CODI: Contextual Object Detection via Image Inpainting

Sinan Çavdar and Selim Aksoy, *Senior Member, IEEE*

***Abstract*—Object detection is a fundamental task in many applications of remote sensing image analysis. Problems such as inter-class similarity, intra-class variability, and existence of heterogeneous backgrounds can benefit from leveraging contextual information to improve the detection performance. Most context modeling architectures introduce additional modules that aim to extract contextual information from the main backbone that is also used for learning the object representations. We propose a new approach for contextual object detection via image inpainting named CODI where a custom module fuses object-specific features from an object detector and contextual representations from an inpainting-based generative model. The resulting multi-scale representation named the contextual pyramid network is injected into the feature extraction backbone so that both the oriented region proposal network and the classification and regression branches of the detector can benefit from the richer representations resulting from this fusion. In contrast to others, our architecture obtains semantic context independently from and without relying on the backbone of the object detection model to guide both localization and labeling. Extensive experiments with quantitative and qualitative evaluation are performed on the DOTA, HRSC2016, and DIOR-R datasets. With mean average precision scores of 76.61%, 90.57%, and 67.63% on these datasets respectively, the proposed model achieves higher performance compared to other models from the literature. Ablation experiments also show that CODI enables improvements in precision and recall with effective control of the score probability threshold during detection.**

***Index Terms*—Object detection, image inpainting, feature fusion**

I. Introduction

Advancements in remote sensing technology have driven many applications in areas such as environmental monitoring, urban development, agriculture, and disaster management. One of the most fundamental problems in these applications is object detection where accurate identification of objects is essential for extracting meaningful information about challenging scenes. Object detection is typically studied as a combination of two sub-problems: localization and classification. However, object detection in remote sensing presents several unique challenges compared to traditional object detection applications.

One of the main challenges in remote sensing object detection is the existence of complex and heterogeneous backgrounds that make it difficult to distinguish objects from the environment. For example, vehicles parked along densely built urban areas can easily blend with their surroundings. Furthermore, objects of interest can have a very high variability in scale, ranging from small vehicles that may occupy only a few pixels to sports fields that are composed of multiple large structures. Finally, visual similarities between different object categories (inter-class similarity) and large variations within the same class (intra-class variability) increase the complexity of the detection and classification tasks. For instance, different types of sports fields can share certain characteristics, and vehicles can significantly vary in color and size.

Solutions to these challenges can benefit from leveraging contextual information to improve the detection performance. For example, vehicles often appear on or near roads, buildings form different types of clusters in different urban settings, and ships appear with certain alignments in docks and harbors. Consequently, context models that exploit spatial and semantic relationships between objects and their surroundings can provide additional cues for accurate localization and classification.

Recent research on remote sensing object detection has been dominated by deep learning architectures. Popular one-stage and two-stage model such as Faster-RCNN [1], You Only Look Once (YOLO) [2], and Retina-Net [3] have been adopted for remote sensing images to overcome large-scale scenes, high-resolution data, and multispectral inputs. Recent improvements in this field focus more on improving small object detection, optimizing models for computational efficiency, and facilitating multi-scale feature learning. There has also been a trend in integrating contextual information for further improvements in performance. Most of these works introduce additional modules that aim to extract contextual information from the main backbone architecture that is also used for learning the object representations [4]–[8]. These modules are typically connected to the later stages of the detection pipeline such as bounding box regression or classification heads to improve object labeling.

Our main motivation in this work is to improve both the localization and the classification of objects by integrating an inpainting-based generative model with a two-stage object detection architecture. Image inpainting has been shown to have a potential for capturing image context while filling missing regions based on their surroundings [9]. We propose a custom fusion module that integrates the contextual information captured in the transformer layers of the Mask Aware Transformer inpainting model [10] with the object representations extracted in the feature pyramid network of the Oriented R-CNN object detector model [11]. The fusion module integrates pairs of transformer units and pyramid levels based on the compatibility of their feature maps. The resulting multi-scale feature representation named the contextual pyramid network is used to replace the feature pyramid network, and is injected into the feature representation backbone given as input to both the object proposal generation stage and the regression and classification branches to guide both localization and detection. This context-enhanced multi-scale representation benefits object detection as larger objects

The authors are with Department of Computer Engineering, Bilkent University, Ankara, 06800, Türkiye. Email: sinan.cavdar@bilkent.edu.tr, saksoy@cs.bilkent.edu.tr.

