Figure 16: A collection of inference results of OE-Net on the ExcaSat dataset. The green bounding boxes denote the excavator class, the red bounding boxes denote the small-vehicle class, and the yellow bounding boxes denote the large-vehicle class. The images are acquired from © Airbus DS (2024).

4.7.2. *Confusion Matrix Analysis*

A confusion matrix summarizes the performance of the classification of the objects. It is used to identify the causes of the misclassifications. For example, it highlights which object classes are mistaken for other object classes or incorrectly classified as background instances. Figure 17 shows the confusion matrix of the test results of OE-Net on TinyDOTA. The matrix indicates that only in two cases, there is semantic confusion among classes: 7% of the large-vehicle ground truths are labeled as the small-vehicle, and 20% of the helicopter ground truths are labeled as planes.

The semantic confusion between the large-vehicle class and the small-vehicle class occurs when the size of the vehicle is the size of a camper van. In this case, the model associates not only the semantic features such as the shape and the color, but also the size with both the object classes. Vehicles that are a fraction smaller (e.g., vans) belong to the small-vehicle class, while vehicles that are a fraction bigger (e.g., trucks) belong to the large-vehicle class.

The bottom row in the confusion matrix represents the false positive detections, and indicates the classes that were associated with the background instances. Figure 17 shows that 71% of the false positives are labeled as the small-vehicle class. Since this class is the most abundant class, and has the most limited feature representation due to its size and shape, OE-Net confuses background instances, such as road surface markings and air-conditioning units, with the small-vehicle class.

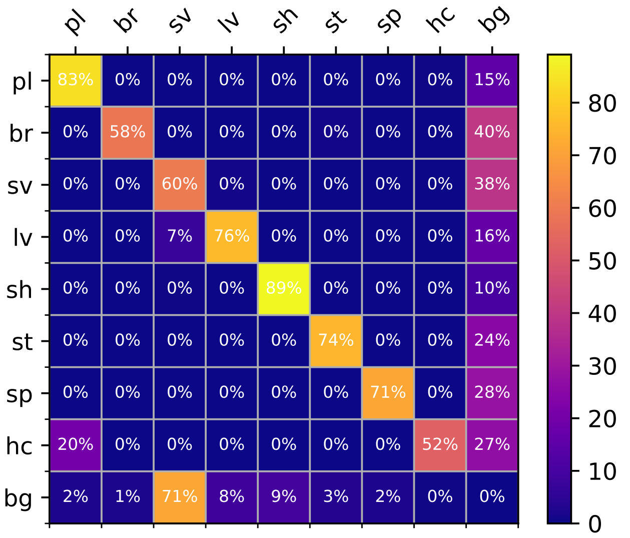

Figure 17: A normalized confusion matrix based on the test results of OE-Net on the TinyDOTA dataset. The rows represent the ground truth labels, and the columns represent the predicted labels. The diagonal percentages represent the correct predictions, and the off-diagonal percentages represent the misclassifications. The abbreviated classes are: plane (pl), bridge (br), small-vehicle (sv), large-vehicle (lv), ship (sh), storage-tank (st), swimming-pool (sp), helicopter (hc), and background (bc).

The rightmost column in the confusion matrix represents the false negative detections, or the missed detections. Object classes with a high recall tend to have a low number of false negatives. Figure 17 shows that this is the case for the plane class, the large-vehicle class, and the ship class. However, the bridge class and the small-vehicle class indicate a high number of false negatives and thus a lower recall. The bridge class is the second least abundant class in the TinyDOTA dataset, and is the largest class in terms of object size. Therefore, the class is not targeted as much by the additional modules in OE-Net that specifically focus on TOD. Conversely, the small-vehicle class suffers from a higher number of false negative detections due to challenges such as occlusion and limited feature representations.

