Universiteit Leiden

Master Computer Science

Oriented Object Detection in Satellite Imagery: A Comparative Study and Hybrid Deep Learning Approach

Name: Rudraksh Kanoongo
Student ID: s4133528

Date: 11-07-2025

Specialisation: Artificial intelligence

1st supervisor: Prof. Dr. D.M. Pelt
2nd supervisor: Prof. Dr. H.R. Doughty

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Object detection in high-resolution imagery is a critical task in many real-world applications, including urban planning, disaster response and infrastructure monitoring. In particular, detection of groundworks, construction-works and roadworks from satellite imagery enables automated surveying, progress tracking and early risk detection in dynamic environments. However, these tasks are challenging due to high object density, varied orientations and large intra-class variation across spatial scales.

The research examines oriented object detection as a solution to these challenges by comparing two key bounding box representations, horizontal bounding boxes (HBB) and oriented bounding boxes (OBB). The research evaluates three detection architectures, Faster R-CNN (F-RCNN), Rotated Faster R-CNN (R-FRCNN), which are CNN-based models, and RoI Trans Swin which is a hybrid CNN-Transformer architecture. The research uses a consistent region proposal network for all models and investigates how feature fusion necks, backbone capacities, optimizer strategies (SGD vs. AdamW), and data availability affect the results.

The research makes significant contributions through its detailed architectural evaluation of HBB and OBB models and its assessment of Transformer-based backbones for global context understanding and its analysis of optimizer effects on CNN vs. Transformer performance and its evaluation of detection accuracy under reduced-data conditions using both FAIR1M benchmark and Worksite dataset.

The experimental findings show that OBB models perform better than HBB models in detecting rotated or densely packed objects. The RoI Trans Swin model which uses a Transformer architecture demonstrates superior performance in complex scenarios when trained with AdamW optimization. The research shows that model performance heavily relies on training data volume because small datasets restrict generalization but large datasets enhance accuracy and recall for all categories.

1 Introduction

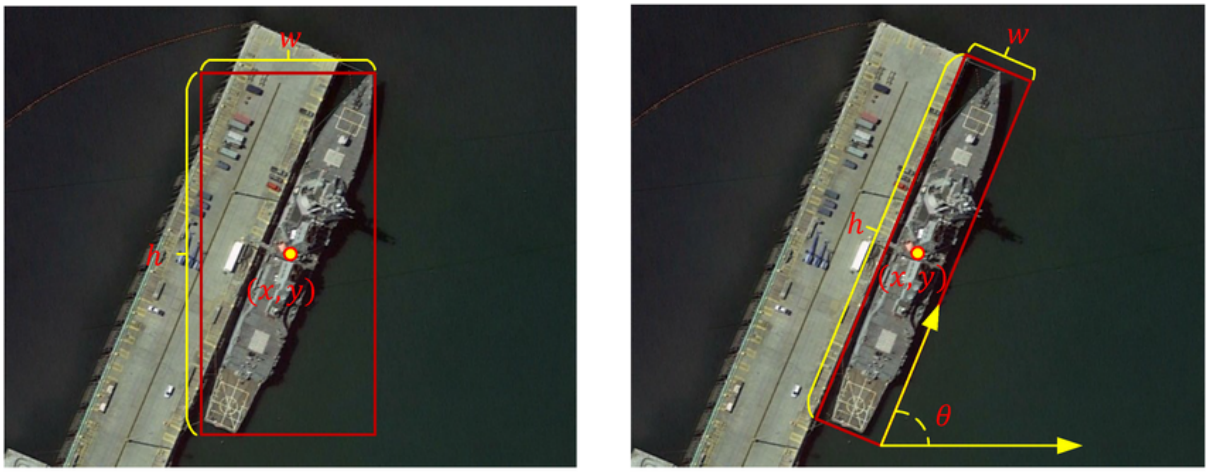
1.1 Motivation

High-resolution satellite and aerial imagery serves essential functions in applications such as urban development, infrastructure monitoring, disaster response, and environmen- tal surveillance [56]. The ability to detect and localize objects from satellite imagery has become essential because of frequent image accumulation and sub-meter ground sample distance (GSD) sensors. This thesis investigates object detection in high-resolution satellite imagery through a comparison of horizontal and oriented bounding box representations in controlled experiments.

The natural image domain has seen significant progress in object detection through deep learning yet conventional detectors struggle to generalize properly to remote sensing applications because of distinctive scene arrangements and diverse object sizes and intricate backgrounds [56, 8]. The traditional detection methods use horizontal bounding boxes (HBBs) as axis-aligned rectangles which depend on the assumption that objects maintain an upright position and follow the image axes. The assumption about object alignment with image axes fails to hold in aerial views because objects appear at random orientations with large size differences and dense arrangements and partial occlusions [53]. The use of horizontal boxes results in detection problems because they fail to match

object edges while including unnecessary background areas and merging with neighboring targets which reduces detection accuracy and produces more false positive results [12].

Remote sensing imagery introduces specific difficulties because of shadows and occlusions and seasonal changes and diverse object appearances within the same class. The "vehicle" class contains objects that span from motorcycles to cargo trucks with diverse orientations and densities and contextual variations [12]. These factors make the object detection pipeline more complex and decrease the effectiveness of standard detectors.



(a) Horizontal bounding box.

(b) Oriented bounding box.

Figure 1: Comparison of horizontal bounding box (HBB) and oriented bounding box (OBB) representations. Image adapted from [45]. While HBB (left) encapsulates the object using axis-aligned rectangles, OBB (right) provides a tighter and rotation-aware fit using five parameters (x, y, w, h, θ).

The detection of oriented objects has become a popular solution to address these problems. The OBB approach enables bounding boxes to rotate freely which matches the actual object geometry. The method decreases background inclusion while enhancing localization precision and reducing detection conflicts in complex scenes. Research conducted on DOTA and FAIR1M datasets demonstrates that OBB methods deliver superior results than HBB methods in different aerial imagery applications [53, 11]. The enhancements deliver the most value for objects that are stretched or tightly grouped or positioned at random angles.

The transition to orientation-aware models receives support from recent advancements in context-aware architectures including Transformer-based models which show better performance in detecting long-range spatial dependencies. These models improve object discrimination in complex scenes through self-attention mechanisms which operate in construction zones and traffic networks and industrial sites [13].

The research aims to improve object detection accuracy in satellite imagery through two main approaches, First, a comparative study between horizontal and oriented detection models using the FAIR1M dataset and Second, the investigation of CNN-Transformer hybrid architectures. The FAIR1M dataset contains high-resolution images with detailed annotations for five categories including Airplane, Ship, Vehicle, Court and Road which represent practical applications and complex visual variations [11].

The research evaluates FAIR1M performance alongside a proprietary industrial dataset

that contains Groundworks and Construction-works and Road-works classes. The dataset remains inaccessible to the public because of confidentiality restrictions but shares similar obstacles which include irregular object shapes and dense spatial layouts and variable orientations that validate the proposed detection methods' practical value and general applicability.

Research Questions:

- How does the choice of bounding box representation (e.g., HBB vs. OBB) impact object detection performance in high-resolution satellite imagery?
- Can hybrid CNN-Transformer architectures improve the detection of arbitrarily oriented and occluded objects?
- How does the choice of optimizer (e.g., SGD vs. AdamW) influence the training dynamics and final detection performance of CNN-based and transformer-based oriented object detection models?
- How do various feature aggregation necks (e.g., FPN-CARAFE vs. PAFPN) affect the effectiveness of oriented object detection models in aerial imagery?
- How does the backbone capacity (e.g., Swin-Tiny vs. Swin-Small) affect the performance and generalization of transformer-based detection models in complex remote sensing environments?
- How robust is the RoI Trans Swin model when trained on a reduced-scale dataset and a limited Worksite dataset that reflects similar constraints, in terms of detection accuracy and class-wise generalization?

The motivation for this work is to bridge the performance gap in satellite object detection by evaluating foundational design choices (HBB vs. OBB) and proposing enhanced architectures that generalize to complex remote sensing environments.

2 Related Works

2.1 Object Detection Paradigms in Remote Sensing Imagery

The core function of computer vision in object detection requires systems to identify objects and their positions in images. In its standard formulation, object detection involves predicting a set of bounding boxes $\{B_i\}_{i=1}^{N}$ and associated class probabilities $\{P_i\}_{i=1}^{N}$, where each box $B_i = (x_i, y_i, w_i, h_i)$ represents the center coordinates and dimensions (width w, height h) of the detected object, and P_i is the softmax output over predefined classes. More advanced formulations include rotated bounding boxes, represented as $B_i = (x_i, y_i, w_i, h_i, \theta_i)$, which introduce an additional orientation angle θ to capture object rotation, common in aerial and satellite imagery [47]. A visual comparison of horizontal and oriented bounding boxes is shown in Figure 1, where the benefits of tighter, rotation-aware OBB representations over standard HBB can be clearly observed.

Object detection functions as a vital component in remote sensing applications which include land-use monitoring and urban planning and disaster management and infrastructure surveillance [7]. Remote sensing imagery differs from natural scene datasets

including COCO [29] and Pascal VOC[14] because objects appear at random orientations while showing large scale differences and being densely packed or partially hidden in scenes. The mentioned conditions result in reduced performance of traditional horizontal bounding box (HBB) detectors including Faster R-CNN [38] and YOLO [37].

The field has developed OBB (oriented object detection) to address these limitations by adapting bounding boxes to match object orientations. The OBB methods have achieved significant improvements in localization accuracy and precision for datasets such as DOTA [47] and FAIR1M [11]. Recent oriented detection architectures, such as Rotated Faster R-CNN, RoI Transformer [12], and R3Det [53] extend the classical detection pipeline with angle-aware region proposal networks, rotated RoI pooling, and rotated non-maximum suppression (NMS).

The detection pipelines now incorporate long-range context and self-attention mechanisms because vision transformers and hybrid models have appeared. The detection pipeline benefits from Swin Transformer [31] and RoI Trans Swin models which unite local convolutional features with transformer-based global reasoning capabilities. The combination of different approaches in this hybrid method enhances both detection precision and reliability when operating in complex visual environments such as construction sites and roads and ports.

The research evaluates horizontal and oriented detection methods on FAIR1M data while assessing how hybrid architectures affect precision and recall performance across different object categories.

Horizontal Bounding Box Detection

The object detection method Horizontal Bounding Box (HBB) uses basic rectangles that are aligned with the image axes. The box is defined by its center point (x, y) and its size, given by the width w and height h. This formulation assumes that object layouts are generally upright and aligned with the image axes, a condition commonly met in natural scene datasets.

The HBB detection pipelines use Faster R-CNN and YOLO models to predict bounding box coordinates and class labels through two separate stages or a single unified pass. These frameworks have proven effective for general object detection tasks and are still widely used due to their computational efficiency and well-established implementation in libraries such as MMDetection [4].

The axis-aligned assumption commonly used in remote sensing applications tends to fail in this context. The objects in aerial imagery such as airplanes and vehicles and linear infrastructure appear at random orientations and different scales. The HBBs in Figure 2 fail to match true object boundaries which results in the capture of substantial background elements and increases the chance of multiple predictions in dense scenes. The misalignment between predicted and ground truth boxes results in lower intersection over union (IoU) scores which negatively affects the effectiveness of NMS post-processing steps. NMS is a filtering technique requires IoU thresholds to eliminate redundant overlapping detections while retaining the most confident predictions.

Oriented Bounding Box Detection

The limitations of axis-aligned bounding boxes (HBBs) can be addressed by using oriented object detection which introduces rotated bounding boxes (OBBs) to better represent the

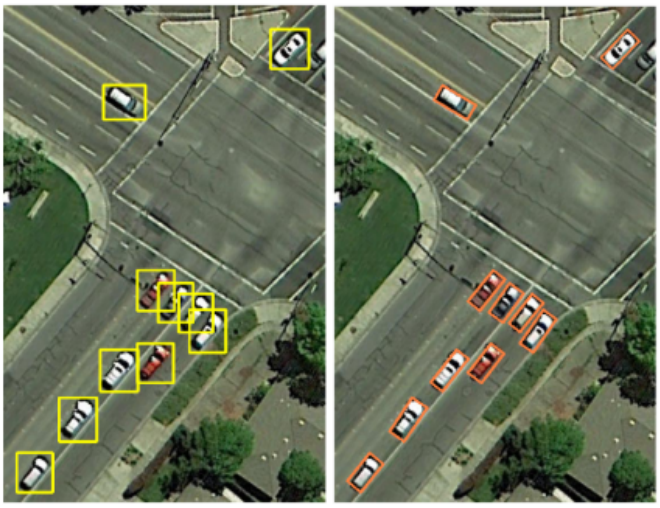


Figure 2: Comparison between horizontal bounding boxes (left, yellow) and oriented bounding boxes (right, orange) for detecting vehicles in remote sensing imagery. In crowded traffic scenes, HBBs often struggle with misalignment and overlapping, while OBBs offer more accurate and tighter localization by matching the true angle and orientation of each object. Image adapted from [46].

spatial geometry of objects in remote sensing imagery.

Although datasets like DOTA and FAIR1M provide oriented bounding box annotations in the form of 8-point polygons (i.e., $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$), most detection models including those implemented in MMRotate internally convert these annotations into a 5-parameter representation $(x_{\text{center}}, y_{\text{center}}, w, h, \theta)$ [47]. This format simplifies the regression task by reducing it to predicting the center position, width, height, and rotation angle of the object.

The use of eight coordinates for direct regression creates geometric constraints which include vertex ordering and convexity that makes optimization challenging and produces unstable training results. The 5-parameter representation allows for more efficient and robust learning through standard loss functions such as Smooth L1 or IoU-based losses. The conversion maintains the fundamental orientation and shape of the object while optimizing both training and inference operations.

The orientation-aware representation produces more precise bounding box alignment with object boundaries particularly for elongated or diagonally oriented objects such as ships, runways, or obliquely parked vehicles. The approach enhances localization accuracy by maximizing object coverage while minimizing background pixel inclusion. The method proves essential for complex environments including ports and construction zones and dense parking lots because adjacent objects frequently overlap. The OBBs system resolves the problem of distinguishing between visually similar objects that differ only in their orientation which is a common issue in horizontal box predictions [54, 53].

The recent developments in oriented object detection have resulted in substantial performance enhancements on existing benchmarks. ReDet achieved a mean Average Precision (mAP) of 76.25% on the DOTA-v1.0 dataset, surpassing the performance of previous methods [17]. Oriented R-CNN achieved an mAP of 75.87% on DOTA-v1.0 and 96.50% on the HRSC2016 dataset, showing its effectiveness in oriented object detection [48]. These models improve traditional detection pipelines by including rotated anchors, rotation-invariant RoI pooling (e.g., Rotated RoIAlign), and angle regression losses to

predict orientation-aware bounding boxes.

The Rotated Faster R-CNN from the MMRotate toolbox functions as the baseline OBB detector throughout this thesis. MMRotate extends the MMDetection framework with essential modules for rotated anchor generation, rotated IoU computation, and rotated non-maximum suppression. Its architectural alignment with Faster R-CNN allows researchers to perform controlled comparisons between different bounding box paradigms.

Research findings demonstrate that OBB models perform better than HBB models in situations where objects have non-uniform orientations and dense or oblique alignments which commonly appear in high-resolution satellite imagery [53, 54, 25].

2.2 CNN-Based Detectors for Satellite Images

Most object detection architectures in remote sensing depend on Convolutional Neural Networks (CNNs) as their fundamental structure. The object detection models Faster R-CNN [38], RetinaNet [28] and Cascade R-CNN [1] use hierarchical feature extraction to detect objects of different scales and classes effectively.

The initial applications of CNNs for satellite imagery involved using Feature Pyramid Networks (FPNs) to extract multi-scale feature representations [27]. The Feature Pyramid Networks (FPNs) enable detectors to detect objects of different sizes by using top-down pathways and lateral connections to propagate low-resolution semantic features across multiple scales. The approach works well for remote sensing scenes because objects in these scenes have different sizes and densities.

The object detection pipelines have incorporated FPNs as well as CNN backbone variants including ResNet [20], HRNet [39] and EfficientNet [41] to enhance feature extraction and generalization performance. These architectures improve both spatial resolution and semantic richness which is important for identifying small and context-dependent objects like vehicles or infrastructure components in high-resolution satellite imagery.

The detection of oriented objects has been enhanced through the extension of conventional CNN-based detectors by using Rotated RoI Align [12] and Rotated Anchor Generators [53] to detect objects of any orientation. These enhancements allow the CNN frameworks to align proposals with object rotation, which improves detection accuracy in scenarios where orientation is a key discriminative feature.