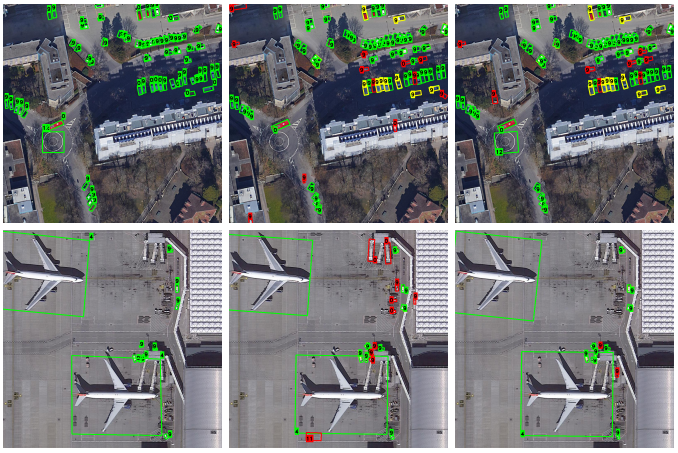


Fig. 1: Example detection results for the proposed model and the oriented R-CNN model for different class probability thresholds on the DOTA validation set where ground truth annotations are available. Green indicates correct detections, yellow indicates detections with wrong class labels, red indicates false positives. Left to right: ground truth, detections of the oriented R-CNN model, detections of the proposed model. On the first row, our model achieves comparable precision with better recall when the threshold is set as 0.04 and 0.02 for oriented R-CNN and our model, respectively. On the second row, our model achieves comparable recall with better precision when the threshold is set as 0.05 and 0.04 for oriented R-CNN and our model, respectively.

are captured in higher layers with broader contexts, while smaller objects appear in lower layers retaining finer details. Consequently, the class probability threshold can be used to control the tradeoff between precision and recall as illustrated in Figure 1, where the proposed model enhances one of these metrics while keeping the other one at a comparable level. By using extensive comparative experiments and ablation studies, we show the effectiveness of the proposed model named CODI for Contextual Object Detection via Image Inpainting.

Compared to previous studies on contextual object detection and image inpainting, our proposed architecture provides a novel strategy to help enhance the objects' representative features for better localization and identification. The main contributions of this work are summarized as follows:

- We propose an architecture to integrate a generative inpainting model with a two-stage object detector to exploit semantic context for remote sensing.
- In contrast to others, our architecture obtains semantic context independently from and without relying on the backbone of the object detection model. This prevents the model from being limited by the feature-extraction biases of the detection backbone.
- We introduce a custom fusion module that creates a multi-scale contextual representation. This replaces the standard feature pyramid network to better handle scale variability, ensuring that large objects benefit from broader context while small objects retain fine-grained details.
- To the best of our knowledge, this is the first work to demonstrate that inpainting-derived features can significantly improve not only the identification (classification) of objects but also their spatial localization (proposal generation) in complex remote sensing scenes.
- Through comprehensive experiments, we demonstrate that our model achieves a superior precision-recall trade-off compared to state-of-the-art models on challenging datasets with heterogeneous backgrounds and high inter-class similarity.

The rest of the paper is organized as follows. Section II discusses the related work on object detection, contextual modeling, and image inpainting. Section III describes the proposed methodology. Section IV presents the experiments and discusses the results. Section V provides the conclusions.

II. Related work

We discuss the related work in three parts: object detection, contextual modeling, and image inpainting in remote sensing.

a) Object detection: Recent years have seen significant developments on object detection in remotely sensed imagery. Xie et al. [11] propose oriented RCNN, a two-stage detector that efficiently generates oriented proposals, resulting in high-quality detections with oriented bounding boxes. Pu et al. [12] introduce the adaptive rotated convolution module, enabling the kernels to rotate adaptively based on an object's orientation, thereby improving feature extraction. Lu et al. [13] describe DecoupleNet, a lightweight backbone network tailored for resource-constrained environments, demonstrating promising classification and detection performance among existing lightweight networks. Xie et al. [14] present a one-stage detector that uses a feature-interaction alignment module to provide mutual assistance between classification and regression heads and jointly optimizes both anchor-based and anchor-free predictions to improve the overall detection accuracy. They also introduce an objectness activation network [15] that is a lightweight fully-convolutional module designed to efficiently filter non-object patches in large aerial images, significantly accelerating inference while maintaining accuracy. Furthermore, they propose a fine-grained object detection network [16] that learns discriminative representations via a dedicated fine-grained branch that is trained by using a confusion-minimized loss for subordinate-level object classification. Wu et al. [17] describe a center-symmetry representation-based localization detector that introduces an enhanced feature pyramid, a center-symmetry proposal generation, and a dual classification head to resolve various inconsistency issues in two-stage oriented detection. A common characteristic of these approaches is that they focus on performing detection based solely on object features. Therefore, they cannot recover false negatives when these objects lack representative features or appear in cluttered backgrounds.