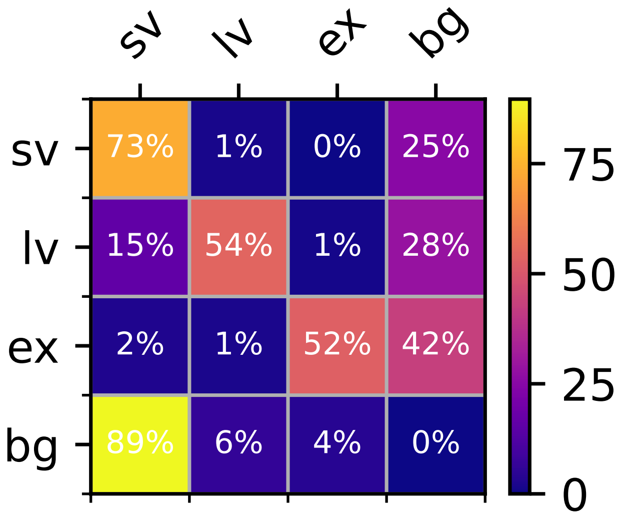

Figure 18: A normalized confusion matrix based on the test results of OE-Net on the ExcaSat dataset. The rows represent the ground truth labels, and the columns represent the predicted labels. The diagonal percentages represent the correct predictions, and the off-diagonal percentages represent the misclassifications. The abbreviated classes are: small-vehicle (sv), large-vehicle (lv), excavator (ex), and background (bc).

Figure 18 shows the confusion matrix of the test results of OE-Net on ExcaSat. The matrix indicates that 42% of the excavator ground truths are labeled as background instances. The ExcaSat test dataset consists of both 30cm and 50cm spatial resolution images. Figure 16 demonstrates that OE-Net is capable of accurately detecting excavators in 30cm images. In 50cm images, however, the excavators completely blend in with the background, and OE-Net is hardly able to detect any of the ground truths instances.

Furthermore, Figure 18 indicates that 15% of the ground truth large-vehicle instances are classified as small-vehicle instances. This is in part due to the inconsistency in labeling ambiguously sized vehicles during the annotation process of the pipeline monitoring images. This inconsistency has resulted in inter-class confusion during both the training process and during inference.

4.7.3. Precision vs. Recall Analysis

Figures 19 and 20 show the Precision-Recall curves of OE-Net on the TinyDOTA dataset and on the ExcaSat dataset, respectively. The curves illustrate the trade-off between the precision and the recall at a certain IoU threshold, which is 0.5 in the case of OE-Net. Precision measures the proportion of true positives out of all the positive predictions, including the false positive predictions. Hence, a high precision indicates that most of the positive predictions are correct. Recall measures the proportion of true positives out of all the positive instances, including the false negative predictions. Hence, a high recall indicates that the model has few missed detections of ground truth positives. In pipeline monitoring, finding a balance between the precision and the recall is crucial. A high precision indicates a low number of false positives, minimizing disturbances for the pipeline operators caused by false alarms. In contrast, a high recall indicates a low number of false negatives, ensuring that most of the critical instances of potential threats are detected. From a safety perspective, false negatives (i.e., missed detections) pose a greater risk to the pipeline safety than false positives, making it imperative to maintain a robust minimum recall threshold.

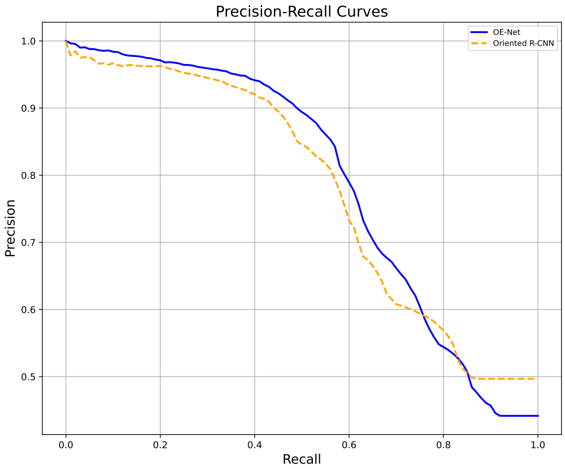

Figure 19: The Precision-Recall curve of OE-Net (blue) and Oriented R-CNN (orange) on TinyDOTA at an IoU of 0.5.