2.3 Transformer-Based and Hybrid Detection Architectures

Vision Transformers (ViTs) have become strong alternatives to traditional convolutional neural networks (CNNs) for visual tasks because they use self-attention to detect global patterns across entire images. The authors Dosovitskiy et al. [13] presented ViTs which divide images into separate non-overlapping patches before applying transformer blocks to process them. The model uses this method to detect image relationships across long distances without performing standard convolutional operations.

The ImageNet [13, 42] image classification tasks showed strong performance for ViTs but their high computational cost and data inefficiency becomes a problem when they are used for dense prediction tasks like object detection and segmentation [31, 2]. The limitations of ViTs led to the development of more efficient hierarchical architectures, most notably the Swin Transformer [31]. The Swin Transformer uses a shifted window attention mechanism that allows linear computational complexity with respect to image

size while preserving locality and context awareness, making it scalable for high-resolution imagery such as remote sensing data.

The architectural comparison between the Vision Transformer and the Swin Transformer reveals key design differences. The original Vision Transformer divides an image into a fixed grid of non-overlapping patches before applying global self-attention across all patches. The design enables the modeling of long-range dependencies but leads to high computational complexity when processing high-resolution inputs which are common in satellite imagery.

The Swin Transformer addresses these problems through two fundamental architectural changes. The model uses hierarchical feature representation through patch merging at different stages to create a feature pyramid structure similar to CNN-based models. The hierarchical structure enables the model to extract features at different scales which proves necessary for detecting objects of different sizes. The shifted window attention mechanism in this model applies self-attention within local windows instead of global windows to achieve substantial computational savings. The windows shift between layers to allow information exchange between windows while maintaining global context aggregation without requiring full global attention.

The new innovations in Swin Transformer architectures improve both efficiency and scalability for dense prediction tasks. The architecture effectively captures both local details and global context while maintaining efficient computation which makes it suitable for object detection and segmentation of high-resolution remote sensing images.

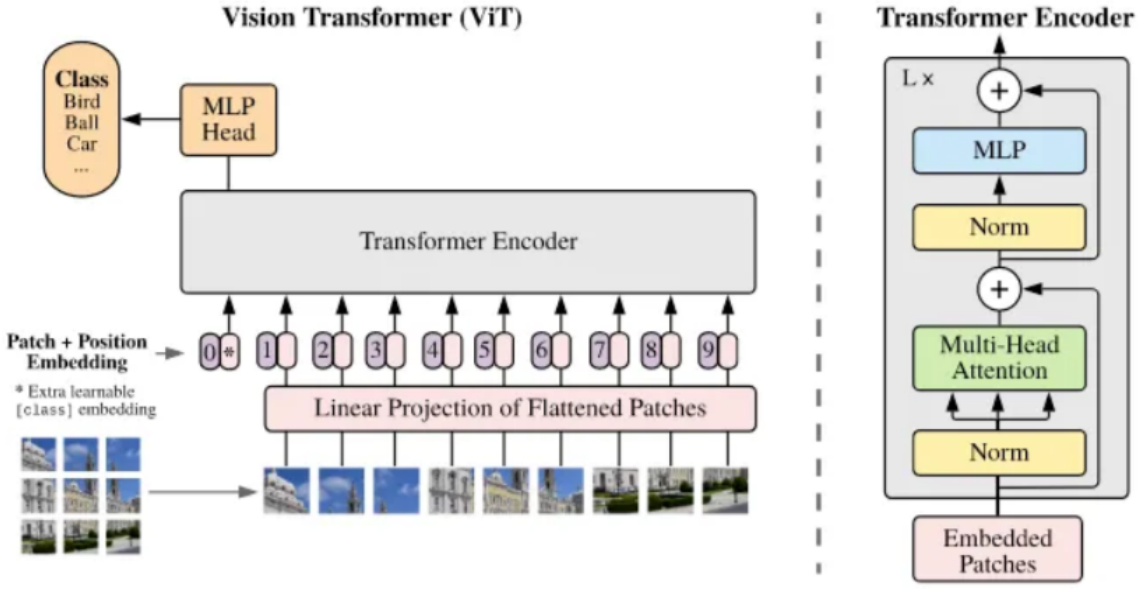


Figure 3: The Vision Transformer (ViT) architecture works by dividing the input image into fixed-size patches, which are then flattened and transformed into embeddings. A special classification token is added to the beginning, and positional information is included to retain spatial context. These embeddings pass through several Transformer Encoder layers that use global self-attention to process the image. The final prediction is made based on the output of the classification token. Adapted from *Liu et al., 2021* [31].

Remote sensing applications benefit from transformer-based detection models includ-

ing Rotated DETR [5], RQ-Transformer [40] and Swin-based pipelines which achieve better results in complex environments. The models achieve better results in cluttered scenes through attention mechanisms that detect long-range contextual relationships and spatial dependencies which enhance the ability to distinguish between overlapping or densely packed objects [40, 54].

The hybrid CNN-transformer architecture represents an effective design which unites CNN local feature extraction capabilities with transformer global reasoning abilities. The combination of local feature extraction and global reasoning capabilities makes these architectures particularly useful for remote sensing applications that require both detailed information and broad contextual understanding.

In this thesis, we employ the RoI Trans Swin model, an advanced hybrid architecture that integrates a Swin Transformer backbone with the RoI Transformer detection head [10]. This combination allows the model to extract semantically rich, context-aware features while maintaining high precision in rotated object localization. As a result, RoI Trans Swin serves as a strong candidate for advancing the performance of oriented object detection in satellite imagery.

2.4 Multi-Scale Feature Fusion in Remote Sensing

Object detection in high-resolution remote sensing imagery requires handling objects of vastly different scales, from small vehicles to large construction zones and airfields. The variation in object size in object detection systems is addressed through multi-scale feature fusion which has become essential for modern object detection architectures. Feature Pyramid Networks (FPN) [27] were one of the first methods to effectively merge low-level spatial features with high-level semantic information by building a top-down feature hierarchy with lateral connections.

This thesis evaluates two advanced FPN-based fusion methods. The first is the Path Aggregation Feature Pyramid Network (PAFPN) [30], which extends the original FPN by introducing a bottom-up pathway that allows stronger bidirectional flow of feature information. This improves the network's ability to detect both small and large objects by reinforcing low-level features with richer semantic context.

The second is FPN-CARAFE [44], which enhances the feature pyramid by replacing traditional upsampling operations with a content-aware reassembly module. CARAFE generates adaptive upsampling kernels that depend on local content, leading to more precise feature reconstruction and better alignment with object boundaries. This proves particularly beneficial in scenes where object boundaries are ambiguous or where small and irregularly shaped instances occur.

The research evaluates these two fusion methods in the RoI Trans Swin detection pipeline to determine their performance in managing spatial complexity and scale diversity in satellite imagery. The experiments reveal how different multi-scale feature designs affect detection results across different object types and datasets.

2.5 Dataset Overview: FAIR1M for Oriented Object Detection

The FAIR1M dataset (Fine-grained Object Recognition in High-Resolution Remote Sensing Imagery) [11] serves as the core dataset for this thesis due to its scale, diversity, and use of oriented bounding box annotations. FAIR1M contains over one million labeled object instances captured from high-resolution satellite images with spatial resolutions

ranging from 0.3 to 0.8 meters per pixel. The dataset is specifically designed to support the development and benchmarking of oriented object detection algorithms.

FAIR1M comprises 37 fine-grained object categories, which are grouped into five high-level superclasses, Airplane, Ship, Vehicle, Court, and Road. These categories represent common objects found in real-world remote sensing scenarios, including ports, highways, airports, and industrial zones.

The Vehicle superclass contains both small and large vehicles which include vans, cargo trucks, trailers and dump trucks that appear at different scales and orientations. The Ship superclass includes fishing boats, warships, tugboats and cargo vessels. The Airplane class contains both commercial jets and military aircraft. The detection task becomes highly challenging because each subclass shows major differences in geometry and texture and orientation.

The extensive annotation quality of FAIR1M makes it suitable for assessing the relative performance of horizontal and oriented detection models when dealing with rotation, occlusion, and intra-class variability.

Figure 4: Data samples of each superclass and their constituent subclasses in the FAIR1M dataset, illustrating the diversity and hierarchical structure of object categories. Figure adapted from Ding et al. [11].

2.6 Comparative Studies: Horizontal vs. Oriented Detection

Research studies have shown that OBB detection methods perform better than HBB methods in all scenarios especially when objects are rotated, densely packed or elongated. Oriented R-CNN achieved an mAP of 75.87% on the DOTA-v1.0 benchmark while ReDet achieved 76.25% on the same dataset, both outperforming conventional HBB detectors [48, 17]. Similar trends are evident on the FAIR1M dataset, where studies have noted substantial gains for categories like ships and vehicles when using orientation-aware frameworks [11].

The research uses the FAIR1M dataset to conduct a specific comparison between HBB detectors and OBB-based detectors for practical performance evaluation.Specifically, Faster R-CNN serves as the baseline horizontal detector, while Rotated Faster R-CNN represents the oriented configuration. The two models employ the same backbone architecture and training process to eliminate any differences that could stem from architectural or training variations.

The experiment delivers quantitative evidence about OBBs in real-world satellite imagery tasks which guides the selection and development of an advanced orientation-aware detection model in this research.

3 Methodology

This section describes the methodology for analyzing how various bounding box representations impact object detection results in high-resolution satellite images. The main comparative study evaluates two state-of-the-art two-stage detectors with matching backbones and training protocols by comparing Faster R-CNN with horizontal bounding boxes (HBB) against Rotated Faster R-CNN which adds oriented bounding boxes (OBB) capabilities through rotation-aware proposals and rotated region alignment and rotation-specific non-maximum suppression.

Both models employ a ResNet-50 backbone with a Feature Pyramid Network (FPN) [27] for multi-scale feature fusion, followed by a Region Proposal Network (RPN) and a shared RoI head for classification and bounding box regression. The Rotated Faster R-CNN leverages rotation-specific modules provided by the MMRotate framework [55], including rotated anchors and rotated RoIAlign, to enable accurate localization of arbitrarily oriented objects.

After establishing the comparative baseline, this study further explores improvements in the oriented detection branch by integrating a Swin Transformer [31] backbone within a RoI Transformer head [12]. The RoI Trans Swin configuration represents an advanced extension which combines hierarchical transformer-based self-attention with orientation-aware region refinement to enhance the accuracy and robustness of OBB detection in complex remote sensing imagery.

3.1 Faster R-CNN

3.1.1 Overview of Faster R-CNN Architecture

Faster R-CNN [38] is a two-stage object detection framework that integrates region proposal generation and object classification into a unified, end-to-end trainable architecture. As illustrated in Figure 5, the model begins with a convolutional neural network backbone typically ResNet-50 [20] which extracts hierarchical feature representations from the

input image. These features are then fed into two main components, the Region Proposal Network (RPN) and the Region of Interest (RoI) head.

The RPN scans the feature maps with a small network to suggest potential object regions. At each location, it uses anchor boxes of various sizes and shapes, which are scored and refined based on their likelihood of containing actual objects. The top-ranked proposals are forwarded to the second stage.

In the second stage, the RoI head refines these proposals. Feature maps corresponding to each proposed region are extracted using RoI pooling (or RoIAlign), producing fixed-size feature representations. These features are passed through fully connected layers to perform two tasks, classification of the object within the proposal, and regression to refine the bounding box coordinates.

The model produces its final output by generating detected objects which are displayed as bounding boxes together with predicted class labels and confidence scores that show the model's detection certainty. The architecture strikes a balance between precision and efficiency which makes it a strong baseline for various object detection applications.

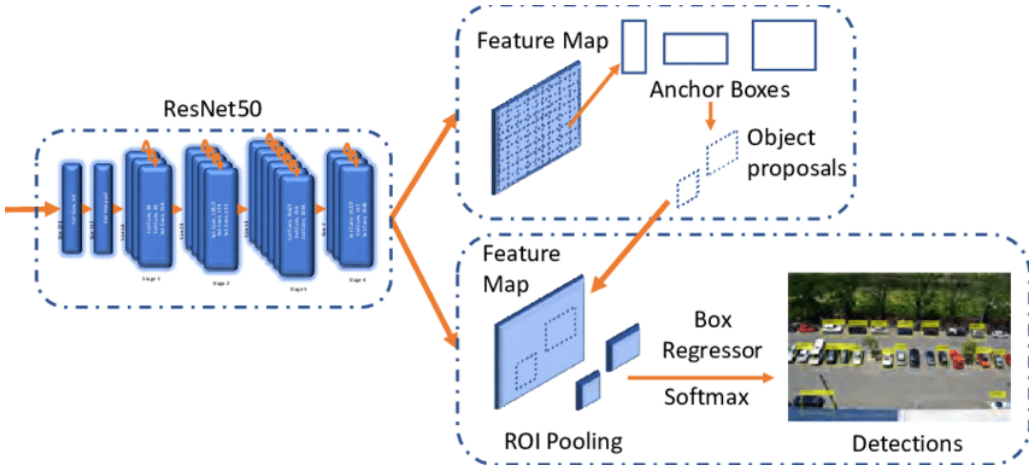


Figure 5: Overview of Faster R-CNN architecture. Feature maps are extracted via ResNet-50, followed by region proposal generation and classification. Figure adapted from [35].

3.1.2 Data Preprocessing

The Faster R-CNN model uses a standard preprocessing step to get input images ready for deep learning. Each image is normalized using preset mean and standard deviation values to standardize pixel intensities, which helps the model learn faster and more effectively during training [24]. The image is also converted from BGR to RGB format to match the input requirements of the backbone network. To ensure it works properly with the downsampling steps in the Feature Pyramid Network (FPN), the image is padded so that both its height and width are divisible by 32 [27].

3.1.3 Backbone

The backbone network used in Faster R-CNN is a ResNet-50 model [20] that has been pre-trained on the ImageNet dataset. It consists of four stages of convolutional layers that gradually extract increasingly detailed features from the input image. Outputs from all four stages are used to capture hierarchical information. To preserve low-level features

and reduce computational overhead, the first stage is frozen during training. Batch normalization layers are included to stabilize training dynamics [21]. This backbone plays a critical role in extracting spatial and semantic features that support region proposal and classification tasks.

3.1.4 Feature Pyramid Network (FPN) Neck

To improve detection across objects of different sizes, the Faster R-CNN architecture incorporates a Feature Pyramid Network (FPN) [27] as the neck component. The FPN is positioned directly after the ResNet-50 backbone, which provides four output feature maps from its convolutional stages, C_2, C_3, C_4, C_5, corresponding to channels of 256, 512, 1024, and 2048, respectively.

The Feature Pyramid Network (FPN) enables object detection across different scales because it generates feature maps at various scales. The network achieves this by combining spatial details from the initial layers with semantic information from the deeper layers. The technique proves particularly beneficial for satellite imagery because it handles objects of varying sizes and clarity that result from altitude position and imaging sensor type and object orientation.

The FPN achieves this through a top-down pathway with lateral connections. Starting from the deepest layer (C_5), high-level features are upsampled and merged with features from earlier layers (C_4 to C_2) that have been passed through 1×1 convolutions to match dimensions. This fusion creates feature maps P_2 through P_5, each with a standardized channel size of 256. An additional pyramid level, P_6, is generated by applying a 3×3 stride-2 convolution on P_5, extending the receptive field for large-scale object detection.

Formally:

$$P_l = \text{Conv}_{1\times1}(C_l) + \text{Upsample}(P_{l+1}),$$

with each P_l subsequently passed through a 3×3 convolution to reduce aliasing effects and enhance localization precision.

The FPN enables the Region Proposal Network (RPN) and RoI heads in Faster R-CNN to function effectively across various object dimensions. The detection performance improves through this method which enhances both recall and precision for objects of all sizes. The multi-scale fusion method is crucial for remote sensing imagery because it addresses both high intra-class variability and object density to achieve balanced detection performance [27].

Overall, the FPN improves the ResNet-50 backbone by adding scale-aware features, allowing Faster R-CNN to better handle the wide range of object sizes commonly found in satellite images.

3.1.5 Region Proposal Network (RPN)

The Region Proposal Network (RPN) is an essential part of the two-stage Faster R-CNN framework [38]. Its main role is to suggest regions in the image, called region proposals, that are likely to contain objects. It operates on the multi-scale feature maps produced by the Feature Pyramid Network (FPN) [27] and generates axis-aligned (horizontal) proposals that are refined in the second detection stage.

The RPN in this configuration is implemented using the standard RPNHead module. It takes input features of 256 channels (from each pyramid level) and passes them through

a shared 3×3 convolution followed by two sibling branches, a classification branch for foreground–background scoring, and a regression branch for bounding box localization.

Anchor boxes are densely generated over the feature maps at five scales, corresponding to strides of [4, 8, 16, 32, 64], using an AnchorGenerator with a base scale of 8 and aspect ratios of [0.5, 1.0, 2.0]. These anchors are matched to ground truth boxes using the MaxIoUAssigner, which assigns positive or negative labels based on IoU overlap.