b) Contextual modeling: With the rapid rise of large language models, transformer-based approaches have gained interest in remote sensing object detection. Wang et al. [18] introduce the ViTAE model that adopts multi-scale window attention with varying orientations to reduce computational cost and enhance object representation. Yu et al. [19] propose

spatial transform decoupling using a separate network to predict bounding-box orientations within the detection head and to improve object features via cascaded activation masks. Li et al. [20] describe the large selective kernel network LSKNet that dynamically adjusts the receptive fields of kernels within its convolutional layers under the feature extraction backbone to handle objects of varying scales. Zhao et al. [21] introduce a point-axis representation to handle loss discontinuities and abrupt rotation changes in bounding boxes, and integrate this representation into a detection transformer framework. Zhao et al. [22] present an end-to-end transformer detector that addresses orientation encoding, missing geometric relations, and feature misalignment via positional encodings and attention mechanisms. Zeng et al. [23] describe an angle classification method and a rotated deformable attention module to improve feature alignment for oriented object detection. These approaches expect the self-attention mechanism to capture the dependencies among different parts of the image and implicitly model the contextual information within the detection pipeline.

Some methods propose to add extra modules that aim to focus on the contextual information. For example, Ma et al. [4] introduce a multi-model decision fusion framework that integrates local and relational contextual features and multi-region object parts to handle the complexities in object appearance and spatial structure. Zhang et al. [5] describe CAD-Net that exploits global and local contextual information via attention modules that operate on the feature representations extracted by the ResNet and feature pyramid backbones. Dong et al. [6] propose a gating function to replace the RoiAlign module to incorporate the local context surrounding each proposal and the global context of the whole image. Min et al. [7] use a contextual transformer module to capture spatial attributes and channel characteristics by integrating global residuals and local fusion mechanisms as well as a decoupled detection head to improve classification and regression tasks. Zhao et al. [8] present SCDNet that includes a dedicated scene classification subnetwork and a context-guided fusion module to improve the representations of tiny objects while suppressing the background information. Xie et al. [24] propose a contextual dependence mining network that constructs features with different receptive fields by stacking convolutional operations to individual layers of the feature pyramid network to capture varying contextual dependencies of objects and aggregates these features into a contextual representation that is used for both classification and regression. These methods share a common theme of deriving the contextual information from the same backbone used for extracting the object features, and integrating these contextual representations into the later detection stages using extra modules. Our proposed approach aims to capture the context via an inpainting based generative mechanism that learns short- and long-range dependencies among different parts of the image independently from the object detection architecture.

c) Image inpainting: Image inpainting models have the potential to capture the image context when they are tasked to generate the contents of an image region conditioned on its surroundings [9]. However, the main focus of image inpainting studies in remote sensing is to address object removal and filling of occluded areas with applications such as cloud and shadow removal, elimination of artifacts, and anonymization of sensitive locations. For example, Khan et al. [25] introduce a spatiotemporal inpainting approach that fills missing surface reflectance information by using the data available from nearby temporal instances. Dong et al. [26] train a deep convolutional generative adversarial network to learn uncorrupted distributions of sea surface temperature from historical images and use this generative model to reconstruct the regions occluded by clouds. Similarly, Sun et al. [27] propose a two-stage generative network for removing clouds from optical remote sensing images by combining a recurrent convolution network for cloud mask generation with an autoencoder employing partial convolutions for cloud removal. Du et al. [28] describe a coarse-to-fine deep generative model with spatial semantic attention to enhance local continuity and global semantic relevance to improve the inpainting performance for missing regions in high-resolution remote sensing images. Sha et al. [29] tackle missing data reconstruction from a single source image by designing a mask extraction network that obtains versatile soft masks of missing regions and an inpainting network that includes dilated pyramidal convolutions and an attention fusion mechanism. Zhang et al. [30] describe a masked image modeling approach that focuses on learning representative features from masked image reconstruction by treating original image patches as reconstructive templates and using a Siamese network to impose context consistency constraints during reconstruction. These inpainting studies apply common use cases such as occlusion or artifact reduction, and neither one exploits inpainting's contextual learning potential to improve the effectiveness of object detection.