Figure 19 demonstrates that OE-Net achieves a superior precision-recall trade-off compared to the Oriented R-CNN baseline, as evidenced by its curve extending closer to the top-right corner. Notably, at a precision of 0.9 (i.e., 90% of the detections are ground truth instances), the recall is 0.5 (i.e., 50% of the ground truth instances are detected). Table 12 presents the performance of OE-Net on TinyDOTA, and indicates that the recall consistently exceeds the precision across all classes. This is advantageous for the pipeline safety, as it minimizes the false negatives. However, the substantial disparity between the number of detections and the number of ground truths across all classes reflects a high false positive rate. This is likely due to the high variability of the satellite images and the presence of background instances that strongly resemble the ground truth class objects.

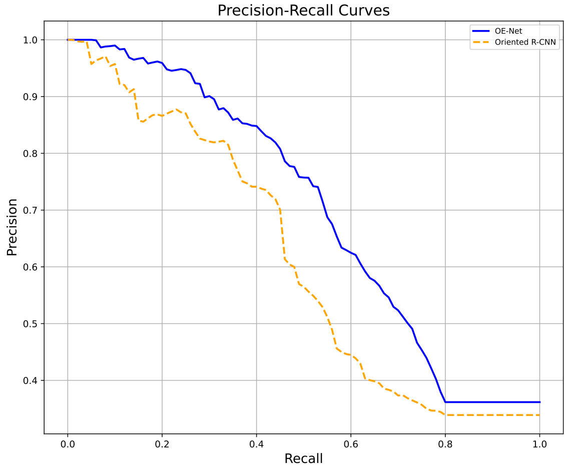

Figure 20: The Precision-Recall curve of OE-Net (blue) and Oriented R-CNN (orange) on ExcaSat at an IoU of 0.5.

Table 12
The detection results of OE-Net on the TinyDOTA test dataset. GTS represents the number of ground truths, and Dets represents the number of detections.

Class	GTS	Dets	Recall	AP
plane	5457	6899	0.860	0.806
bridge	801	2603	0.635	0.499
small-vehicle	70950	97407	0.652	0.569
large-vehicle	8729	19719	0.887	0.764
ship	24983	31082	0.912	0.890
storage-tank	4526	6449	0.789	0.700
swimming-pool	1323	2842	0.756	0.654
helicopter	139	714	0.777	0.613
mAP				0.687

Table 13
The detection results of OE-Net on the ExcaSat test dataset. GTS represents the number of ground truths, and Dets represents the number of detections.

Class	GTS	Dets	Recall	AP
small-vehicle	6260	9147	0.798	0.691
large-vehicle	360	1116	0.781	0.586
excavator	149	324	0.624	0.501
mAP				0.593

Figure 20 illustrates that OE-Net achieves a significantly better precision-recall trade-off than the Oriented R-CNN baseline. Table 13 shows that OE-Net achieves a high recall on the small-vehicle class and the large-vehicle class in ExcaSat. For the excavator class, which is the most critical class for pipeline monitoring, OE-Net achieves a recall of 0.624 and an AP of 0.501. For excavators, the model identifies 93 true positives out of 125 ground truths, misses 56 instances, and produces 231 false positives. These results highlight the need for further refinement in detecting the excavator class.

5. Conclusion

This paper introduces OE-Net, a modular extension of Oriented R-CNN designed for tiny object detection (TOD) in satellite imagery. OE-Net addresses three critical challenges in TOD: information loss in the feature fusion stage, insufficient positives labels for tiny objects in the proposal stage, and semantic confusion between foreground and background instances in the classification stage. OE-Net tackles these challenges by integrating a Dilated Fusion (DF) module, a Balanced Ranking Assigner (BRA), and a Vector Mapping (VM) module. The DF module enriches the spatial context of the deepest feature map using dilated convolutions and adaptive spatial fusion. The BRA refines anchor assignment by incorporating the Normalized Wasserstein Distance (NWD) and the objectness score into a ranking-based assigner. The VM module replaces the fully connected layer in the classification branch with a pre-trained set of feature vectors. The VM module then maps the extracted features to a set of class probabilities by calculating the cosine similarity between the extracted features and the pre-trained feature vectors. OE-Net outperforms state-of-the-art methods, setting new benchmarks on the TinyDOTA dataset and the newly proposed ExcaSat dataset for pipeline monitoring. Importantly, the modularity of OE-Net allows its components to be integrated seamlessly into other anchor-based two-stage object detectors.