Bounding box regression is parameterized by the four coordinates (x, y, w, h) and optimized using the Smooth L1 loss function, defined as:

$$L_{\text{bbox}} = \sum_i \text{smooth}_{L_1}(t_i - t_i^*),$$

where t_i and t_i^* denote the predicted and target parameters respectively [15]. The classification loss is computed using a sigmoid Cross-Entropy loss function over the objectness score. The regression targets are encoded using the DeltaXYWHBBoxCoder, which normalizes the bounding box transformations using zero-mean and unit-variance scaling.

The RPN configuration is designed to achieve computational efficiency and general-purpose object detection. The simplicity and effectiveness of this approach make it a strong baseline for high-resolution satellite imagery where most objects retain upright orientations or only moderate deviations.

3.1.6 RoI Head

The Region of Interest (RoI) head performs the last operations of object detection in the Faster R-CNN model by identifying objects and adjusting their bounding boxes [38]. The RoIAlign operation [19] enables the RoI head to extract fixed-size feature maps from the backbone network which maintains precise spatial information. RoIAlign maintains spatial accuracy through its method of avoiding rounding errors during the extraction process.

These feature maps are then passed through the Shared2FCBBoxHead, which consists of two fully connected layers. One branch outputs the class probabilities, while the other predicts adjustments to improve the accuracy of the bounding boxes.

Bounding box regression is performed using the DeltaXYWHBBoxCoder, which encodes the offsets in box center (x, y), width (w), and height (h) relative to the anchor. A class-agnostic regressor is used, meaning the same regression parameters are shared across all object classes. The classification loss is computed using Cross-Entropy Loss, while bounding box regression uses Smooth L1 Loss with $\beta = 1.0$ for stability.

3.1.7 Loss Functions

Faster R-CNN implements a multi-task loss function which unites two elements, classification loss and bounding box regression loss. The model applies this formulation to both the Region Proposal Network (RPN) and the Region of Interest (RoI) head to optimize classification accuracy and localization precision simultaneously.

The classification component uses the standard cross-entropy loss to evaluate the predicted class probabilities against the ground truth labels. For a binary classification setting (e.g., object vs. background in the RPN), the loss is given by:

$$\mathcal{L}_{\text{cls}}(p, p^*) = -p^* \log(p) - (1 - p^*) \log(1 - p) \tag{1}$$

where p is the predicted probability of the proposal being an object, and $p^* \in \{0, 1\}$ is the ground truth label. For multi-class classification at the RoI head, the loss generalizes to:

$$\mathcal{L}_{\text{cls}} = -\sum_{c=1}^{C} y_c \log(\hat{p}_c) \tag{2}$$

where C is the number of object classes, y_c is the one-hot encoded ground truth label, and $\hat{p}_c$ is the predicted softmax probability for class c.

To refine the coordinates of the predicted bounding boxes, Faster R-CNN uses the Smooth L1 loss, which is less sensitive to outliers than the standard L2 loss. It is defined as:

$$\mathcal{L}_{\text{reg}}(t, t^*) = \sum_{i \in \{x,y,w,h\}} \text{smooth}_{L_1}(t_i - t_i^*) \tag{3}$$

where $t = (t_x, t_y, t_w, t_h)$ represents the predicted bounding box offsets and $t^* = (t_x^*, t_y^*, t_w^*, t_h^*)$ are the ground truth offsets. The Smooth L1 function is defined as:

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{4}$$

This loss formulation ensures that small errors are penalized quadratically (L2-like) for stability, while large errors are penalized linearly (L1-like) to reduce the impact of outliers.

The overall loss function for Faster R-CNN is a weighted sum of the classification and regression losses:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda \cdot \mathcal{L}_{\text{reg}} \tag{5}$$

where λ is a scalar weight used to balance the contribution of the regression loss relative to the classification loss. In practice, λ is typically set to 1.0.

This multi-task objective allows the network to learn both what an object is (classification) and where it is located (regression) in a coordinated and efficient manner [38].

3.1.8 Training and Testing Configuration

During training, ground truth bounding boxes are matched to anchors and region proposals using a MaxIoUAssigner, which assigns positive and negative labels based on predefined Intersection over Union (IoU) thresholds. A RandomSampler is then used to pick a balanced mix of positive and negative samples for both the Region Proposal Network (RPN) and the Region of Interest (RoI) head. This helps the model learn more effectively and fairly. During each training step, up to 2000 region proposals are processed per image.

During inference, the model follows the same two-stage detection process. The RPN first generates a set of region proposals, which are then sent to the RoI head for classification and refinement of the bounding boxes. NMS is applied at both the proposal and detection stages to eliminate duplicate or overlapping boxes. The final results are filtered based on a confidence threshold, and only the top-scoring detections are kept. The output includes axis-aligned bounding boxes, each with a predicted class label and a confidence score [38].

3.2 Rotated Faster R-CNN

3.2.1 Overview of Rotated Faster R-CNN Architecture

Rotated Faster R-CNN is built on top of the standard Faster R-CNN framework, keeping the main components like the ResNet-50 backbone and the Feature Pyramid Network (FPN) the same. This ensures consistent feature extraction and multi-scale representation.

The key difference lies in its orientation-aware improvements. It introduces a RotatedRPNHead that generates proposals better suited for detecting objects at various angles. The RoI head is also upgraded to a RotatedStandardRoIHead, which uses RotatedRoIAlign to accurately extract features aligned with the object's orientation. Additionally, the bounding box regression includes an angle parameter (θ), allowing the model to predict five values per object which are center coordinates, width, height, and angle, instead of the usual four [12, 53].

The modifications enable the detector to generate tightly aligned bounding boxes that are rotation-aware which enhances detection performance in scenes with densely packed objects at different orientations. Aside from these rotation-specific changes, the loss functions, training process, and inference steps remain the same as in Faster R-CNN. The consistent approach enables stable training and facilitates comparison between horizontal and oriented object detection results.

3.2.2 Data Preprocessing

Rotated Faster R-CNN follows a preprocessing pipeline very similar to that of Faster R-CNN, using the same steps for image normalization, color conversion, and padding. However, due to the need to detect arbitrarily oriented objects, additional rotation-aware augmentations are introduced. These include operations such as RResize and RRandomFlip, which allow the model to generalize to various object orientations by simulating flips along horizontal, vertical, and diagonal axes during training [9].

3.2.3 Backbone

The backbone network in Rotated Faster R-CNN is also ResNet-50, pre-trained on ImageNet and structured identically to that used in Faster R-CNN [20]. It includes four convolutional stages, from which feature maps are extracted. The first stage remains frozen to preserve low-level features, and batch normalization is used to ensure consistent learning behavior.

3.2.4 Neck

The neck in Rotated Faster R-CNN uses a Feature Pyramid Network (FPN), just like in Faster R-CNN [27]. It combines feature maps from different layers of the ResNet backbone to create multi-scale feature representations, helping the model detect objects of various sizes more effectively.

3.2.5 Rotated Region Proposal Network (R-RPN)

The Region Proposal Network (RPN) in Rotated Faster R-CNN generally follows the same structure as in standard Faster R-CNN, including anchor generation, classification

and regression branches, and the use of Smooth L1 and Cross-Entropy loss functions. However, for oriented object detection, the RPN is modified to handle rotated bounding boxes using the RotatedRPNHead module from the MMRotate framework [33, 34].

Unlike the standard RPN, which generates axis-aligned proposals defined by (x, y, w, h), the rotated version (R-RPN) produces proposals in the form (x, y, w, h, θ) where θ represents the rotation angle of the object relative to the horizontal axis. The orientation-aware proposal generation method is essential for remote sensing imagery because ships aircraft and buildings in the images often appear at different angles. [51, 11].

The anchor generation strategy remains consistent with standard RPNs, employing anchors of multiple aspect ratios across multi-scale FPN feature maps. However, the regression branch in the R-RPN is adapted to predict five parameters, including the angle. During training, proposal matching is performed using a modified MaxIoUAssigner which computes the Intersection-over-Union (IoU) for rotated bounding boxes rather than axis-aligned ones [54]. Proposal filtering and suppression at inference time also leverage rotated IoU metrics to preserve angular alignment.

The RPN in Rotated Faster R-CNN retains the architectural backbone of the standard implementation and the rotation-specific adaptations make it significantly more effective for detecting arbitrarily oriented objects in dense high-resolution aerial scenes.

3.2.6 Bounding Box Coding

Bounding box coding in Rotated Faster R-CNN uses a hybrid approach. The RPN continues to use the DeltaXYWHBBoxCoder, consistent with the original Faster R-CNN. However, the RoI head adopts the DeltaXYWHAHBBoxCoder, which introduces a fifth parameter to represent the rotation angle (θ). This coder also includes mechanisms to stabilize and disambiguate angle predictions [53, 9].

3.2.7 RoI Head

The RoI head in Rotated Faster R-CNN replaces the standard RoI head with a RotatedStandardRoIHead, which incorporates RotatedRoIAlign to extract features from regions that are not necessarily aligned with the image axes [12, 53]. The prediction head, RotatedShared2FCBBoxHead, outputs five values for each detected object which are the center coordinates (x, y), the width and height, and the angle (θ) that represents the object's orientation. The classification branch remains similar to that of Faster R-CNN.

3.2.8 Loss Functions

Rotated Faster R-CNN uses the same loss functions as Faster R-CNN, including cross-entropy loss for classification and Smooth L1 loss for bounding box regression. However, the regression loss is extended to include five parameters instead of four, accounting for the object's orientation [53, 9].

3.2.9 Training and Testing Configuration

The training procedure for Rotated Faster R-CNN closely follows the structure used in Faster R-CNN. Ground truth bounding boxes are matched to anchors and region proposals using a MaxIoUAssigner, which determines positive and negative samples based on Intersection over Union (IoU) thresholds. A RandomSampler is then used to select

a balanced set of positive and negative examples for training both the Region Proposal Network (RPN) and the Region of Interest (RoI) head. Although the sampling strategy is the same as in the standard model, the key difference is that the IoU calculations and regression targets are based on rotated bounding boxes [53].

At inference time, Rotated Faster R-CNN maintains the same two-stage detection pipeline, wherein the RPN generates candidate proposals and the RoI head performs classification and oriented bounding box regression. The predicted bounding boxes include orientation and are expressed using the le90 angle convention, where angles are constrained to the range $(-90°, 0°)$ and then rotated Non-Maximum Suppression is applied [53, 9]. The output consists of rotated bounding boxes along with their associated class labels and confidence scores, making the model well-suited for object detection in aerial and satellite imagery.

3.3 RoI Trans Swin

3.3.1 Overview of RoI Trans Swin Architecture

The RoI Trans Swin model enhances the standard two-stage detection pipeline by introducing a Swin Transformer as the backbone within the existing RoI Transformer framework [12]. While components like the Feature Pyramid Network and the orientation-aware stages for region proposal and alignment remain similar to those in Rotated Faster R-CNN, the key advancement lies in replacing the conventional ResNet backbone with the more powerful and flexible Swin Transformer Tiny variant [31].

The Swin Transformer uses a shifted window-based self-attention mechanism that allows it to efficiently capture global context in an image, while still preserving a hierarchical structure of features across different layers. This design enhances the model's ability to capture long-range dependencies and subtle structural cues in high-resolution remote sensing scenes.

RoI Trans Swin achieves better localization accuracy for densely packed or arbitrarily oriented objects through its combination of transformer-based global reasoning with rotation-aware detection components beyond CNN-based backbones. The FPN neck together with rotated RoIAlign and rotated non-maximum suppression components maintain the same design principles which were explained for Rotated Faster R-CNN.

3.3.2 Swin Transformer Backbone

The RoI Trans Swin model uses the Swin Transformer as its backbone to extract detailed features from high-resolution satellite images through its hierarchical efficient architecture. Liu et al. [31] introduced the Swin Transformer to address the problems of Vision Transformers (ViTs) [13].

Swin Transformer introduces two core innovations, first, a hierarchical representation similar to convolutional neural networks (CNNs), and second, a shifted window-based self-attention mechanism. The model starts by partitioning the input image $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$ into non-overlapping patches of size 4×4. Each patch is flattened and linearly projected into a 96-dimensional embedding vector, resulting in a token sequence $\{\mathbf{z}_i \in \mathbb{R}^{96}\}_{i=1}^{N}$ where $N = \frac{H \cdot W}{P^2}$ and P is the patch size.

These patch embeddings are processed through four successive stages, where each stage consists of several Swin Transformer blocks. The depth of each stage in the Swin-Tiny configuration is defined as [2, 2, 6, 2], with corresponding channel dimensions [96,

192, 384, 768]. Between stages, a patch merging operation reduces the spatial resolution by half and doubles the number of channels, constructing a feature hierarchy similar to that found in CNN-based backbones like ResNet.

Each Swin Transformer block within a stage performs self-attention within fixed-size local windows (e.g., 7×7), which significantly reduces computational complexity. The self-attention operation within a window is defined as:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + B\right)V,$$

where Q, K, and V are the query, key, and value matrices computed from the input embeddings, d_k is the dimensionality of the key vectors, and B denotes a relative positional bias matrix that encodes spatial relationships within the window [31].

Swin enables cross-window interactions and global context modeling through its shifted window mechanism. The model connects distant spatial regions through multiple layers by shifting windows by a fixed number of pixels (usually half the window size) in alternating layers. The method preserves local attention's linear computational complexity while approximating global receptive fields across depth which makes it suitable for detecting large-scale contextual information in complex scenes like satellite images.

In the RoI Trans Swin model, the outputs from each of the four stages are extracted and passed to the Feature Pyramid Network (FPN) neck. The integration process enables downstream multi-scale feature fusion which effectively detects objects of different sizes and orientations. The Swin Transformer backbone establishes a strong base for precise object detection that handles orientation in remote sensing applications.

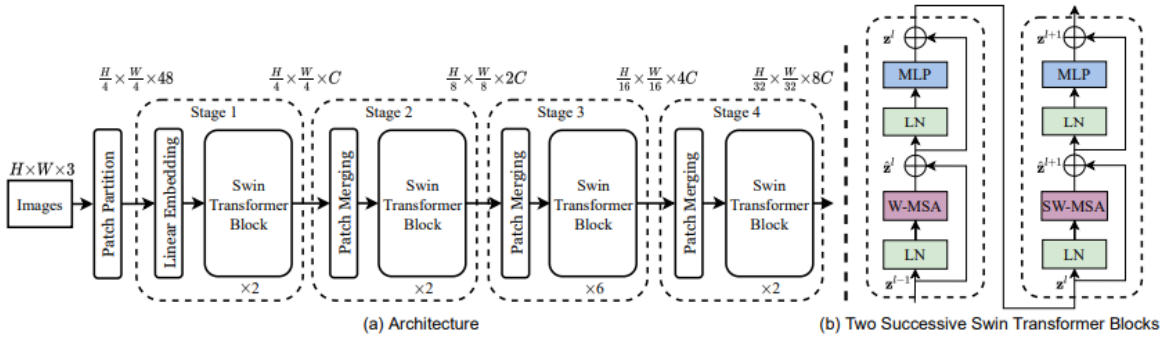


Figure 6: (a) Overview of the Swin Transformer architecture, which hierarchically extracts features through four stages using patch merging to reduce spatial resolution and increase channel capacity. (b) Two consecutive Swin Transformer blocks are shown: each includes Layer Normalization (LN), either window-based self-attention (W-MSA) or its shifted variant (SW-MSA), followed by a multi-layer perceptron (MLP) and residual connections. This alternating self-attention mechanism captures both local and cross-window dependencies. Adapted from Liu et al., 2021 [31].

3.3.3 Feature Pyramid Network (FPN) Neck

The RoI Trans Swin model also uses a Feature Pyramid Network (FPN) [27] as its neck, with a design and function similar to the FPN used in Faster R-CNN. As mentioned earlier, the FPN helps combine features from different scales using a top-down pathway and lateral connections, improving the model's ability to detect objects of various sizes.

However, a key difference in the RoI Trans Swin configuration lies in the nature of the backbone. Instead of using a convolutional ResNet as in Faster R-CNN, this model uses a hierarchical Swin Transformer backbone, which outputs feature maps from four stages with channels of 96, 192, 384, and 768 respectively. These transformer-derived features are spatially rich and semantically diverse due to the window-based self-attention mechanism unique to Swin architectures [31].

To adapt these transformer outputs for pyramid construction, each stage's feature map is passed through a 1×1 convolution to normalize channel dimensions before being aggregated through top-down upsampling and addition, similar to the conventional FPN pipeline. This produces pyramid levels P_2 to P_5, and an additional P_6 level is created by applying a 3×3 convolution with stride 2 to P_5, mirroring the strategy used in the standard setup.