III. Methodology

The main goal of the proposed methodology is to improve both the detection (localization) and the classification (labeling) performance of the object detection model by injecting the contextual information into the feature representation pipeline. Our approach builds on three key aspects: obtaining multi-scale object features using a pyramid network, extracting contextual information via image inpainting, and fusing them into the representation backbone given as input to the proposal generation as well as classification and regression branches. The proposed architecture for CODI is given in Figure 2. The following sections describe the oriented object detector, the inpainting model, the fusion module, and the overall training strategy.

A. Oriented object detector

Well-known one-stage and two-stage detectors such as YOLO [2] and Faster-RCNN [1] have also been applied to remote sensing imagery. Despite improvements, one-stage models face challenges such as localization in complex environments [31] and detecting small objects [32]. Therefore, we choose two-stage detection as our baseline approach. In particular, we use the oriented R-CNN [11] architecture that has been one of the best performing models among two-stage detectors while providing competitive efficiency with respect

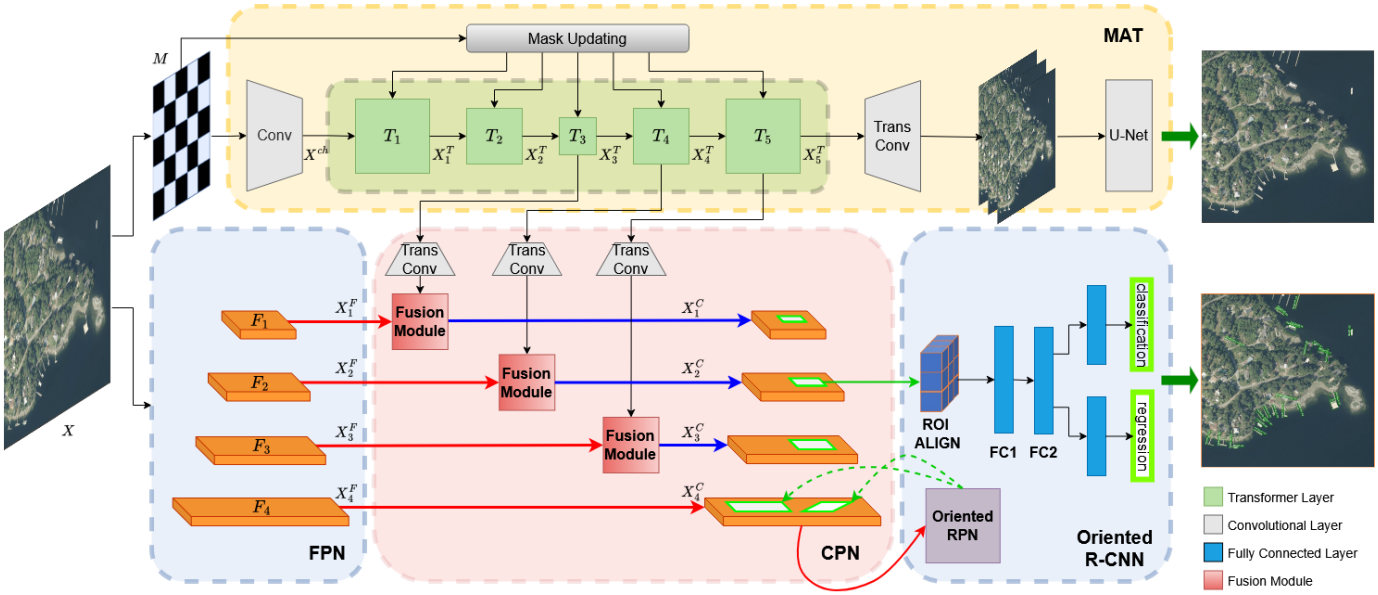


Fig. 2: Overall architecture for the proposed Contextual Object Detection via Inpainting (CODI) model. The parts highlighted in blue correspond to the feature pyramid network (FPN) for obtaining multi-scale object features as well as the oriented region proposal network (RPN) and classification and regression heads that are part of the oriented R-CNN object detector as described in Section III-A. The part highlighted in yellow corresponds to the mask-aware transformer (MAT) architecture used for inpainting-based contextual information extraction as described in Section III-B. The part highlighted in red illustrates the proposed fusion module that combines the object-specific local feature representations encoded in the FPN levels with the contextual feature representations captured by the transformer layers (highlighted in green) to produce the contextual pyramid network (CPN) that is used for both object localization and classification stages as described in Section III-C.

to one-stage detectors. Its oriented region proposal generation and rotated alignment-based feature extraction strategy makes it particularly effective in the detection of arbitrary-oriented objects in remote sensing images.