Despite these advances, OE-Net has limitations that underscore the challenges of TOD in satellite imagery. The model exhibits a relatively high false positive rate. This is primarily due to the complex backgrounds in satellite imagery and the semantically ambiguous background objects that closely resemble ground truth objects. For example, frequent background instances, such as air-conditioning units on top of buildings and road surface markings, are confused with the small-vehicle class and the large-vehicle class. Inter-class confusion persists between semantically similar classes, and is particularly evident for vehicles whose sizes place them between the small-vehicle class and the large-vehicle class, such as camper vans. These vehicles often share features with both classes, leading the model to misclassify them. Furthermore, while OE-Net achieves promising results for pipeline monitoring, the detection of excavators remains sensitive to the spatial resolution of the images. At resolutions coarser than 30 cm, objects tend to blend into the background, leading to missed detections. Additionally, the availability of satellite images containing excavators is limited, restricting the potential to generalize across a wider range of scenarios.

The findings of this paper open several avenues for future research. First, the development of a larger and a more diverse excavator dataset is crucial for advancing satellite-based pipeline monitoring. Second, the fixed feature vectors used in the VM module suggest the need for further investigation to determine whether alternative pre-training strategies could better capture the semantic characteristics of specific classes. The results suggest that replacing the traditional fully connected layers for classification is beneficial but requires further exploration into the behavior and the optimization of the predefined feature vectors. Third, the integration of the objectness scores into the anchor assignment process in the BRA demonstrates potential in exploring more fundamental methods to incorporate the feature map information into the assignment process. Finally, while the Dilated Fusion module enhances the richness of the features in the deepest levels, investigating comparable modules for shallower feature maps may yield further improvements in the detection precision.

In summary, OE-Net represents a significant step forward in TOD for satellite imagery and its application to pipeline monitoring. OE-Net provides modular innovations that seamlessly integrate with baseline object detectors, and enhance the detection precision without adding computational overhead. By addressing its limitations and pursuing the proposed directions for future work, this research provides a foundation for continued advancements in the field.

Acknowledgements

I am incredibly grateful to the team at Orbital Eye for making this research project possible and granting me the opportunity to delve into a field that deeply interests me. I am equally grateful to Dr. Jamshidnejad for her exceptional supervision, unwavering support, and the many insightful check-in meetings that challenged my thinking and refined my academic approach to this topic. Her guidance has been instrumental in shaping this research paper.

References

Bai, Y., Zhang, Y., Ding, M., Ghanem, B., 2018. Sod-mtgan: Small object detection via multi-task generative adversarial network, in: Proceedings of the European conference on computer vision (ECCV), pp. 206–221.

Bengio, Y., Simard, P., Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks 5, 157–166.

Cai, Z., Vasconcelos, N., 2018. Cascade r-cnn: Delving into high quality object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6154–6162.

Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H., 2019. Gcnet: Non-local networks meet squeeze-excitation networks and beyond, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pp. 1971–1980.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers, in: European conference on computer vision, Springer. pp. 213–229.

Cheng, G., Han, J., 2016. A survey on object detection in optical remote sensing images. ISPRS journal of photogrammetry and remote sensing 117, 11–28.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine learning 20, 273–297.

Dai, L., Liu, H., Tang, H., Wu, Z., Song, P., 2023. Ao2-detr: Arbitrary-oriented object detection transformer. IEEE Transactions on Circuits and Systems for Video Technology 33, 2342–2356.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee. pp. 248–255.

Devaki, P., Vineetha, P., Reddy, C., Bharathi, P., Karimulla, S., Kumar, S., 2023. Fine-grained feature enhancement for object detection in remote sensing images. International Research Journal of Modernization in Engineering Technology and Science 5, 2112–2118.

Dhillon, A., Verma, G.K., 2020. Convolutional neural network: a review of models, methodologies and applications to object detection. Progress in Artificial Intelligence 9, 85–112.

Ding, J., Xue, N., Long, Y., Xia, G.S., Lu, Q., 2019. Learning roi transformer for oriented object detection in aerial images, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 2849–2858.