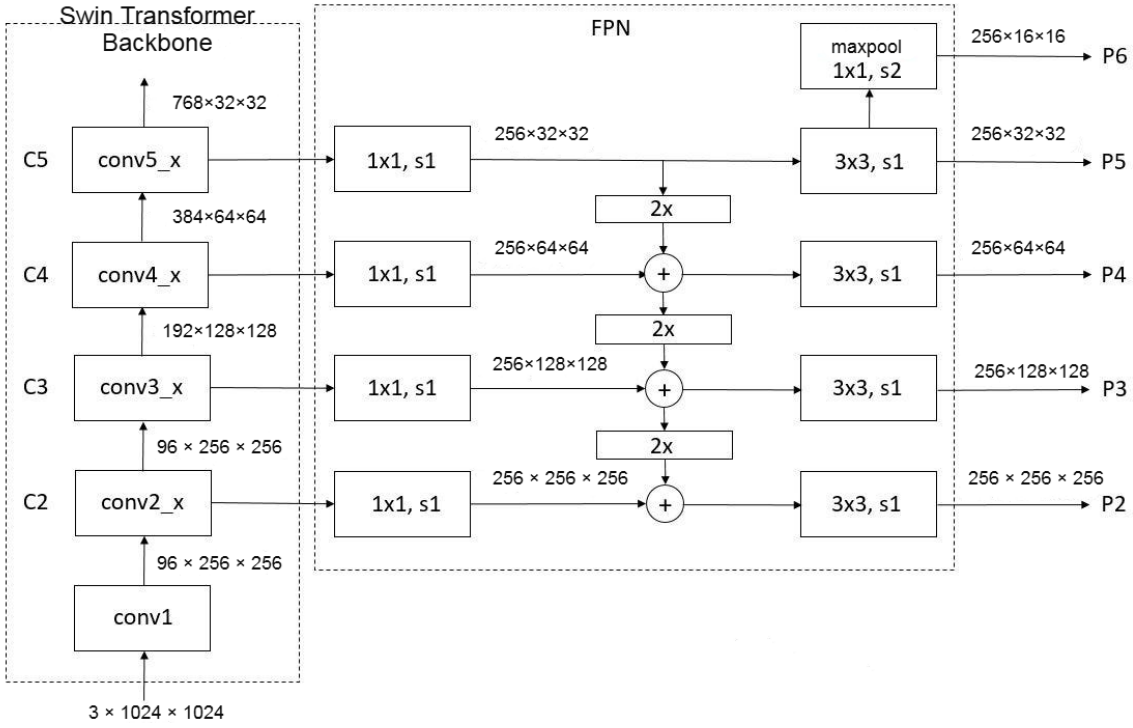


Figure 7: A schematic of the Feature Pyramid Network (FPN) integrated with a Swin Transformer backbone. Feature maps from four hierarchical stages (C2–C5) of the Swin Transformer are passed through 1×1 convolutions and combined in a top-down pathway using upsampling and element-wise addition to produce pyramid features P2–P5, followed by max pooling to create P6. For the convolutional layers, 1×1 and 3×3 denote the kernel sizes, while s=1 and s=2 indicate stride values of 1 (no downsampling) and 2 (spatial downsampling by half), respectively. CHW format is used (Channels × Height × Width). Image adapted and modified from [26].

The FPN architecture stays the same but the Swin backbone produces features that contain more global context than CNNs do. The pyramid benefits from this improvement because it can detect long-range dependencies which proves useful for complex object arrangements in cluttered and high-resolution satellite scenes.

By reusing the same FPN structure but applying it to transformer-based features, the RoI Trans Swin model leverages both the global reasoning of attention mechanisms and the scale-sensitive aggregation of pyramidal design. This combination of global attention-

based features from the Swin Transformer and multi-scale fusion via FPN has been shown to improve detection performance in challenging remote sensing scenarios, particularly for arbitrarily oriented or densely packed objects [31, 12, 53].

3.3.4 Rotated Region Proposal Network (R-RPN)

The RoI Trans Swin Tiny model utilizes the same Rotated Region Proposal Network (R-RPN) structure and configuration as described for Rotated Faster R-CNN. This includes the use of the RotatedRPNHead, anchor generation at multiple aspect ratios and feature map scales, prediction of rotated proposals in (x, y, w, h, θ) format, and the use of rotated IoU metrics for both training and inference. These consistent design choices across both models ensure fair architectural comparison and align with MMRotate's rotation-aware detection framework [33, 34, 54].

3.3.5 Two-Stage RoI Head with Spatial Transformer

The RoI Trans Swin model uses a two-stage Region of Interest (RoI) head which is designed to improve the detection of arbitrarily oriented objects in remote sensing imagery. The first stage of this model is similar to the standard RoI head which is used in Faster R-CNN for extracting fixed-size features using RoIAlign from axis-aligned proposals, but it also extends it by introducing a second stage for orientation-aware refinement.

In the second stage, the RoI head integrates a RoI Transformer module [12], which predicts a rotated bounding box for each axis-aligned RoI by estimating a 5-tuple transformation $(\Delta x, \Delta y, \Delta w, \Delta h, \Delta\theta)$. These parameters are used to generate Rotated RoIs (RRoIs) that more precisely align with the geometry of the underlying objects.

To extract features from these RRoIs, the model uses Rotated RoIAlign, which samples along the orientation of the object, ensuring that spatial features are captured in alignment with the true object orientation. This helps overcome the misalignment issues seen in standard detection pipelines when dealing with rotated or skewed objects.

The entire transformation is implemented through a learnable Spatial Transformer Network (STN) [22], allowing the model to dynamically adapt its feature sampling to the shape and orientation of each object.

The two-stage head achieves better localization and classification results for aerial imagery datasets like FAIR1M by combining RoI-level spatial transformation and orientation-specific feature extraction [50].

3.3.6 Bounding Box Coding and Loss Functions

In contrast to the single-stage regression used in Faster R-CNN, the RoI Trans Swin model adopts a two-stage detection head, with each stage having its own classification and regression branches. The primary distinctions are in the bounding box coders employed and the progressive refinement approach designed for oriented object detection, even though the general multi-task loss configuration stays the same, combining cross-entropy loss for classification with Smooth L1 loss for bounding box regression.

The first stage in the RoI head uses the DeltaXYWHAHBBoxCoder, the same coder employed in the R-FRCNN architecture. The second stage of the model uses DeltaXYWHAOBBoxCoder to perform detailed refinement of oriented bounding boxes (OBBs). The coder includes features that address rotation-specific ambiguities while enhancing alignment with objects that have arbitrary orientations. The model processes features

obtained from Rotated RoIAlign after spatial transformation while applying tight normalization parameters for accurate adjustments.

The classification and regression losses in both stages follow the same formulation as described in the Faster R-CNN methodology. However, the RoI Trans Swin model uses different bounding box coders, tailored for oriented objects, and stage-specific β values in the Smooth L1 loss to control sensitivity, a larger $\beta = 1.0$ for the coarse first stage, and a smaller value (e.g., $\beta = 1.0/9$) for the fine-grained refinement in the second stage.

The RoI Trans Swin model enhances both localization and orientation accuracy through its method of transforming axis-aligned proposals into precise oriented bounding boxes while adjusting loss sensitivity levels. The proposed refinement method works best for remote sensing images because objects in these images tend to be rotated and elongated and densely packed [12, 53].

3.3.7 Training Configuration and Optimization

The RoI Trans Swin model is trained using the AdamW optimizer, a variant of the Adam optimizer that decouples weight decay from the gradient-based parameter updates [32]. This optimizer has become a preferred choice for training transformer-based models due to its improved regularization behavior and stability in large-scale vision tasks [18, 42]. The optimizer is configured with a base learning rate of 0.0001 which aligns with commonly recommended learning rates for transformer backbones and moment decay rates of $\beta_1 = 0.9$ and $\beta_2 = 0.999$, these values are widely adopted in transformer literature for stable convergence [43]. A high weight decay of 0.05 is used to improve generalization by penalizing overly complex models as shown effective in Swin-based and ViT-based models [31, 13].

To further improve generalization and stability during training, a parameter-wise optimization strategy is adopted. Specifically, parameters associated with positional embeddings and normalization layers are assigned zero weight decay, a strategy inspired by recent works [42, 31] which show that decoupling regularization from sensitive components (such as layer norms and embeddings) preserves representational integrity and improves convergence. This ensures that essential spatial encoding and normalization behaviors are not disrupted during training.

The training pipeline includes multiple data augmentation steps which enhance model robustness while improving its ability to generalize to new scenarios. Each training image is resized to a fixed resolution of 1024×1024 pixels using bilinear interpolation. This size offers a trade-off between preserving object level details in high-resolution satellite images and maintaining computational feasibility [11]. The RRandomFlip module in MMRotate applies random flipping operations across horizontal, vertical and diagonal directions with equal probability (0.25 each). Such multi-directional flipping is important in remote sensing tasks where objects can appear in arbitrary orientations, unlike natural image datasets which typically exhibit upright objects [51].

After geometric augmentations, images are normalized using the ImageNet mean and standard deviation values ($[123.675, 116.28, 103.53]$ mean and $[58.395, 57.12, 57.375]$ std), ensuring compatibility with the pretrained Swin Transformer backbone [31]. Padding is applied to maintain alignment with the model's spatial requirements specifically, a size divisor of 32 is used so that input dimensions remain consistent with the architecture's stride and pooling operations. This is a standard configuration in both MMRotate and MMDetection frameworks [9]

Training is performed for 12 epochs while the learning rate schedule starts with a linear warm-up phase of 500 iterations followed by step-wise decay at epochs 8 and 11. This schedule has been shown to stabilize early training and prevent gradient instability, especially in transformer-based models [18, 42]. The batch size is limited to 1 per GPU because of memory restrictions caused by working with high-resolution inputs and transformer computations. The model achieves stable and effective convergence through strong regularization and robust augmentation pipeline even with the small batch size.

The training configuration follows best practices in deep learning for vision transformers and oriented object detection. The chosen hyperparameters are adapted from default configurations and are motivated by recent research showing their effectiveness in maintaining a balance between model complexity, training stability and generalization capacity.

3.3.8 Inference and Post-processing

The RoI Trans Swin model executes forward passes on test images to generate object class predictions and oriented bounding box (OBB) predictions during the inference phase. The inference process maintains the architectural pipeline from training but excludes stochastic components including data augmentation and dropout. The model generates multiple region proposals which contain both classification scores and le90 format oriented bounding boxes.

The model employs Rotated NMS (R-NMS) as a specialized version of Non-Maximum Suppression (NMS) to eliminate duplicate or overlapping detection results.

The R-NMS module receives candidate boxes and their confidence scores as input before it performs suppression based on a predefined IoU threshold. The algorithm selects the highest-scoring box first and then removes all other boxes that exceed the threshold of overlap. The suppression process is crucial for minimizing false positives in areas with high object density such as vehicle parking lots and shipping ports and construction zones.

After R-NMS is applied, the final set of predictions is output in the form:

$$\{(x_i, y_i, w_i, h_i, \theta_i, s_i, c_i)\}_{i=1}^{N},$$

where x_i, y_i are the center coordinates, w_i, h_i are the width and height, θ_i is the rotation angle, s_i is the classification confidence score, and c_i is the predicted class label. These predictions can be visualized by drawing rotated rectangles on the original image, and are used to evaluate detection accuracy using metrics such as mAP@50.

The RoI Trans Swin pipeline for inference and post-processing is designed to achieve high precision in remote sensing applications. The model achieves robust real-world aerial imagery performance through its rotation-aware bounding boxes and adapted suppression mechanisms which handle complex spatial arrangements and orientation variance.

4 Experimental Setup

The experimental framework described in this section evaluates horizontal and oriented object detection in high-resolution satellite imagery. The experiments use identical architecture and training schedules and evaluation metrics to create a balanced comparison between horizontal bounding box (HBB) and oriented bounding box (OBB) representations.

To this end, Faster R-CNN from the MMDetection framework is used as the baseline model for HBB detection, while Rotated Faster R-CNN from the MMRotate framework is employed for OBB detection. All models are trained using consistent configurations, including backbone architecture, optimizer settings, learning schedules and data augmentation pipelines to ensure experimental parity.

The full FAIR1M dataset was used for training but a smaller version was also created to mimic real-world industrial scenarios where there may be limited annotated training data available. The smaller dataset was created by downsampling the original training set while keeping the class distribution the same to evaluate the performance of the RoI Trans Swin model in data-constrained environments and its robustness under limited supervision. Additional experiments are conducted using a separate industrial worksite dataset to evaluate the model's applicability in real-world conditions.

4.1 Dataset Preparation

For all experiments conducted in this study, the FAIR1M dataset [11] was used as the benchmark for evaluating object detection performance. To facilitate training, validation, and testing of both horizontal and oriented object detection models, the dataset's training portion which consists of 16,488 images was randomly partitioned into three subsets, 13,190 images (80%) for training, 1,649 images (10%) for validation, and 1,649 images (10%) for testing. This split ensures the model is tested on data it hasn't seen before, while also keeping a balanced mix of object classes in each subset for fair and consistent evaluation.

Table 1: FAIR1M Dataset Split for Training, Validation, and Testing

Subset	**Number of Images**	**Percentage**
Training	13,190	80%
Validation	1,649	10%
Testing	1,649	10%
Total	16,488	100%

The dataset alignment with practical infrastructure monitoring applications required a class aggregation strategy to simplify annotation processes. Fine-grained subclasses with similar appearance or function were combined into broader superclasses. For example, various airplane models such as Boeing747, A350, and ARJ21 were merged into a unified Airplane category. Similarly, ship types including Dry-Cargo-Ship, Fishing-Boat, and Tugboat were grouped under the Ship class, while land vehicles like Cargo-Truck, Small-Car, and Trailer were combined into a Vehicle category, as shown in Table 2.

Table 2: Superclass and corresponding merged subclasses used in the detection task

Superclass	**Subclasses**
Airplane	A220, A321, A330, A350, ARJ21, C919, Boeing737, Boeing747, Boeing777, Boeing787, other-airplane
Ship	Dry-Cargo-Ship, Engineering-Ship, Fishing-Boat, Liquid-Cargo-Ship, Motorboat, Passenger-Ship, Warship, other-ship, Tugboat
Vehicle	Bus, Cargo-Truck, Dump-Truck, Excavator, Small-Car, Tractor, Trailer, Truck-Tractor, Van, other-vehicle

The Basketball-Court, Tennis-Court, Intersection, and Bridge categories maintained their separate classes because they have unique geometric characteristics which are essential for remote sensing applications as shown in Table 3.

Table 3: Classes retained as separate categories due to their distinct geometry and relevance in remote sensing contexts.

Category	**Classes**
Court	Basketball-Court, Tennis-Court, Football-Field, Baseball-Field
Road Infrastructure	Intersection, Roundabout, Bridge

The result of this aggregation is a detection task modeled as a 10-class classification problem, with subclass mappings summarized in Table 4.

Table 4: Number of instances per class for the training, validation, and test subsets.

Class	**Train**	**Val**	**Test**
Airplane	24,363	2,990	2,971
Baseball-Field	846	85	131
Basketball-Court	1,011	128	132
Bridge	816	104	88
Football-Field	693	74	86
Intersection	5,132	577	659
Roundabout	467	35	61
Ship	24,984	3,325	3,153
Tennis-Court	2,413	275	236
Vehicle	251,990	31,135	34,506
Total	**312,715**	**38,728**	**41,023**

To support training for both horizontal and oriented object detection, the original annotations provided as 8-point polygons were converted into two separate formats. For the oriented detection model used in MMRotate [55], the annotations were transformed into the DOTA-style format [47]. This format defines each bounding box using the coordinates of its four corners in clockwise order, along with the object class and a difficulty flag. It preserves the object's exact shape and orientation, making it ideal for rotation-aware detection tasks.

For horizontal detection using the MMDetection framework [3], the same polygon annotations were converted into axis-aligned rectangles by calculating the smallest enclosing horizontal bounding box. These converted annotations were then saved in the COCO-style JSON format, which is widely used in object detection frameworks. Although this conversion results in a loss of rotation information, it enables a fair comparison between horizontal and oriented models using identical underlying data.

The dual-format annotation strategy maintains consistent object instances and class labels throughout detection pipelines which allows performance differences to be traced back to detection architecture rather than input data or annotation format inconsistencies.

4.2 Model Setup and Configuration Consistency

To ensure a fair and reliable comparison between horizontal and oriented object detection, this study aligns the architectural and training configurations of both models wherever possible. Specifically, the Faster R-CNN model from the MMDetection framework [3] was modified to match the training setup and architectural parameters used in the Rotated Faster R-CNN model provided by MMRotate [55]. In doing so, all aspects unrelated to bounding box representation such as feature extraction, proposal generation, and training procedures remain equivalent across both models. This setup guarantees that the observed performance differences can be attributed directly to the representation of bounding boxes (HBB vs. OBB), and not to confounding factors like architecture or hyperparameters.