Oriented R-CNN uses a feature pyramid network (FPN) [33] backbone to produce the feature map that is given as input to an oriented region proposal network (RPN) for the proposal generation stage which is followed by the proposal classification and regression stage. These stages are illustrated in the bottom part of Figure 2. During inference, if a ground truth object does not have any overlap with a selected proposal or a proposal has a large location offset that leads to noisy features, the detection performance deteriorates with false negatives or mis-classifications, respectively. Similar to other detection networks, oriented R-CNN uses a class probability threshold that affects the final number of output detections. Increasing this threshold leads to fewer detections and potentially increased precision, and decreasing it leads to more candidate detections and potentially increased recall.

Our aim in this work is to enhance the feature representation using contextual information for a more precise control and improvement on both precision and recall. In the oriented R-CNN framework, FPN offers a multi-scale representation that supports generating object proposals and refining the resulting detections. This capability enables accurate identification of objects with various sizes and orientations. While there is information flow between the layers of the network through residual connections, oriented R-CNN particularly uses four of these levels as input to the classification and regression stages. We denote these levels as F_1, F_2, F_3, and F_4 from top to bottom with the upper layers corresponding to higher-level semantics and larger objects due to their larger receptive fields while the lower layers capture finer details and are particularly useful for smaller objects. Given the input image $X \in \mathbb{R}^{3\times H\times W}$ where H and W are image height and width in number of pixels, respectively, the representation at each FPN level $F_i, i = 1,\ldots,4$ is $X_i^F \in \mathbb{R}^{256\times H_i^F\times W_i^F}$ where $H_1^F \times W_1^F = H/32 \times W/32$, $H_2^F \times W_2^F = H/16 \times W/16$, $H_3^F \times W_3^F = H/8 \times W/8$, and $H_4^F \times W_4^F = H/4 \times W/4$. Each level has 256 feature channels, and for an example image of 1024×1024 pixels, the four levels have sizes 32×32, 64×64, 128×128, and 256×256 pixels, respectively. These representations are used in the contextual feature fusion in Section III-C.

B. Inpainting model

We use the mask-aware transformer (MAT) architecture [10] for inpainting that utilizes the self-attention mechanism to model long-range dependencies among different parts of the image, helping the model effectively understand the relationships between both distant and neighboring regions. MAT is differentiated from other inpainting models that use only convolutional approaches with a natural restriction to look only at limited receptive fields to fill the masked regions.

The MAT components are specifically designed to process high-resolution images, making it suitable for analyzing remote sensing data with a lot of details corresponding to a wide range of object types appearing in many different scales. The

architecture consists of a convolutional head, a transformer body, a convolutional tail, and a reconstruction module. The convolutional head is used to extract down-scaled image patches as tokens. The transformer body contains five stages of transformer blocks at varying resolutions to model the long-range interactions among the tokens. The convolutional tail that employs transposed convolutions is used to upsample the output tokens back to the input resolution. Finally, the reconstruction module uses a convolutional U-net structure to refine the high-frequency details and produce the final output image. These components are illustrated from left to right in the top part of Figure 2. The original architecture also includes a style manipulation module that supports pluralistic generation to handle multiple plausible solutions by changing the weight normalization of the convolution layers with an additional noise input. Even though the style manipulation module is used as part of the full inpainting pipeline during training, it is omitted from Figure 2 for simplicity because it is not used in our inference phase.

An important design decision in adapting the MAT architecture as our contextual backbone is the choice for the input mask. The conventional inpainting scenario requires an input mask to explicitly identify image regions or objects that need to be filled. However, our setting has no explicit mask as the objects of interest in the image are unknown. Furthermore, unlike natural scenes where objects are few and large, remote sensing images often contain densely populated areas with many different and small objects where the commonly used random masks may lose important local details. Therefore, we employ a generic mask in the form of a chessboard to capture the general layout of the image scene. The chessboard structure consists of alternating square cells that correspond to valid (available) and invalid (masked) image regions. The controllable cell size in the chessboard grid is a hyperparameter that we investigate in the ablation study in Section IV-D.