Ding, J., Xue, N., Xia, G.S., Bai, X., Yang, W., Yang, M.Y., Belongie, S., Luo, J., Datcu, M., Pelillo, M., et al., 2021. Object detection in aerial images: A large-scale benchmark and challenges. IEEE transactions on pattern analysis and machine intelligence 44, 7778–7796.

Dong, C., Loy, C.C., He, K., Tang, X., 2015. Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence 38, 295–307.

Ghiasi, G., Lin, T.Y., Le, Q.V., 2019. Nas-fpn: Learning scalable feature pyramid architecture for object detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 7036–7045.

Girshick, R., 2015. Fast r-cnn, in: Proceedings of the IEEE international conference on computer vision, pp. 1440–1448.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580–587.

Guo, C., Fan, B., Zhang, Q., Xiang, S., Pan, C., 2020. Augfpn: Improving multi-scale feature learning for object detection, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12592–12601.

Han, J., Ding, J., Li, J., Xia, G.S., 2021a. Align deep features for oriented object detection. IEEE transactions on geoscience and remote sensing 60, 1–11.

Han, J., Ding, J., Xue, N., Xia, G.S., 2021b. Redet: A rotation-equivariant detector for aerial object detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 2786–2795.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence 37, 1904–1916.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

Ji, H., Gao, Z., Mei, T., Ramesh, B., 2019. Vehicle detection in remote sensing images leveraging on simultaneous super-resolution. IEEE Geoscience and Remote Sensing Letters 17, 676–680.

Kang, J., Yang, H., Kim, H., 2024. Simplifying two-stage detectors for on-device inference in remote sensing. arXiv preprint arXiv:2404.07405 .

Kim, J., Lee, J.K., Lee, K.M., 2016. Accurate image super-resolution using very deep convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1646–1654.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. nature 521, 436–444.

Lee, C., Park, S., Song, H., Ryu, J., Kim, S., Kim, H., Pereira, S., Yoo, D., 2022. Interactive multi-class tiny-object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14136–14145.

Lei, S., Shi, Z., Mo, W., 2021. Transformer-based multistage enhancement for remote sensing image super-resolution. IEEE Transactions on Geoscience and Remote Sensing 60, 1–11.

Li, H., Zhang, R., Pan, Y., Ren, J., Shen, F., 2024a. Lr-fpn: Enhancing remote sensing object detection with location refined feature pyramid network. arXiv preprint arXiv:2404.01614 .

Li, J., Liang, X., Wei, Y., Xu, T., Feng, J., Yan, S., 2017. Perceptual generative adversarial networks for small object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1222–1230.

Li, J., Xie, C., Wu, S., Ren, Y., 2024b. Uav-yolov5: A swin-transformer-enabled small object detection model for long-range uav images. Annals of Data Science , 1–30.

Li, Q., Mou, L., Xu, Q., Zhang, Y., Zhu, X.X., 2019a. R3-net: A deep network for multioriented vehicle detection in aerial images and videos. IEEE Transactions on Geoscience and Remote Sensing 57, 5028–5042.

Li, W., Chen, Y., Hu, K., Zhu, J., 2022. Oriented reppoints for aerial object detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 1829–1838.

Li, Y., Chen, Y., Wang, N., Zhang, Z., 2019b. Scale-aware trident networks for object detection, in: Proceedings of the IEEE/CVF international conference on computer vision, pp. 6054–6063.

Li, Y., Hou, Q., Zheng, Z., Cheng, M.M., Yang, J., Li, X., 2023. Large selective kernel network for remote sensing object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 16794–16805.

Li, Y., Li, X., Dai, Y., Hou, Q., Liu, L., Liu, Y., Cheng, M.M., Yang, J., 2024c. Lsknet: A foundation lightweight backbone for remote sensing. International Journal of Computer Vision , 1–22.

Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017a. Feature pyramid networks for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117–2125.

Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017b. Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, pp. 2980–2988.

Liu, S., Qi, L., Qin, H., Shi, J., Jia, J., 2018. Path aggregation network for instance segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8759–8768.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C., 2016. Ssd: Single shot multibox detector, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, Springer. pp. 21–37.