The training process is synchronized across both implementations. Optimization is performed using stochastic gradient descent (SGD) with a learning rate of 0.0025, momentum of 0.9, and a weight decay of 10^{-4}. These values follow standard practices in object detection literature (e.g.,[38, 20]) where SGD with momentum is favored for its stability and generalization ability in CNN-based models. The learning rate strategy includes a linear warm-up phase over 500 iterations to prevent early training instability, this technique has shown to improve convergence when using deep architectures[16]. After warm-up, the learning rate follows a step decay policy, reducing at epoch 8 and 11 within a total 12 epoch schedule. This configuration is often seen in MMDetection default setups and provides a balance between training time and convergence without overfitting [31].

Each GPU processes 16 images per batch which is a widely adopted batch size that offers a good trade-off between training stability and computational efficiency. Batch sizes that are too small can lead to noisy gradient estimates while excessively large batches may lead to worse generalization performance [23].

The detection and training pipelines in both models also use identical preprocessing routines. Images are resized to 1024×1024 pixels, this resolution preserves detail while being computationally feasible [52]. Inputs are then normalized using ImageNet mean and standard deviation values to ensure compatibility with pretrained backbones. A random horizontal flip with probability 0.5 introduces basic data augmentation to improve generalization. Finally, padding is applied to ensure input dimensions are divisible by 32 which aligns with the stride of the deep convolutional layers and avoids spatial mismatches during downsampling operations.

The loss functions are consistent across both setups, cross-entropy is used for classification and Smooth L1 loss for bounding box regression. Both are standard in object

detection pipelines and provide stable training signals [15].

The detailed side-by-side comparison of model components is presented in Table 5, which shows the architectural symmetry and training parity across the HBB and OBB configurations.

Table 5: Comparison of configurations between Faster R-CNN (HBB) and Rotated Faster R-CNN (OBB)

Component	**Faster R-CNN (HBB)**	**Rotated Faster R-CNN (OBB)**
Framework	MMDetection	MMRotate
Backbone	ResNet-50 (pretrained on ImageNet)	ResNet-50 (pretrained on ImageNet)
Neck	FPN (5 levels)	FPN (5 levels)
Head	StandardRPNHead + Shared2FCBBoxHead	RotatedRPNHead + RotatedShared2FCBBoxHead
Anchor Generator	Scales [8], Ratios [0.5, 1.0, 2.0]	Scales [8], Ratios [0.5, 1.0, 2.0]
RoI Pooling	RoIAlign, output size 7	Rotated RoIAlign, output size 7
Loss Functions	CrossEntropy (cls), SmoothL1 (reg)	CrossEntropy (cls), SmoothL1 (reg)
BBox Type	Horizontal Bounding Box (HBB)	Oriented Bounding Box (OBB, le90 format)
NMS Type	IoU threshold = 0.5	Rotated IoU threshold = 0.5
Augmentation	Resize to 1024×1024, random flip (0.5), normalization (ImageNet mean/std), padding (32)	Same as HBB
Scheduler	12 epochs, StepLR (milestones at 8, 11), linear warmup (500 iters)	Same as HBB
Optimizer	SGD (lr=0.0025, momentum=0.9, weight decay=1e-4)	Same as HBB
Batch Size	16 per GPU	16 per GPU
Evaluation Metric	mAP at IoU = 0.5	mAP at IoU = 0.5

The experimental design standardizes all model components and procedures except for the bounding box formulation to determine how horizontal versus oriented bounding boxes affect detection performance.

4.3 SGD vs. AdamW Optimizer Comparison for Oriented Object Detection

The research investigates how optimization strategies affect oriented object detection results through a study that evaluates Stochastic Gradient Descent (SGD) and AdamW optimizers. The evaluation examines how different optimizer settings affect detection accuracy between Rotated Faster R-CNN (a CNN-based model) and RoI Trans Swin Tiny (a transformer-based model). The key goal of this experiment is to identify which

optimizer is better suited for which model, providing practical guidance for future training setups.

For a fair evaluation, both models were trained using the same settings for each optimizer. In the case of the SGD configuration, the learning rate was set to 0.0025, with a momentum of 0.9 and a weight decay of 0.0001. These values follow widely accepted best practices for training convolutional neural networks in object detection tasks [20, 38, 15]. Momentum helps smooth gradient updates and accelerates convergence, while weight decay provides regularization to prevent overfitting. Gradient clipping was applied using a maximum norm of 35 to stabilize training by preventing gradient explosion, this technique has shown to improve training stability in deeper networks [36]. The learning rate schedule used a linear warmup over the first 500 iterations to prevent unstable early updates [16] followed by a step decay policy with reductions at the 8th and 11th epochs. The total number of epochs was fixed at 12 which balances training duration and convergence efficiency and is consistent with typical settings used in MMDetection and MMRotate benchmarks [55, 3].

For the AdamW configuration, the learning rate was set to 0.0001 with (β_1, β_2) values of (0.9, 0.999) and a weight decay of 0.05. These settings are commonly used in training Transformer-based vision models like Swin Transformer [31, 13]. AdamW decouples weight decay from the gradient update which helps in offering more stable and generalizable training in models with LayerNorm and attention layers [32]. In addition, a parameter-wise configuration was used to set the weight decay multiplier to zero for position embeddings and normalization layers. This is a recommended practice in vision Transformers as these components are highly sensitive to regularization and may degrade model performance if over-penalized [6, 49].

By default, the Rotated Faster R-CNN model in MMRotate is trained using SGD, and this experiment evaluates how the model performs when AdamW is used instead. Conversely, the RoI Trans Swin Tiny model is typically optimized using AdamW, and here it is also tested under the SGD setting to observe any performance degradation or improvement.

This experimental setup enables a direct comparison of optimizer performance across CNN-based and transformer-based object detectors. Ultimately, it allows for determining which optimizer configuration is better suited to each model architecture when applied to oriented object detection in high-resolution satellite imagery.

4.4 Reduced FAIR1M Subset for Limited-Data Evaluation

The evaluation of the RoI Trans Swin model under limited training conditions was performed by creating a reduced version of the FAIR1M dataset to simulate real-world industrial scenarios where annotated data is often scarce. This setup helps assess the model's robustness and generalization ability when object classes have fewer instances closer to the constraints observed in operational satellite imagery applications.

The reduced dataset was obtained by downsampling the original FAIR1M dataset while maintaining class diversity. Table 6 shows the number of instances per class across the training, validation, and test sets.

Table 6: Instance distribution per class in the reduced FAIR1M dataset.

Class	Train	Validation	Test
Airplane	1923	126	261
Baseball-Field	342	38	44
Basketball-Court	377	57	49
Bridge	331	45	49
Football-Field	315	36	51
Intersection	759	75	134
Roundabout	328	37	35
Ship	2804	174	395
Tennis-Court	551	84	56
Vehicle	25627	2897	3518

This reduced configuration allows us to evaluate how well the RoI Trans Swin model can learn under data-constrained settings and whether its architectural advantages (e.g., Swin backbone and orientation-aware RoI head) continue to provide reliable performance with fewer samples per class.

4.5 Worksite Dataset

To evaluate the generalization capability of the object detection models in real-world industrial applications, an additional dataset referred to as the worksite dataset was used. This dataset comprises 469 high-resolution satellite images captured by Airbus' Pleiades Neo and Planet's SkySat satellites, with spatial resolutions of approximately 0.3 meters and 0.5 meters, respectively. The imagery was originally collected for pipeline corridor monitoring, with the primary objective of performing change detection. As such, the dataset is particularly tailored to detect objects related to infrastructure activities such as groundworks, construction-works, and road-works in the vicinity of pipeline routes.

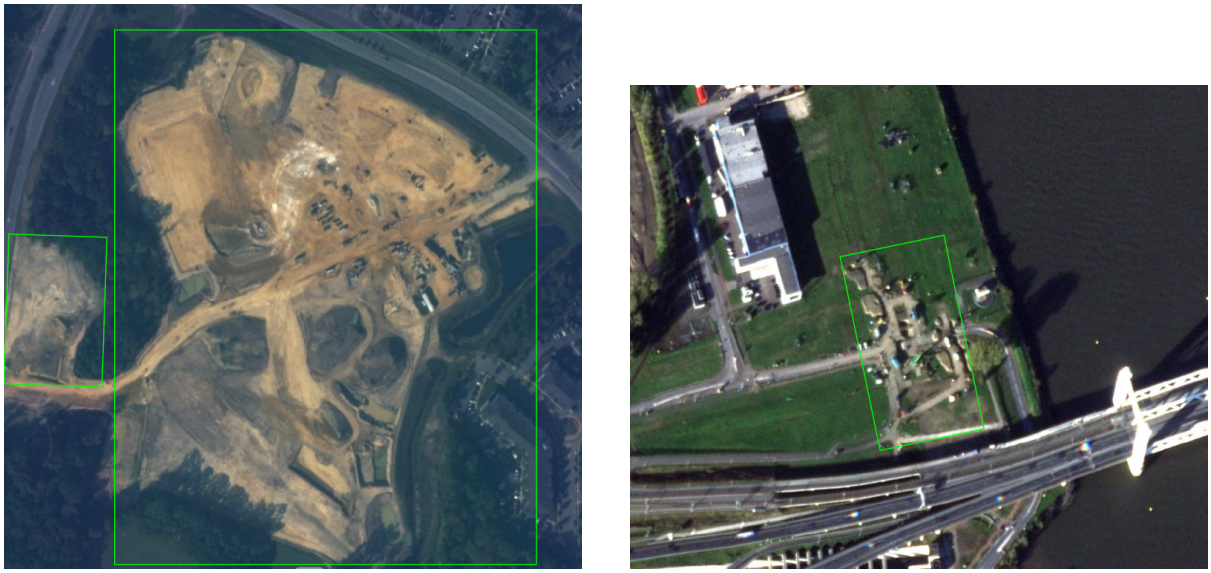
The 469 images were split into training, validation, and test subsets using an 8:1:1 ratio, resulting in 375 images for training, 46 for validation, and 48 for testing. Table 7 provides the number of annotated object instances for each class across these three subsets.

Table 7: Instance distribution across the worksite dataset splits.

Class	Train	Validation	Test
Groundworks	525	59	74
Construction-works	97	8	11
Road-works	75	13	2
Total Instances	**697**	**80**	**87**

To assess detection performance on this domain-specific dataset, the RoI Trans Swin Tiny model was applied to predict the three target classes. This evaluation allows for analysis of the model's behavior under real-world industrial data conditions and limited training samples.

To provide visual context, representative examples of annotated instances from the worksite dataset are shown below. The images demonstrate the diverse nature of industrial monitoring objects which include different appearances, sizes, and orientations.

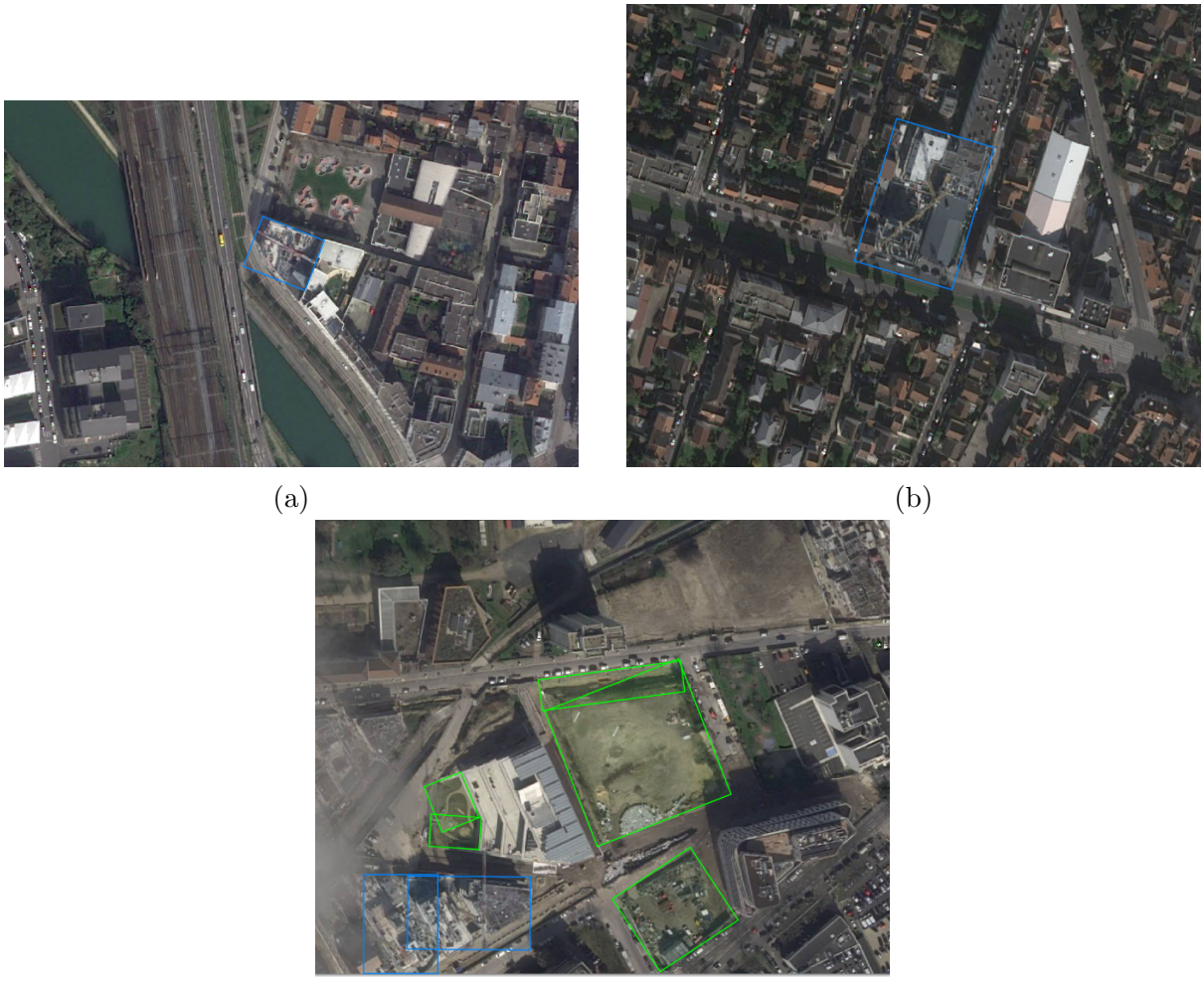


(a) (b)

Figure 8: The green bounding boxes represent groundworks instances, which can vary significantly in shape and size. Image acquired from © 2024 Planet Labs PBC.

(a) (b)

Figure 9: The red bounding boxes represent road-works instances. These objects typically follow linear paths and appear as narrow, elongated structures, though their shape and orientation can vary across locations. Image acquired from © 2024 Planet Labs PBC and © Airbus DS (2024).



(a) (b)

(c)

Figure 10: The blue bounding boxes indicate construction-works objects. Image (c) displays a scene containing both construction-works and groundworks classes. Image acquired from © 2024 Planet Labs PBC and © Airbus DS (2024).

4.6 Evaluation and Fairness Considerations

To ensure a fair and unbiased comparison between horizontal and oriented detection approaches, both models, Faster R-CNN for horizontal bounding boxes (HBB) and Rotated Faster R-CNN for oriented bounding boxes (OBB), were trained and tested using the same dataset splits from the FAIR1M dataset.

The evaluation of detection performance relied on standard metrics which included Intersection over Union (IoU), Precision, Recall, Average Precision (AP), and mean Average Precision (mAP). The metrics used in this study are widely used in object detection benchmarks including remote sensing applications [14, 11, 55].

Intersection over Union (IoU) measures the overlap between a predicted bounding box B_p and its corresponding ground truth box B_{gt} and is defined as:

$$\text{IoU} = \frac{B_p \cap B_{gt}}{B_p \cup B_{gt}}$$

The IoU metric serves dual purposes in both training and evaluation stages. The MaxIoUAssigner module uses IoU to determine positive and negative samples during training.

The IoU threshold determines which overlapping detections with lower confidence scores should be eliminated through NMS during inference.

Precision measures the number of accurate positive predictions relative to all positive predictions made:

$$\text{Precision} = \frac{TP}{TP + FP}$$

where TP refers to true positives and FP to false positives.

The Recall metric calculates the ratio of correctly predicted positive results to the total number of actual positive instances:

$$\text{Recall} = \frac{TP}{TP + FN}$$

where FN denotes false negatives. These two metrics are especially important in satellite imagery detection, where dense and overlapping instances are common.

The Average Precision (AP) measures the total area under the precision-recall curve. It summarizes the trade-off between precision and recall across different detection thresholds. The study calculates AP@50 by using an IoU threshold of 0.5. The final metric combines Average Precision (AP) scores from all object classes into a single value called mean Average Precision (mAP):

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

where N is the total number of classes and AP_i is the average precision for class i. MMDetection computes COCO-style mAP by averaging over multiple IoU thresholds (from 0.50 to 0.95 in steps of 0.05), while MMRotate uses Pascal VOC-style mAP, typically evaluated at a single IoU threshold of 0.5 (mAP@50), which is more common in remote sensing benchmarks.