The input to the MAT architecture is the input image $X \in \mathbb{R}^{3\times H\times W}$ and the binary mask $M \in \{0,1\}^{H\times W}$. The convolutional head produces the down-scaled image $X^{\text{ch}} \in \mathbb{R}^{180\times H/8\times W/8}$. Each transformer layer $T_i, i = 1,\ldots,5$, uses an attention module with shifted windows and dynamic mask updating. The attention module uses the swin transformer strategy [34] where non-local interactions are efficiently computed using only the valid tokens as determined by the mask at that layer. The swin transformer blocks use different sizes of shifted windows to capture different ranges of relationships between masked and unmasked image regions. The mask updating module automatically updates the mask at each layer so that all tokens in a window become valid after attention as long as there exists at least one valid token whereas they remain invalid if all tokens are invalid in the window.

The transformer layers follow an encoder-decoder design with the largest receptive field in the central layer. The token sequence in each layer is reshaped into the corresponding image size at that layer to construct the contextual feature maps that form the outputs of the transformer layers $X_i^T \in \mathbb{R}^{180\times H_i^T\times W_i^T}$ where $H_1^T \times W_1^T = H/8 \times W/8$, $H_2^T \times W_2^T = H/16 \times W/16$, $H_3^T \times W_3^T = H/32 \times W/32$, $H_4^T \times W_4^T = H/16 \times W/16$, and $H_5^T \times W_5^T = H/8 \times W/8$.

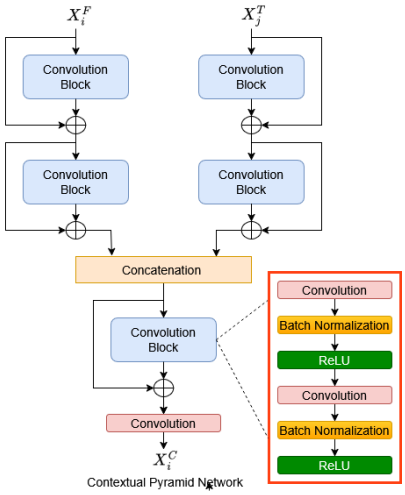


Fig. 3: Architecture of the contextual fusion module. X_i^F (top-left input) is the representation obtained from the FPN level F_i and X_j^T (top-right input) is the output of the matching transformer layer T_j after the transposed convolution. The resulting feature map X_i^C (bottom output) is used to construct the contextual pyramid network.

Each layer has 180 feature channels, and for an example image of 512×512 pixels, the five layer outputs have sizes 64×64, 32×32, 16×16, 32×32, and 64×64 pixels, respectively. The input images for MAT are down-scaled to half size for computational reasons, to keep the number of tokens at a reasonable level.

C. Contextual feature fusion

The goal of the fusion module is to effectively combine the object-specific local feature representations encoded in the FPN levels of the oriented object detector with the contextual feature representations captured by the transformer layers of the inpainting model. During fusion, we match the FPN levels with the transformer layers based on the compatibility of the resolutions of their feature maps. First, we apply transposed convolutions to the outputs of the transformer layers T_3, T_4, and T_5 so that their outputs are upsampled by a factor of 2. Then, the FPN levels F_1, F_2, and F_3 are paired with the transformer layers T_3, T_4, and T_5 with matching resolutions of $H/32 \times W/32$, $H/16 \times W/16$, and $H/8 \times W/8$ pixels, respectively. Since there is elementwise addition between T_1 and T_5 and between T_2 and T_4 in the inpainting model, all layers effectively contribute to the fusion process even though only three layers are directly paired with the FPN levels.

The proposed fusion module architecture is given in Figure 3. The input tensor pair (X_i^F, X_j^T) for the two branches in the fusion module correspond to the paired feature maps (X_1^F, X_3^T), (X_2^F, X_4^T), and (X_3^F, X_5^T). Our design contains sequences of convolution blocks for learning of latent spaces for effective fusion of information from both sources. Each convolution block includes two convolution layers with batch normalization and ReLU activation applied after each convolution. These convolutions do not modify the number of

channels in the input tensors to preserve the relative amount of contribution of each source before concatenation, as we hypothesize that the object-specific feature representations (with 256 channels in X^F) should have a higher weight in the fusion than the contextual feature representations (with 180 channels in X^T). For combination of the outputs of the two branches, we use channel-wise concatenation rather than element-wise addition to both handle this difference in the number of channels, and allow the following convolution block to learn an effective combination of the two individual sets of channels representing the learned latent spaces in the earlier branches. The final convolution layer maps the concatenated 436 channels back to 256 so that these feature maps form the contextual pyramid network (CPN) that can directly replace the FPN as the input to the oriented RPN in the oriented object detector. As shown in Figure 2, the resulting CPN consists of four levels constructed as follows: $X_1^C = \text{Fusion}(X_1^F, X_3^T)$, $X_2^C = \text{Fusion}(X_2^F, X_4^T)$, $X_3^C = \text{Fusion}(X_3^F, X_5^T)$, and $X_4^C = X_4^F$. The feature map X_4^F that corresponds to the lowest FPN level F_4 is used directly in the CPN as it captures the finest object-specific local feature representations.