Liu, Y., Sun, P., Wergeles, N., Shang, Y., 2021. A survey and performance evaluation of deep learning methods for small object detection. Expert Systems with Applications 172, 114602.

Loshchilov, I., 2017. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 .

Noh, J., Bae, W., Lee, W., Seo, J., Kim, G., 2019. Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection, in: Proceedings of the IEEE/CVF international conference on computer vision, pp. 9725–9734.

Ouyang, L., Guo, G., Fang, L., Ghamisi, P., Yue, J., 2023. Pcldet: Prototypical contrastive learning for fine-grained object detection in remote sensing images. IEEE Transactions on Geoscience and Remote Sensing 61, 1–11.

Pang, J., Li, C., Shi, J., Xu, Z., Feng, H., 2019. $\mathcal{R}^2$ -cnn: Fast tiny object detection in large-scale remote sensing images. IEEE Transactions on Geoscience and Remote Sensing 57, 5512–5524.

Peyré, G., Cuturi, M., et al., 2019. Computational optimal transport: With applications to data science. Foundations and Trends® in Machine Learning 11, 355–607.

Pu, Y., Wang, Y., Xia, Z., Han, Y., Wang, Y., Gan, W., Wang, Z., Song, S., Huang, G., 2023. Adaptive rotated convolution for rotated object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6589–6600.

Qiao, S., Chen, L.C., Yuille, A., 2021. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10213–10224.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788.

Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence 39, 1137–1149.

Shermeyer, J., Van Etten, A., 2019. The effects of super-resolution on object detection performance in satellite imagery, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 0–0.

Singh, B., Davis, L.S., 2018. An analysis of scale invariance in object detection snip, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3578–3587.

Singh, B., Najibi, M., Davis, L.S., 2018. Sniper: Efficient multi-scale training. Advances in neural information processing systems 31.

Sun, X., Wang, P., Yan, Z., Xu, F., Wang, R., Diao, W., Chen, J., Li, J., Feng, Y., Xu, T., Weinmann, M., Hinz, S., Wang, C., Fu, K., 2022. Fair1m: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery. ISPRS Journal of Photogrammetry and Remote Sensing 184, 116–130.

Tayara, H., Soo, K.G., Chong, K.T., 2017. Vehicle detection and counting in high-resolution aerial images using convolutional regression neural network. Ieee Access 6, 2220–2230.

Tian, Z., Shen, C., Chen, H., He, T., 2019. Fcos: Fully convolutional one-stage object detection, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9626–9635.

Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W., 2013. Selective search for object recognition. International journal of computer vision 104, 154–171.

Wang, D., Zhang, Q., Xu, Y., Zhang, J., Du, B., Tao, D., Zhang, L., 2022. Advancing plain vision transformer toward remote sensing foundation model. IEEE Transactions on Geoscience and Remote Sensing 61, 1–15.

Wang, G., Zhao, H., Lyu, S., Cheng, G., Chang, Q., Feng, W., Zhao, Q., Shi, Z., 2024. Swin-tod: Smooth wasserstein distance and instance-level neighboring enhancement for remote sensing tiny object detection. IEEE Transactions on Geoscience and Remote Sensing 62, 1–15.

Wang, J., Yang, W., Guo, H., Zhang, R., Xia, G.S., 2021. Tiny object detection in aerial images, in: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 3791–3798.

Wen, L., Cheng, Y., Fang, Y., Li, X., 2023. A comprehensive survey of oriented object detection in remote sensing images. Expert Systems with Applications 224, 119960.

Xia, G.S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L., 2018. Dota: A large-scale dataset for object detection in aerial images, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Xiao, J., Guo, H., Zhou, J., Zhao, T., Yu, Q., Chen, Y., Wang, Z., 2023. Tiny object detection with context enhancement and feature purification. Expert Systems with Applications 211, 118665.

Xie, X., Cheng, G., Wang, J., Yao, X., Han, J., 2021. Oriented r-cnn for object detection, in: Proceedings of the IEEE/CVF international conference on computer vision, pp. 3520–3529.

Xu, C., Ding, J., Wang, J., Yang, W., Yu, H., Yu, L., Xia, G.S., 2023. Dynamic coarse-to-fine learning for oriented tiny object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7318–7328.