The evaluation protocol ensures that the results are directly comparable and that any observed performance differences arise solely from the bounding box representation, horizontal vs. oriented, rather than inconsistencies in model architecture or experimental setup. Such methodological parity is crucial for drawing valid and interpretable conclusions in remote sensing object detection research.

5 Results and Discussion

5.1 Quantitative Comparison of OBB and HBB Detection Models

A quantitative comparison was conducted between the horizontal bounding box (HBB) model using Faster R-CNN from MMDetection and the oriented bounding box (OBB) model using Rotated Faster R-CNN from MMRotate. Both models were trained under identical conditions on the FAIR1M dataset to ensure a controlled and fair evaluation.

Performance was assessed using the mean Average Precision at an IoU threshold of 0.5 (mAP@50). Table 8 presents the class-wise and overall mAP@50 results. A more detailed interpretation of these outcomes is provided in the following section.

Table 8: Class-wise mAP@50 comparison between MMDetection (HBB) and MMRotate (OBB)

Class	mAP@50 (HBB)	mAP@50 (OBB)	Best Model
Airplane	0.909	0.908	HBB
Baseball-Field	0.911	0.896	HBB
Basketball-Court	0.701	0.777	OBB
Bridge	0.408	0.312	HBB
Football-Field	0.762	0.768	OBB
Intersection	0.684	0.665	HBB
Roundabout	0.871	0.870	HBB
Ship	0.534	0.659	OBB
Tennis-Court	0.879	0.908	OBB
Vehicle	0.489	0.793	OBB
Overall mAP@50	0.715	**0.756**	OBB

5.1.1 Class-Level Analysis and Interpretation

The overall results indicate that both models achieved competitive detection performance, with the OBB model attaining a higher overall mAP@50 score (0.756) compared to the HBB model (0.715). The class-wise analysis reveals detailed strengths and limitations of each approach regarding object shape, orientation and contextual complexity.

1. Vehicle: The most significant performance gain is observed in the Vehicle category, where the OBB model (Rotated Faster R-CNN) achieves a considerably higher mAP@50 of 0.793, compared to only 0.489 for the HBB model (Faster R-CNN). This performance gap can be directly attributed to the nature of vehicle appearances in high-resolution satellite imagery, vehicles are often oriented at arbitrary angles, tightly packed in parking lots, or distributed irregularly along roads and construction zones.

The below comparison in Figure 11 illustrates this difference. Subfigure (a) shows predictions made by the HBB model using axis-aligned bounding boxes, while subfigure (b) displays predictions from the OBB model using oriented boxes. The HBB prediction fails to detect many vehicles that are angled or parked closely together (e.g., in the lower-middle parking lot) because horizontal boxes overlap. The axis-aligned representation fails to distinguish between vehicles that are diagonally aligned because it requires extensive background inclusion which results in suppressed detections after NMS.

The OBB model in subfigure (b) provides superior individual vehicle instance detection because it uses rotation-aware bounding boxes that tightly fit each object regardless of its orientation or proximity to other objects. The model achieves better localization and reduced overlaps and improved precision especially in high-density scenarios.

The observations confirm previous research findings [12, 53] which demonstrate that orientation-aware proposals enhance IoU consistency and decrease false positives in complex urban environments.

(a) Vehicle predictions using HBB (Faster R-CNN).

(b) Vehicle predictions using OBB (Rotated Faster R-CNN).

Figure 11: Comparison of HBB vs. OBB predictions for the Vehicle class. The OBB model better detects angled and closely parked vehicles, while HBB suffers from missed or overlapping detections.

2. Airplane and Baseball-Field: Both the Airplane and Baseball-Field classes exhibit only marginal performance differences between the HBB and OBB models, with the HBB model slightly outperforming OBB in terms of mAP@50 (Airplane: 0.929 vs. 0.908, Baseball-Field: 0.911 vs. 0.892). These categories typically contain objects that are large, well-structured, and isolated in the imagery, which reduces the need for orientation-aware detection.

In the case of airplanes (see Figure 12), most instances are uniformly distributed around the terminal and are sufficiently spaced apart. Their long bodies and wings are symmetrically aligned along the runways and surrounding airport areas, causing minimal overlap or occlusion. As a result, the horizontal bounding boxes (HBB) already encapsulate the targets with minimal background, and the rotated bounding boxes (OBB) provide little additional localization benefit. This explains why both detectors perform almost equivalently on this class, with little room for OBB to improve IoU or detection precision.

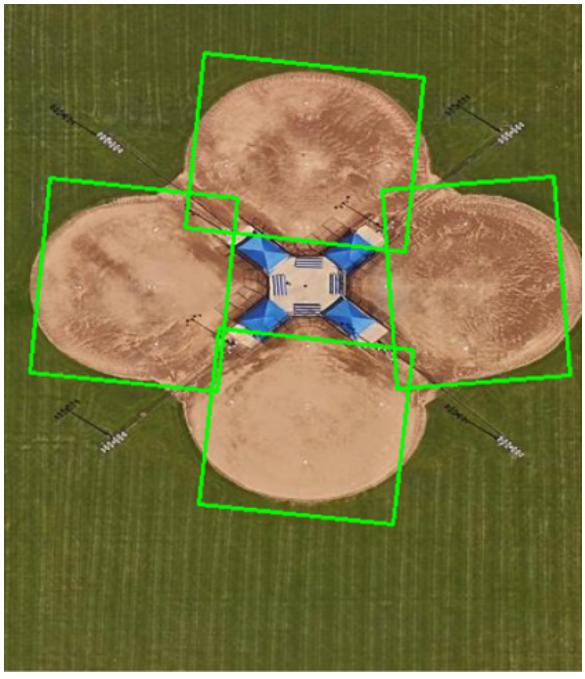
The Baseball-Field class in Figure 13 displays a four-quadrant field configuration which presents symmetrical and axis-aligned characteristics. The fields maintain distinct spatial positions while showing regular visual patterns through their defined shapes and uniform measurements. The circular or elliptical shape of these fields makes horizontal bounding boxes capable of effective enclosure. The angled contours of rotated bounding boxes lead to overfitting or misalignment when they try to tightly enclose rounded field elements. The application of OBB does not lead to improved performance and may create minor disturbances in the results.

(a) Airplane predictions using HBB (Faster R-CNN).

(b) Airplane predictions using OBB (Rotated Faster R-CNN).

Figure 12: Comparison of airplane detection between HBB and OBB models. Both models accurately detect individual aircraft due to their large size, consistent shape, and separation from nearby objects.

(a) Baseball-Field predictions using HBB (Faster R-CNN).



(b) Baseball-Field predictions using OBB (Rotated Faster R-CNN).

Figure 13: Comparison of baseball field detection using HBB and OBB. The fields are well-structured and symmetric, reducing the need for rotation-aware boxes.

3. Bridge: While the Bridge class intuitively seems suited for oriented detection, the actual performance does not reflect that advantage yielding lower accuracy for the

OBB model (0.312) compared to the HBB model (0.408). This counterintuitive result is evident in the visual predictions shown in Figure 22.

In Figure 22(a), the Faster R-CNN model with horizontal bounding boxes (HBB) captures the bridge entirely but includes a large portion of background, especially the adjacent waterbody and surrounding trees. Due to this, the predicted box overlaps significantly with the ground truth area, contributing to a higher IoU and being counted as a true positive.

The Rotated Faster R-CNN prediction (OBB) in Figure 22(b) produces a tighter fit that follows the bridge orientation. The IoU drops below 0.5 because of small misalignment and an overly narrow aspect ratio. The red bounding box indicates that this prediction does not qualify as a true positive. The OBB model shows high sensitivity to small rotation and box scaling mistakes when detecting elongated structures such as bridges.

These results highlight a limitation of the OBB approach, although it attempts to provide tighter and more geometrically aligned boxes, small localization errors in narrow and elongated objects can significantly reduce IoU, leading to missed detections.

(a) Bridge predictions using HBB (Faster R-CNN). The horizontal box includes a lot of background but is still counted as correct.

(b) Bridge predictions using OBB (Rotated Faster R-CNN). The rotated box is well-oriented but fails IoU threshold, hence marked red.

Figure 14: Comparison of bridge predictions between HBB and OBB models. While HBB includes more background, it often exceeds the IoU threshold. OBB predictions, though aligned, may not count as true positives due to strict IoU criteria.

4. Basketball-Court, Football-Field, Ship, and Tennis-Court: This group of categories provides a strong visual and statistical illustration of how the orientation-aware capability of the OBB model improves detection precision, especially when objects are not perfectly aligned with the image axes.

Basketball-Court & Tennis-Court: The two classes contain structured courts which are densely packed in different directions which creates difficulties for horizontal bounding box (HBB) detectors. The Faster R-CNN model using HBB encounters difficulties with these cases because its axis-aligned boxes exceed actual object boundaries when courts are placed diagonally which results in substantial overlap between neighboring instances. The excessive overlap between detected objects causes Non-Maximum Suppression (NMS) to incorrectly eliminate valid detection results which leads to reduced

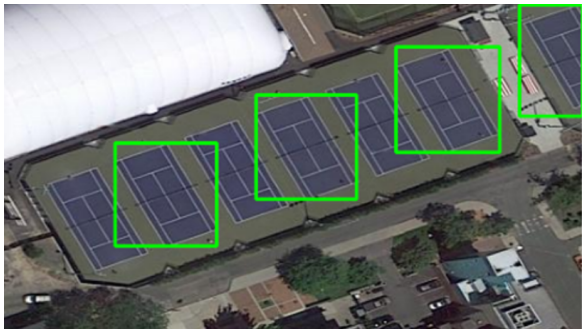
recall and under-detection.

The Oriented Bounding Box (OBB) model solves these problems through its ability to position bounding boxes according to the actual direction of each court. The reduced overlap between objects enables NMS to maintain all valid instances which leads to better localization accuracy and increased detection numbers. The OBB model achieved substantial mAP@50 improvement by increasing the score from 0.701 to 0.777 for Basketball-Court and from 0.879 to 0.908 for Tennis-Court.

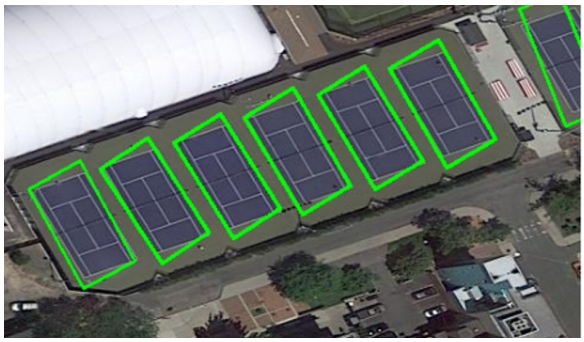
(a) Basketball-Court predictions using HBB (Faster R-CNN).

(b) Basketball-Court predictions using OBB (Rotated Faster R-CNN).

Figure 15: Comparison of basketball-court predictions showing misalignment in HBB vs. accurate orientation-fit in OBB.



(a) Tennis-Court predictions using HBB (Faster R-CNN).



(b) Tennis-Court predictions using OBB (Rotated Faster R-CNN).

Figure 16: OBB enables better alignment and individual court separation, reducing overlap.

Football-Field: For the football-field class, both HBB and OBB perform similarly well, with only a slight increase in mAP@50 (from 0.762 to 0.768). This can be attributed to the fact that football fields in satellite images are typically large, well-separated, and mostly axis-aligned. The limited variation in orientation means that horizontal bounding boxes are already a close fit, and rotating them offers minimal additional benefit.

(a) Football-Field predictions using HBB (Faster R-CNN).

(b) Football-Field predictions using OBB (Rotated Faster R-CNN).

Figure 17: Both models perform well due to the large, axis-aligned layout of the football field.

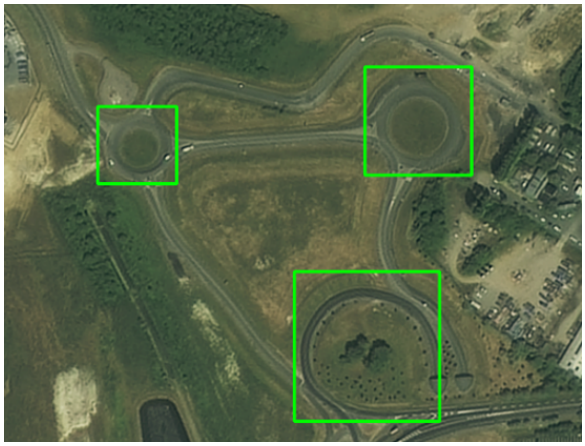
Ship: The ship category proves difficult to analyze because harbor spaces contain many objects and ships of different dimensions and directions. In the HBB prediction (below), we can observe that the two large vessels near the center are detected as a single object, resulting in merged detections where separate ships are not individually identified. The axis-aligned nature of HBBs produces extensive overlapping boxes when ships dock near each other or when they face at angles. The NMS system selects the highest-scoring bounding box for retention while suppressing all other nearby boxes that overlap. The bounding box detection system produces a single detection that covers multiple adjacent ships which results in missed detections. The OBB model outperforms HBB by providing individual vessel detection through precise rotated bounding boxes that match ship orientations. The reduced overlap between bounding boxes enables NMS to maintain all valid detections which results in better localization accuracy and a performance boost from 0.534 (HBB) to 0.659 (OBB) in mAP@50.

(a) Ship predictions using HBB (Faster R-CNN).

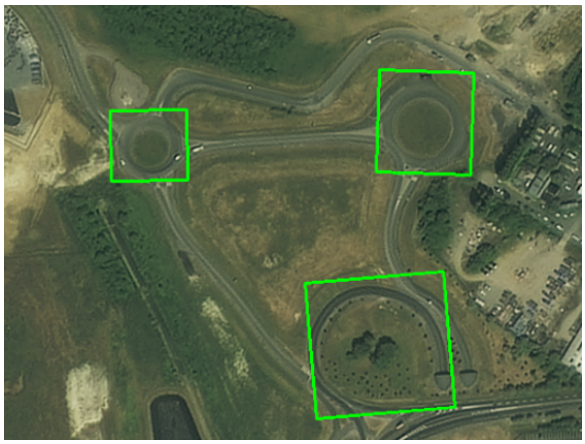
(b) Ship predictions using OBB (Rotated Faster R-CNN).

Figure 18: HBB fails to distinguish closely docked ships, while OBB separates and localizes them precisely.

5. Roundabout and Intersection: For the Roundabout class, both models, HBB (Faster R-CNN) and OBB (Rotated Faster R-CNN) perform almost equally well, with minimal difference in mAP@50 (0.871 for HBB vs. 0.870 for OBB). This parity is largely due to the symmetric, circular nature of roundabouts, which are naturally centered within axis-aligned boxes. As seen in the visual examples below, both models can enclose the structures precisely, with only slight differences in alignment. The benefit of using rotated boxes becomes negligible here since circular or near-symmetric features fit tightly into horizontal rectangles without significant background noise or localization error.



(a) Roundabout predictions using HBB (Faster R-CNN).



(b) Roundabout predictions using OBB (Rotated Faster R-CNN).

Figure 19: Both HBB and OBB models successfully localize roundabouts due to their symmetric shapes.

The HBB model demonstrates a slight improvement in the case of Intersection through its results (0.684 vs. 0.665). The HBB model achieves better results in axis-aligned crossroad layouts that make up most of the intersection regions. The axis-aligned properties of HBB create a slight advantage which allows it to detect square-like or rectangular traffic intersections without needing rotation parameters. The OBB predictions show reduced precision because they produce unnecessary angular offsets or misalignment when the orientation approaches 0° or 90°. The comparison shows HBB achieving better intersection localization with precise fits yet OBB predictions sometimes result in small spatial misplacements.

5.1.2 Discussion and Model Selection Justification

The comparison results in Table 8 show that the oriented bounding box (OBB) detection approach offers clear advantages in cases where object orientation varies widely or where complex layouts lead to overlapping objects. While the horizontal bounding box (HBB) model achieved slightly higher scores for certain structured and axis-aligned classes such as Airplane , Baseball-Field , Bridge, Intersection, and Roundabout, these differences are generally modest and can be attributed to the fact that such objects tend to follow clear, consistent alignments and are less affected by the lack of orientation modeling.

The OBB model shows lower performance in Bridge because angled proposals create false positives when objects match the orientation of roads or water edges. The research confirms previous studies which show that excessive bounding box angle parameters can negatively affect detection of straight or uniformly aligned infrastructure [12].

(a) Intersection predictions using HBB (Faster R-CNN).

(b) Intersection predictions using OBB (Rotated Faster R-CNN).