The design in the fusion module adds the skip connection after the ReLU activation as opposed to the traditional residual approach [35] that applies the activation function after the skip connection. Let x be the input to the convolution block, $f(x)$ be the mapping learned until the ReLU activation, and y be the output resulting from adding the skip connection as

$$y = \text{ReLU}(f(x)) + x. \tag{1}$$

In the forward pass, this design ensures that only the activated features from $f(x)$ contribute to the output. Moreover, adding the original input x helps that the essential information is preserved and propagated to subsequent layers, balancing the complexity introduced by the nonlinear transformation. During backpropagation, the derivative of the loss function L (to be discussed later) with respect to the input x becomes

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y}\text{ReLU}'(f(x))f'(x) + \frac{\partial L}{\partial y}, \tag{2}$$

where the gradient flows directly through the identity path $\partial L/\partial y$ regardless of the activation state of $f(x)$. In contrast, the traditional residual block [35] that is defined as

$$y = \text{ReLU}(f(x) + x) \tag{3}$$

yields the derivative

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y}\text{ReLU}'(f(x) + x)(f'(x) + 1) \tag{4}$$

where the gradient passing through the skip connection is adjusted by the derivative of the ReLU activation. The design in (1) that is used in our fusion module provides an unaltered gradient path through the residual connection and helps mitigating vanishing gradient issues. Thus, the derivative in (2) facilitates more stable training as opposed to (4) that has no backpropagation when the ReLU activation is 0.

He et al. [36] presents a comprehensive study regarding the placement and ordering of different blocks (e.g., convolution, batch normalization, ReLU activation) in the design of skip connections in residual models. They conclude that, among different alternatives, moving the ReLU before the addition leads to better convergence in the optimization process and lower training errors compared to the original version in [35]. Our observations regarding the improvements in training and overall performance are consistent with those in [36].

D. Training strategy

The loss function that is used for the oriented object detector [11] considers both the class labels and the bounding boxes for the predicted objects. First, each object anchor is assigned a binary label where the anchor is labeled positive if its intersection over union (IoU) with any ground truth box exceeds 0.7, or if it has the highest IoU among all anchors when its IoU with a matched ground truth box is between 0.3 and 0.7. Anchors with IoU scores below 0.3 are labeled negative and are ignored in the training process. As in [11], we adopt a cross-entropy term $\mathcal{L}_{\text{cls}}$ for classification over all sampled anchors and a smooth L_1 term $\mathcal{L}_{\text{reg}}$ for refining the bounding boxes, applied only to positive anchors. The overall loss for the oriented RPN in the detector is given by

$$\mathcal{L}_{\text{det}} = \frac{1}{M}\sum_{i=1}^{M}\mathcal{L}_{\text{cls}}(p_i, p_i^*) + \frac{p_i^*}{M}\sum_{i=1}^{M}\mathcal{L}_{\text{reg}}(b_i, b_i^*) \tag{5}$$

where M denotes the number of sampled anchors, p_i represents the predicted class label for the i'th anchor, p_i^* is the ground truth label for that anchor, b_i is the set of predicted offsets of the oriented bounding box, and b_i^* corresponds to the ground truth offsets that enable the network to learn accurate rotation and shape adjustments.

The inpainting model employs a versatile loss framework that includes an adversarial loss, R_1 regularization, and a perceptual loss. The adversarial loss $\mathcal{L}_{\text{g}}$ has two parts: a generator loss that aims to produce realistic images that fool the discriminator, and a discriminator loss that distinguishes between real and generated images. The R_1 regularization penalizes the discriminator's gradient magnitude relative to its input, stabilizing the training by preventing overconfidence. The perceptual loss $\mathcal{L}_{\text{p}}$ ensures that the generated images align with the target images not only in pixel intensity but also in texture, style, and contextual semantics, ultimately enhancing the visual coherence of the inpainted regions. Finally, the overall loss for the inpainting model is defined as

$$\mathcal{L}_{\text{inp}} = \mathcal{L}_{\text{g}} + \gamma R_1 + \lambda\mathcal{L}_{\text{p}} \tag{6}$$

where the coefficients γ (set as 10) and λ (set as 0.1) adjust the contributions of R_1 regularization and perceptual loss, respectively.