Xu, C., Wang, J., Yang, W., Yu, H., Yu, L., Xia, G.S., 2022a. Detecting tiny objects in aerial images: A normalized wasserstein distance and a new benchmark. ISPRS Journal of Photogrammetry and Remote Sensing 190, 79–93.

Xu, C., Wang, J., Yang, W., Yu, H., Yu, L., Xia, G.S., 2022b. Rfla: Gaussian receptive field based label assignment for tiny object detection, in: European conference on computer vision, Springer. pp. 526–543.

Xu, C., Wang, J., Yang, W., Yu, L., 2021. Dot distance for tiny object detection in aerial images, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 1192–1201.

Xu, Y., Fu, M., Wang, Q., Wang, Y., Chen, K., Xia, G.S., Bai, X., 2020. Gliding vertex on the horizontal bounding box for multi-oriented object detection. IEEE transactions on pattern analysis and machine intelligence 43, 1452–1459.

Yang, X., Yan, J., Feng, Z., He, T., 2021a. R3det: Refined single-stage detector with feature refinement for rotating object, in: Proceedings of the AAAI conference on artificial intelligence, pp. 3163–3171.

Yang, X., Yan, J., Ming, Q., Wang, W., Zhang, X., Tian, Q., 2021b. Rethinking rotated object detection with gaussian wasserstein distance loss, in: International conference on machine learning, PMLR. pp. 11830–11841.

Yang, X., Yang, J., Yan, J., Zhang, Y., Zhang, T., Guo, Z., Sun, X., Fu, K., 2019. Scrdet: Towards more robust detection for small, cluttered and rotated objects, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 8231–8240.

Yu, F., Koltun, V., 2016. Multi-scale context aggregation by dilated convolutions, in: Bengio, Y., LeCun, Y. (Eds.), 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings.

Yu, H., Tian, Y., Ye, Q., Liu, Y., 2024. Spatial transform decoupling for oriented object detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 6782–6790.

Yu, X., Gong, Y., Jiang, N., Ye, Q., Han, Z., 2020. Scale match for tiny person detection, in: Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp. 1257–1265.

Yu, X., Shi, Z., 2015. Vehicle detection in remote sensing imagery based on salient information and local shape feature. Optik 126, 2485–2490.

Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13, Springer. pp. 818–833.

Zhang, F., Zhou, S., Wang, Y., Wang, X., Hou, Y., 2024. Label assignment matters: A gaussian assignment strategy for tiny object detection. IEEE Transactions on Geoscience and Remote Sensing 62, 1–12.

Zhao, Q., Sheng, T., Wang, Y., Tang, Z., Chen, Y., Cai, L., Ling, H., 2019. M2det: A single-shot object detector based on multi-level feature pyramid network, in: Proceedings of the AAAI conference on artificial intelligence, pp. 9259–9266.

Zhiqiang, W., Jun, L., 2017. A review of object detection based on convolutional neural network, in: 2017 36th Chinese control conference (CCC), IEEE. pp. 11104–11109.

Zhong, Y., Wang, J., Peng, J., Zhang, L., 2020. Anchor box optimization for object detection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1286–1294.

Zhou, Y., Wang, S., Zhao, J., Zhu, H., Yao, R., 2022a. Fine-grained feature enhancement for object detection in remote sensing images. IEEE Geoscience and Remote Sensing Letters 19, 1–5.

Zhou, Y., Yang, X., Zhang, G., Wang, J., Liu, Y., Hou, L., Jiang, X., Liu, X., Yan, J., Lyu, C., Zhang, W., Chen, K., 2022b. Mmrotate: A rotated object detection benchmark using pytorch, in: Proceedings of the 30th ACM International Conference on Multimedia.

Zhou, Z., Zhu, Y., 2024. Libra-sod: Balanced label assignment for small object detection. Knowledge-Based Systems 302, 112353.

Zhu, H., Zhou, Y., Xu, C., Zhang, R., Yang, W., 2024. Enhancing fine-grained object detection in aerial images via orthogonal mapping. arXiv preprint arXiv:2407.17738 .