Figure 20: HBB captures the box-like layout of intersections more effectively due to axis-aligned design.

The OBB model provides better results for categories that have both high intra-class variation and directional diversity. The Vehicle category shows a significant performance gap which proves that rotation-aware proposals and region alignment improve IoU consistency and decrease misclassification in dense parking areas and construction zones [53].

Additional improvements are seen for Basketball-Court, Football-Field, Ship, and Tennis-Court, all of which benefit from orientation alignment due to frequent oblique appearances in satellite views.

The research confirms that HBB detection works well for big objects with minimal occlusion and regular alignment but OBB detection provides better results in real-world remote sensing applications with skewed or densely packed or irregularly oriented instances.

To fully capitalize on these advantages and address residual challenges, this thesis adopts the RoI Trans Swin model within the MMRotate framework for further experiments. This model enhances the standard OBB pipeline by integrating a Swin Transformer backbone for rich contextual feature extraction with the RoI Transformer head for orientation-sensitive proposal refinement. Together, these elements are expected to improve detection robustness and accuracy in diverse, cluttered, and infrastructure-heavy satellite imagery scenes explored in subsequent chapters.

5.2 Results of RoI Trans Swin Model

Table 9 presents a comprehensive breakdown of the per-class detection performance achieved by the RoI Trans Swin model. The table shows the total number of annotated ground truth instances and model-generated detections along with recall rate and average precision (AP) at 0.5 IoU threshold for each object category.

The model demonstrates excellent recall performance for dominant categories, it achieves recall values of 0.972 for Airplane, 0.962 for Tennis-Court and 0.863 for Vehicle. The AP scores show that the model achieves precise localization results for both structured and complex categories. The model achieves high precision scores for structured targets like Airplane (0.909) and Tennis-Court (0.909) and shows notable improvements for challenging and cluttered classes such as Vehicle (0.806) and Bridge (0.437).

Table 9: Detailed per-class detection results of the RoI Trans Swin model, including ground truth counts (gts), detections (dets), recall, and AP at IoU 0.5.

Class	Ground Truth (gts)	Detections (dets)	Recall	AP@50
Airplane	2971	3208	0.972	0.909
Baseball-Field	131	174	0.954	0.900
Basketball-Court	132	192	0.856	0.782
Bridge	88	268	0.591	0.437
Football-Field	86	180	0.919	0.863
Intersection	659	1488	0.777	0.659
Roundabout	61	88	0.918	0.873
Ship	3153	4391	0.772	0.694
Tennis-Court	236	267	0.962	0.909
Vehicle	34480	40393	0.863	0.806
Overall mAP@50	–	–	–	**0.783**

The normalized confusion matrix in Figure 21 helps to better understand inter-class confusion patterns. The majority of classes show strong diagonal dominance which indicates high classification accuracy. The model demonstrates high accuracy in classifying Airplane (97%), Football-Field (84%) and Roundabout (87%) with very little confusion. The model shows significant confusion in specific classes because it mistakes Bridge for background (38%) and also Intersection and Ship with background at a rate of 22%. The lower object count and more variable visual patterns of these classes most likely cause this confusion. The class imbalance and visual similarity to surrounding context create similar problems for Basketball-Court which shows 13% confusion with background.

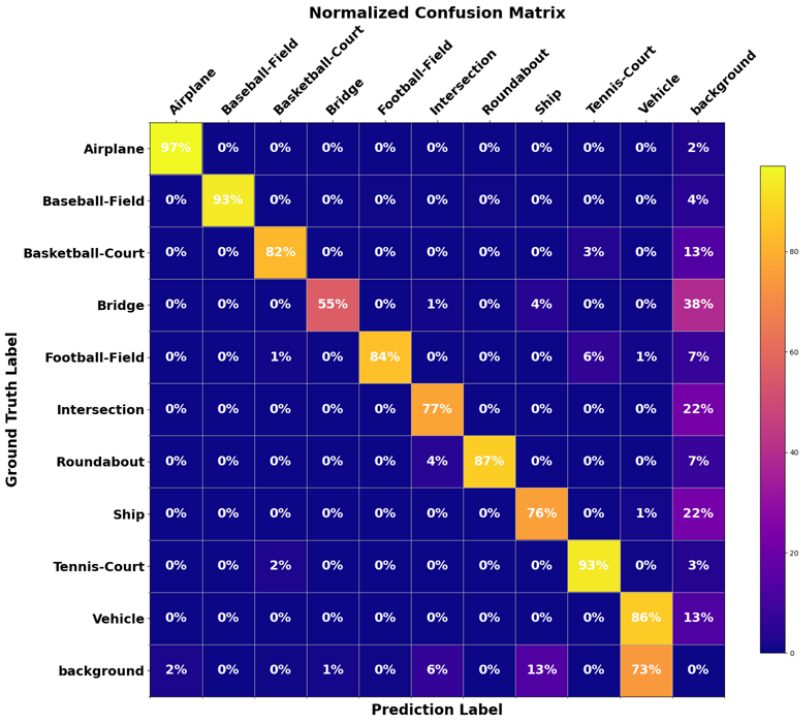


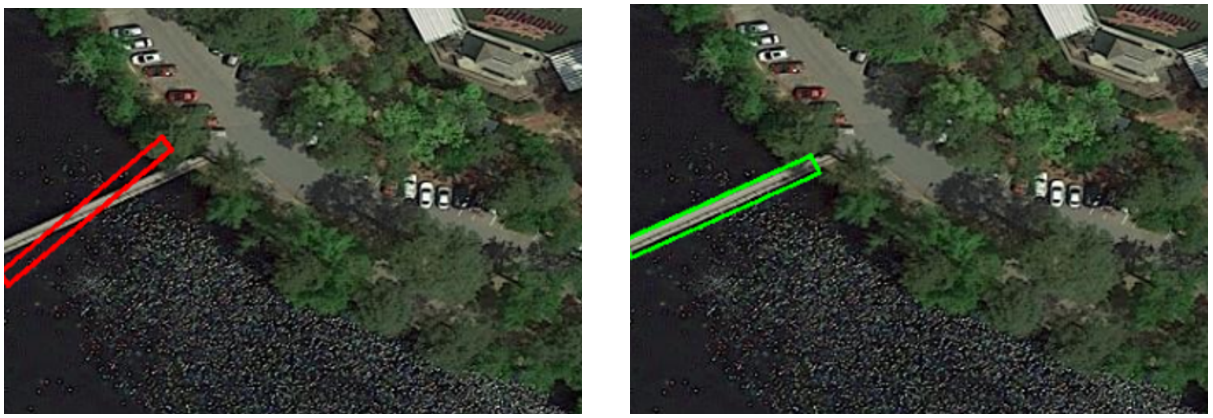
Figure 21: Normalized confusion matrix of RoI Trans Swin model on the FAIR1M test set.

The comparison between RoI Trans Swin model and Rotated Faster R-CNN baseline (R-FRCNN) per-class mAP@50 results is presented in Table 10. The RoI Trans Swin model demonstrates better performance across most categories while achieving significant AP improvements in Bridge, Football-Field and Ship classes because it benefits from its ability to perform fine-grained geometric alignment and enhanced global context understanding.

Table 10: Comparison of class-wise mAP@50 between Rotated Faster R-CNN (R-FRCNN) and RoI Trans Swin.

Class	**R-FRCNN (OBB)**	**RoI Trans Swin**	**Improvement**
Airplane	0.908	0.909	+0.001
Baseball-Field	0.896	0.900	+0.004
Basketball-Court	0.777	0.782	+0.005
Bridge	0.312	0.437	+0.125
Football-Field	0.768	0.863	+0.095
Intersection	0.665	0.659	-0.006
Roundabout	0.870	0.873	+0.003
Ship	0.659	0.694	+0.035
Tennis-Court	0.908	0.909	+0.001
Vehicle	0.793	0.806	+0.013
Overall mAP@50	0.756	**0.783**	+0.027

In Figure 22, it is evident that the RoI Trans Swin model successfully detects the bridge by aligning its prediction with the correct orientation whereas the Rotated Faster R-CNN fails to do so. The bounding box predicted by R-FRCNN is misaligned and does not meet the IoU threshold which leads to a false negative (FN). In contrast, the RoI Trans Swin prediction achieves proper alignment and surpasses the 0.5 IoU threshold and being counted as a true positive (TP).



(a) Rotated Faster R-CNN bridge prediction. (b) RoI Trans Swin bridge prediction.

Figure 22: Comparison of bridge detection performance between Rotated Faster R-CNN and RoI Trans Swin. The red bounding box in (a) shows a misaligned prediction with IoU < 0.5, which is not counted as a true positive (TP). The green bounding box in (b) shows a correctly aligned prediction from RoI Trans Swin that exceeds the IoU threshold and is classified as TP.

The additional performance evaluation demonstrates that the transformer backbone improves the model's ability to detect distant relationships and global spatial information which supports the precise orientation modeling of the RoI Transformer head. The synergy between these components solves the problems that CNN-based OBB detectors face when operating in cluttered or geometrically complex remote sensing scenes.

In light of these improvements, the RoI Trans Swin is selected as the final architecture for the remaining experiments in this thesis. Its hybrid design provides a robust foundation for further investigation into hybrid detection pipelines, advanced post-processing methods, and deployment strategies for practical remote sensing applications.

The RoI Trans Swin architecture receives selection as the final model for upcoming experiments in this thesis because of its enhancements. The hybrid structure of this model establishes a solid base to study hybrid detection pipelines and practical deployment methods for remote sensing applications.

5.3 Optimizer Impact on Oriented Object Detection Models

To examine the impact of optimizer selection on detection performance, an experiment was conducted comparing SGD and AdamW optimizers across both the Rotated Faster R-CNN (R-FRCNN) and RoI Trans Swin Tiny models. The configurations fosr each optimizer were kept consistent across the models, as described in the experimental setup, ensuring a fair and controlled comparison. This evaluation is particularly relevant as R-FRCNN is conventionally trained with SGD, while the RoI Trans Swin model typically uses AdamW.

The table below summarizes the mean Average Precision (mAP) results obtained for each combination:

Table 11: Performance comparison of R-FRCNN and RoI Trans Swin Tiny models with SGD and AdamW optimizers.

Model	**SGD (mAP)**	**AdamW (mAP)**
R-FRCNN	0.756	0.743
RoI Trans Swin Tiny	0.766	0.783

The results show that R-FRCNN performs slightly better with SGD (0.756 mAP) than with AdamW (0.743 mAP), indicating that traditional convolution-based architectures benefit from the regularizing effect of SGD and its momentum-driven updates. This aligns with the general observation that SGD remains highly effective for CNN-dominated architectures, especially when feature maps are refined incrementally.

The RoI Trans Swin Tiny model which used a transformer-based backbone achieved better accuracy with AdamW (0.783 mAP) than with SGD (0.766 mAP). The difference can be attributed to the AdamW optimizer's better handling of weight decay for transformer-based models, where layer normalization and attention mechanisms are sensitive to learning dynamics. AdamW's adaptive learning and decoupled weight decay likely help the model converge more effectively and generalize better in these scenarios.

Based on this outcome, all subsequent experiments in the thesis involving the RoI Trans Swin Tiny model were conducted using the AdamW optimizer to ensure optimal performance.

5.4 Additional Neck Variants in RoI Trans Swin

To evaluate the individual contributions of advanced neck modules within the RoI Trans Swin architecture, three configurations were tested, the standard Feature Pyramid Network (FPN), the Path Aggregation Feature Pyramid Network (PAFPN), and FPN equipped with CARAFE upsampling. This comparative analysis assesses whether sophisticated feature aggregation and content-aware upsampling can further improve detection accuracy in high-resolution remote sensing scenes.

Table 12: Comparison of mean Average Precision (mAP@50) for different neck configurations in the RoI Trans Swin model.

Neck Configuration	**mAP@50**
Feature Pyramid Network (FPN)	**0.783**
Path Aggregation FPN (PAFPN)	0.771
FPN with CARAFE	0.733

5.4.1 Effects of FPN

The standard FPN was used as the baseline and achieved the highest mAP@50 score of 0.783. Its proven top-down structure with lateral connections does a good job of combining features at different scales, working well alongside the Swin Transformer's hierarchical window design. This indicates that a conventional FPN remains a robust and efficient choice for scale-aware detection in this hybrid architecture.

5.4.2 Effects of PAFPN

The replacement of FPN with PAFPN caused a minor decrease in mAP@50 to 0.771. The PAFPN model extends FPN through its bottom-up augmentation path which strengthens lower-level feature maps with more semantic information. The Swin Transformer already processes extensive cross-scale and long-range context so the added complexity results in mild over-smoothing instead of delivering distinct advantages.

5.4.3 Effects of FPN CARAFE

Integrating CARAFE into the FPN neck produced the lowest mAP@50 of 0.733. CARAFE is designed to enhance upsampling by dynamically aggregating neighborhood features based on content. The transformer-guided pipeline restricted the advantages of content-aware reassembly which resulted in minor interpolation artifacts that reduced boundary localization precision.

The results in Table 12 demonstrate that the classic FPN stands as the most appropriate neck for the Swin-based RoI Trans detector. The model provides the optimal combination of multi-scale feature consistency and architectural simplicity which results in accurate orientation-aware object detection in satellite imagery.

5.5 Effects of Backbone Variation: Swin-Tiny vs. Swin-Small in RoI Trans Swin

To evaluate how increasing backbone capacity influences detection performance in the RoI Trans Swin architecture, an additional experiment was performed by replacing the Swin-Tiny backbone with the larger Swin-Small variant. In the Swin Transformer family, Swin-Small differs from Swin-Tiny primarily in its increased depth and wider feature dimensions, it has more transformer blocks per stage and a larger channel width, enabling it to capture richer feature hierarchies and finer spatial detail [31].

Table 13: Per-class detection performance of RoI Trans Swin using Swin-Small backbone (IoU threshold 0.5).

Class	**Ground Truth (gts)**	**Detections (dets)**	**Recall**	**AP@50**
Airplane	2971	3148	0.972	0.909
Baseball-Field	131	160	0.954	0.899
Basketball-Court	132	176	0.848	0.777
Bridge	88	230	0.636	0.507
Football-Field	86	158	0.884	0.805
Intersection	659	1517	0.800	0.674
Roundabout	61	93	0.967	0.879
Ship	3153	3991	0.774	0.699
Tennis-Court	236	258	0.958	0.909
Vehicle	34480	38806	0.842	0.803
Overall mAP@50	–	–	–	**0.786**

As shown in Table 13, using Swin-Small yielded an overall mAP@50 of 0.786, a negligible increase over the 0.783 achieved with Swin-Tiny. This slight improvement can be attributed to the fact that the Swin-Tiny backbone already provides sufficiently strong multi-scale representations for the FAIR1M dataset, therefore, simply increasing depth and channel capacity has limited impact when the Feature Pyramid and RoI Transformer head are already effective at refining spatial context.

The Bridge class demonstrates an interesting improvement from 0.437 (Tiny) to 0.507 (Small). The model achieves better results at detecting thin and low-contrast structures like bridges through its increased depth and attention capacity because these structures commonly appear in cluttered urban or river environments.

The Football-Field class experienced a decrease in its score from 0.863 to 0.805. The larger transformers might cause mild over-smoothing and slight boundary blurring which results in reduced localization precision for large open objects because they diffuse sharp edges in broad homogeneous areas.

In summary, while Swin-Small theoretically offers higher representational power than Swin-Tiny, the gains in this specific remote sensing context are marginal and not uniformly distributed across classes. This result indicates that in this dataset, the scenes and objects are already handled well by the Swin Tiny backbone, so using a larger backbone adds little benefit but increases computation.

5.6 Performance on Reduced FAIR1M Dataset

To evaluate the robustness of the RoI Trans Swin model in low-resource settings, we conducted an additional experiment using a significantly smaller version of the FAIR1M dataset. This reduction simulates an industrial scenario with limited labeled training data. The class-wise distribution was reduced across training, validation, and test sets, as detailed in Table 6.

Table 14 presents a comparison of Average Precision (AP) for each class on both the reduced and full FAIR1M datasets. The results reveal a substantial decline in detection performance when fewer training examples are available, with the overall mAP dropping from 0.783 on the full dataset to 0.682 on the smaller one.

Table 14: Comparison of AP scores per class between full and reduced FAIR1M datasets.

Class	**AP (Small Dataset)**	**AP (Full Dataset)**
Airplane	0.797	0.909
Baseball-Field	0.900	0.900
Basketball-Court	0.686	0.782
Bridge	0.293	0.437
Football-Field	0.807	0.863
Intersection	0.461	0.659
Roundabout	0.870	0.873
Ship	0.395	0.694
Tennis-Court	0.815	0.909
Vehicle	0.799	0.806
mAP	**0.682**	**0.783**

The reduction in dataset size led to a notable performance drop across most categories. Particularly, classes like Ship (AP: 0.395 vs. 0.694) and Bridge (AP: 0.293 vs. 0.437) suffered the most. On the other hand, classes like Baseball-Field and Roundabout maintained relatively stable APs.

The Vehicle class maintained a strong AP (0.799 vs. 0.806) because of its overwhelming representation in the dataset, even after reduction. This indicates that the absolute number of training instances per class plays a significant role in maintaining detection accuracy.

The research demonstrates that large diverse datasets are essential for developing detection models which achieve high performance in aerial imagery complex scenes.

5.7 Performance on Worksite Dataset

The RoI Trans Swin Tiny model was evaluated on the test split of the worksite dataset to assess its effectiveness in detecting infrastructure-related objects such as groundworks, construction-works, and road-works. Table 15 summarizes the class-wise detection performance in terms of number of ground truth instances (gts), predicted detections (dets), recall, and average precision (AP). The final row reports the overall mean Average Precision (mAP).

The results indicate that the model performs moderately well on the groundworks and construction-works classes, achieving recall values close to 0.5, meaning that nearly

Table 15: Detection performance of RoI Trans Swin Tiny on the worksite dataset.

Class	**gts**	**dets**	**Recall**	**AP**
Groundworks	74	213	0.473	0.313
Construction-works	11	42	0.455	0.390
Road-works	2	20	0.000	0.000
mAP				**0.234**

half of the actual instances were correctly identified. The higher detection count for the groundworks class also includes predictions for potentially valid instances that may have been missed during manual annotation. This highlights the model's capacity to generalize beyond the labeled training data and detect plausible features characteristic of groundworks.

The model experienced alignment issues with its bounding boxes because groundworks regions exhibit significant variability in their shape, size and texture. The model failed to meet the 0.5 IoU threshold for true positive classification which resulted in several predictions below the threshold and decreased both precision and AP.

The performance on the construction-works class was slightly better in terms of Average Precision (AP), despite having significantly fewer training samples compared to the groundworks class. This indicates that construction-works may exhibit more consistent and distinguishable structural characteristics such as the presence of cranes, scaffolding, or construction machinery that the model could effectively learn even from limited data. These features likely offer stronger visual cues, enhancing the model's ability to detect and classify construction-works instances with greater accuracy even from limited data.

In contrast, the model completely failed to detect the road-works class in the test set. The recall and AP scores for this category were both zero. This underperformance can largely be attributed to the very small number of training instances (only 75 in the training set and 2 in the test set), which limited the model's ability to generalize and recognize this class. Furthermore, road-works are often narrow and irregularly shaped, increasing the difficulty of precise localization under sparse training data conditions.

The RoI Trans Swin Tiny model showed potential for detecting infrastructure-related features including groundworks and construction-works yet the results highlighted the difficulties stemming from restricted data availability and high object variability especially for road-works that need precise localization accuracy.

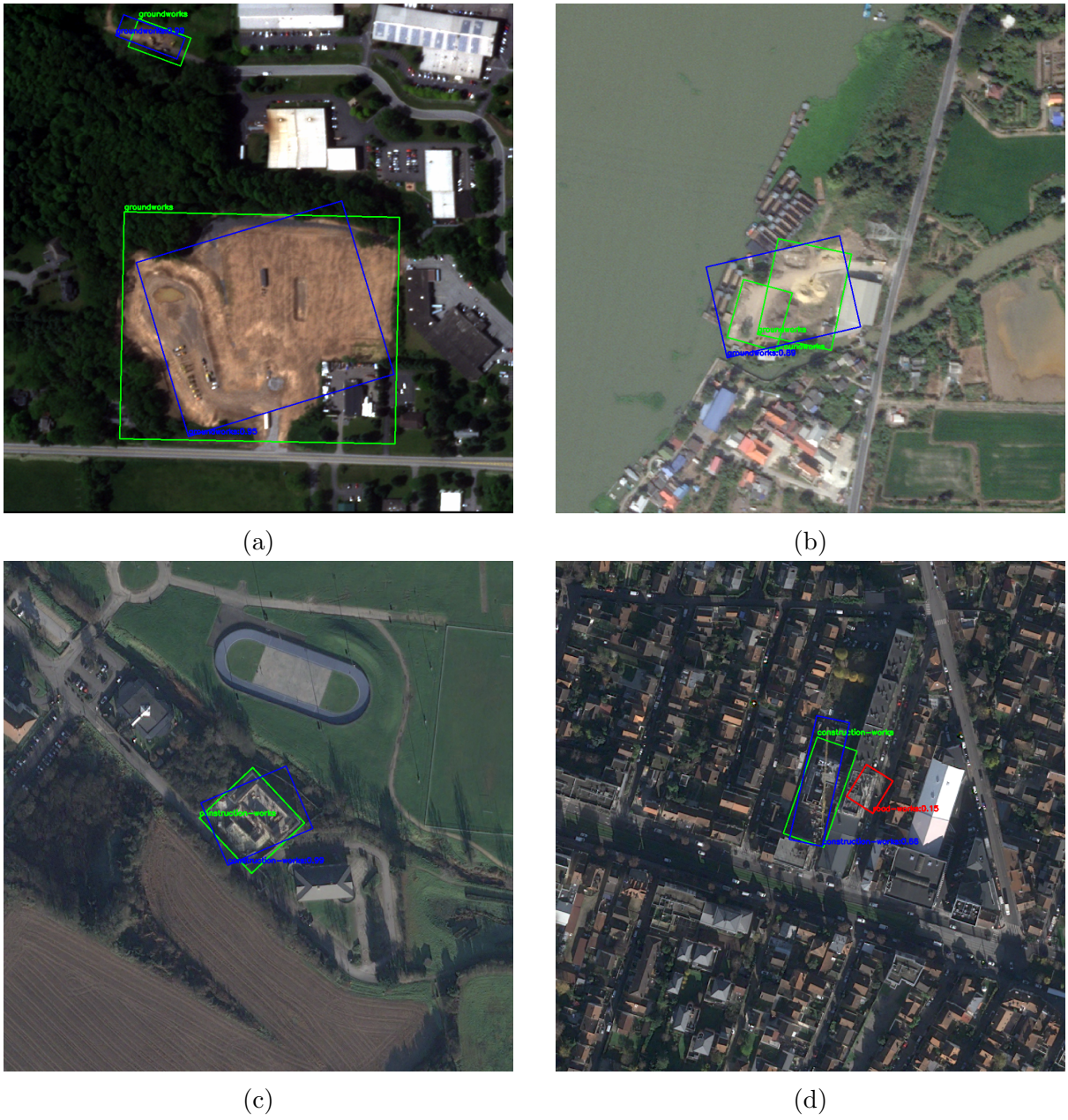


Figure 23: Visualized inference outcomes on the Worksite dataset. Green bounding boxes indicate ground truth annotations. Blue bounding boxes represent correct predictions with Intersection over Union (IoU) ≥ 0.5. Red bounding boxes denote predictions that do not sufficiently overlap with any ground truth object (IoU < 0.5). This visualization highlights both accurate detections and common failure cases. Images acquired from © 2024 Planet Labs PBC and © Airbus DS (2024).

6 Conclusion

This thesis introduces a comparative study on horizontal bounding box (HBB) and oriented bounding box (OBB) detection techniques for high-resolution satellite imagery with a focus on evaluating the effectiveness of hybrid CNN-Transformer architectures. The work implements Faster R-CNN from MMDetection for HBB detection and Rotated Faster R-CNN from MMRotate for OBB detection. The study explores the impact of

bounding box representation, neck and backbone variants and dataset scale on detection accuracy with the FAIR1M dataset serving as the primary benchmark. The research further assesses the practicality of advanced architectures like the Swin Transformer backbone integrated with the RoI Transformer detection head in complex aerial scenes.

The findings provide comprehensive answers to the research questions. First, the comparison between HBB and OBB models reveals that OBB detection generally outperforms HBB, especially for object categories with arbitrary orientations or cluttered arrangements. This supports the hypothesis that bounding box representation significantly affects detection accuracy. Second, the integration of a Transformer-based backbone in the RoI Trans Swin model demonstrates clear performance gains over standard CNN-based OBB models, especially in classes like Bridge, Ship and Football-Field, validating the benefits of global context modeling. Third, the experiment comparing the impact of optimizers revealed that SGD led to better performance for the R-FRCNN model, achieving a mAP of 0.756, while AdamW yielded superior results for the RoI Trans Swin Tiny model with a mAP of 0.783. This confirms that optimizer choice should be based on architecture, SGD is better suited for CNN-based models, while AdamW works more effectively for transformer-based models. Fourth, experiments with neck variations (FPN, PAFPN, FPN-CARAFE) show that the standard FPN yields the best performance in Swin-based models, while more complex necks offer diminishing or even adverse returns. Fifth, switching the Swin-Tiny backbone to Swin-Small results in only marginal gains this indicates that Swin-Tiny already captures sufficient spatial detail for FAIR1M-scale tasks. Lastly, training on a reduced FAIR1M dataset causes a marked decline in mAP (from 0.783 to 0.682) confirming that the RoI Trans Swin model, while robust, remains data-dependent and struggles to generalize under low-resource conditions, particularly for classes like Ship and Bridge that require greater intra-class diversity. The model was still able to predict the Vehicle class well. When applied to the Worksite dataset, the RoI Trans Swin model achieved recall close to 50% for groundworks and construction-works, indicating moderate success despite limited training data. However, it failed to detect any road-works, primarily due to the scarcity of labeled instances. Given the performance improvement observed from the reduced to the full FAIR1M dataset, it is reasonable to conclude that increasing the volume of annotated samples in the Worksite dataset could similarly lead to enhanced detection performance across all classes.

The research contains several limitations despite its valuable findings. The results depend on a single run per configuration which leads to potential variance because of weight initialization and batch ordering and random augmentation stochastic elements. The mAP scores reported in this study may not accurately represent the model's average performance or stability. The models received benchmarking on FAIR1M but the lack of additional datasets restricts the broad applicability of the research findings.

The research can advance through multiple directions. The model will achieve better generalization performance through data augmentation methods including rotation jitter and CutMix and Mosaic augmentation especially when dealing with class imbalance or object occlusion. Another area for improvement is in post-processing, where methods tailored to MMRotate such as orientation-aware Non-Maximum Suppression or uncertainty-based filtering could help reduce false positives and improve localization accuracy. Lastly, lightweight Transformer backbones or knowledge distillation strategies could be employed to accelerate inference and enable deployment on resource-constrained platforms.

In summary, The research demonstrates through empirical and analytical methods

that OBB detection using RoI Trans Swin hybrid models produces superior results in detecting objects within intricate remote sensing images. The research demonstrates that orientation modeling together with architectural design and dataset diversity enable the development of accurate deployable object detection systems for satellite imagery.

7 Acknowledgements

I would like to express my sincere gratitude to the entire team at Orbital Eye for providing me with the opportunity to work on this research and explore a subject I am deeply passionate about. Their support and resources have been invaluable throughout the project. I am also deeply thankful to Dr. D.M. Pelt and Dr. H.R. Doughty for their outstanding supervision and consistent support. Their thoughtful feedback, regular discussions, and critical insights have significantly shaped the direction and quality of this work. This thesis would not have been possible without their guidance and mentorship.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[5] Kai Chen, Xue Yang, Wen Yang, and Junchi Yan. Rotated detr: Detection transformer for oriented objects. In *AAAI*, 2022.

[6] Xiaojie Chen, Lingxi Xie, Zihang Liu, Jianmin Bao, Dongdong Guo, Jifeng Dai, Xingang Dong, and Lu Yuan. An empirical study on training vision transformers. *arXiv preprint arXiv:2104.03402*, 2021.

[7] Gong Cheng, Junwei Han, and Xiaoqiang Lu. A survey of remote sensing image classification: From statistical modeling to deep learning. *Remote Sensing*, 12(13):2209, 2020.

[8] Gong Cheng, Peicheng Zhou, and Junwei Han. Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):7405–7415, 2016.

[9] MMRotate Contributors. Mmrotate: Openmmlab rotated object detection toolbox and benchmark. https://github.com/open-mmlab/mmrotate, 2022.

[10] Tianyang Dai, Xin Chen, Yue Zhang, Runmin Wu, and Xian Gao. Roi trans: Rotated object detection with deformable roi pooling. In *IEEE GRSL*, 2022.

[11] Jian Ding, Nanxuan Chen, Xiang Liu, Zezhong Wu, Xiaoqiang Yang, Junwei Han, and Errui Ding. Fair1m: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 178:102–117, 2021.

[12] Jian Ding, Nan Xue, Ying Long, and Gui-Song Xia. Learning roi transformer for oriented object detection in aerial images. In *CVPR*, pages 2849–2858, 2019.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[14] Mark Everingham, Luc Van Gool, Christopher K.I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–338, 2010.

[15] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[16] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017.

[17] Junjie Han, Jian Ding, Nan Xue, and et al. Redet: A rotation-equivariant detector for aerial object detection. *CVPR*, 2021.

[18] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[21] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 448–456, 2015.

[22] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2017–2025, 2015.

[23] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[25] Kun Li, Jian Wang, Yuxiang Qin, Yong Liu, Xiang Bai, and Tianfei Zhou. Highly efficient anchor-free oriented small object detection for remote sensing images via periodic pseudo-domain. *arXiv preprint arXiv:2308.01134*, 2023.

[26] Yuntao Li, Qing Guo, and Feng Zhang. A novel multiple targets detection method for service robots in the indoor complex scenes. *ResearchGate Preprint*, 2023. Accessed: 2025-07-10.

[27] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. *European Conference on Computer Vision*, pages 740–755, 2014.

[30] Shu Liu, Lu Qi, Haifang Qin, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8759–8768, 2018.

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[32] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2018.

[33] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hongyang Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, 20(11):3111–3122, 2018.

[34] Qi Ming, Xinlong Zheng, Pan Wang, and Dahua Lin. Dynamic r-cnn: Towards high quality object detection via dynamic training. *arXiv preprint arXiv:2103.10158*, 2021.

[35] Shahin Mirzadeh, Maxim A Dulebenets, and Javad Pasha. Parking occupancy detection and slot delineation using deep learning: A tutorial. *IEEE Intelligent Transportation Systems Magazine*, 2021.

[36] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *ICML*, pages 1310–1318, 2013.

[37] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, volume 28, 2015.

[39] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[40] Yifan Sun, Yibing Wang, Ying Wang, and et al. Rq-transformer: Attention-based detection for rotated objects in aerial images. In *IEEE Transactions on Geoscience and Remote Sensing*, 2022.

[41] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning*, 2019.

[42] Hugo Touvron, Matthieu Cord, Matthijs Douze, and et al. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[44] Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin. Carafe: Content-aware reassembly of features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3007–3016, 2019.

[45] Qiang Wang, Xinyu Liu, Shaohui Zhao, et al. Center-ness and repulsion constraints to improve remote sensing object detection via reppoints. *ResearchGate Preprint*, 2023.

[46] Xiangyu Wang, Hailong Liu, Xiang Bai, and Jianan Ding. Highly efficient anchor-free oriented small object detection for remote sensing images via periodic pseudo-domain. *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

[47] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3974–3983, 2018.

[48] Enze Xie, Jian Ding, Wenhai Wang, Xiaojie Zhan, and et al. Oriented r-cnn for object detection. *arXiv preprint arXiv:2105.11111*, 2021.

[49] Lingxi Xie, Xiaojie Chen, Xiangyu Zhang, and et al. Simt: Simple training enhancements for vision transformers. *arXiv preprint arXiv:2206.02089*, 2022.

[50] Yongchao Xu, Weijian Zhou, Zheng Yang, and Zhiqiang Zhou. Gliding vertex on the horizontal bounding box for multi-oriented object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5580–5589, 2020.

[51] Xue Yang, Dengxin He, Ying Zhou, and et al. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[52] Xue Yang, Hao Sun, Kebin Fu, Wen Yang, Zhen Guo, and Junchi Yan. Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks. *Remote Sensing*, 10(1):132, 2018.

[53] Xue Yang, Jirui Yan, Xinyu Zhao, Ruofeng Tang, and Jingyu Yang. R3det: Refined single-stage detector with feature refinement for rotating object. *IEEE Transactions on Image Processing*, 30:1305–1318, 2021.

[54] Tianfei Zhou, Xiang Bai, and Yong Liu. Oriented object detection: A literature review. *arXiv preprint arXiv:2209.10408*, 2022.

[55] Xinlong Zhou, Wenwei Zhang, Jifeng Dai, et al. Mmrotate: A rotated object detection toolbox based on pytorch. *arXiv preprint arXiv:2204.00160*, 2022.

[56] Xiao Xiang Zhu, Devis Tuia, Lichao Mou, Gui-Song Xia, Liangpei Zhang, Feng Xu, and Friedrich Fraundorfer. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36, 2017.