As the overall training strategy, first, the inpainting model is trained using the loss function $\mathcal{L}_{\text{inp}}$ in (6). Then, this model is used in the inference mode while the parameters of the object detector and the fusion module are learned using the loss function $\mathcal{L}_{\text{det}}$ in (5).

IV. Experiments and Results

This section presents the datasets used in performance evaluation, implementation details, experimental results, ablation studies, and discussion of experimental findings.

A. Dataset

We use the DOTA [37], HRSC2016 [38], and DIOR-R [39] datasets that are widely recognized benchmarks supporting oriented annotations for remote sensing object detection. The DOTA dataset provides multi-class high-quality annotations for the following 15 object categories: plane (PL), baseball diamond (BD), bridge (BR), ground track field (GTF), small vehicle (SV), large vehicle (LV), ship (SH), tennis court (TC), basketball court (BC), storage tank (ST), soccer ball field (SBF), roundabout (RA), harbor (HA), swimming pool (SP), and helicopter (HC). It has 188,282 object instances with rotated bounding box annotations in 2806 images with sizes around 4000×4000 pixels. Consistent with [11], we split the images into 1024×1024 pixel tiles with 200-pixel strides for single-scale experiments. Similarly, we resize the original images at three scales (0.5, 1.0, and 1.5) and split them into 1024×1024 pixel tiles with 500-pixel strides for multi-scale experiments. The oriented object detector takes the 1024×1024 pixel images as input during both training and inference. The inpainting model uses resized images of 512×512 pixels due to computational reasons. We present both single- and multi-scale experiments for the DOTA dataset.

The HRSC2016 dataset provides challenging maritime scenes, including coastal areas, harbors, and open seas. The rotated bounding box annotations in 1061 images capture ship orientations and scale variations. Also consistent with [11], we resize the shorter sides of the images to 800 pixels while the longer sides become less than or equal to 1333 pixels without any splitting. The inpainting model also uses resized images due to computational reasons. All HRSC2016 images are processed at a single scale.

The DIOR-R dataset is a large-scale oriented object detection benchmark. It provides rotated bounding box annotations for 23,463 images and 192,518 object instances across 20 categories: airplane (APL), airport (APO), baseball field (BF), basketball court (BC), bridge (BR), chimney (CH), dam (DAM), expressway service area (ESA), expressway toll station (ETS), golf field (GF), ground track field (GTF), harbor (HA), overpass (OP), ship (SH), stadium (STA), storage tank (STO), tennis court (TC), train station (TS), vehicle (VE), and windmill (WM). All images have a fixed size of 800×800 pixels. The dataset provides 5862 images for training, 5863 for validation, and 11,738 for testing. All DIOR-R images are processed at a single scale without any splitting.

B. Implementation details

For all datasets, we use horizontal and vertical flipping as well as color normalization for data augmentation. For the chessboard mask used as input to the inpainting model, we use 32×32 pixel square grid cells that alternate as valid and invalid image regions in our default setting. For the DOTA example, this corresponds to 16 squares along both horizontal and vertical axes for an input image of 512×512 pixels. We train the inpainting model using only the DOTA dataset. The experiments for the HRSC2016 and DIOR-R datasets use this model to further demonstrate the generalizability of the approach. We set the class probability threshold to 0.01 for all datasets. Both the mask grid size and the probability threshold are further evaluated in the ablation experiments in Section IV-D.

We use the authors' code provided for the oriented R-CNN detector [11] as the baseline model. The code is used to train the model on our server with its original settings. All experiments are performed by using a single RTX 4090 GPU accompanied with 24 GB RAM. We use a ResNet-50 backbone to allow faster training and inference with a convenient batch size.

During the training process for all datasets, we utilize stochastic gradient descent with a momentum parameter of 0.9, an initial learning rate of 0.02, and a weight decay coefficient of 0.00013. To maintain numerical stability, we incorporate gradient clipping with a maximum L2 norm of 35. A stepwise learning rate schedule is implemented, reducing the learning rate by a factor of 10 at epochs 12 and 16 for DOTA and DIOR-R, and at epochs 36 and 48 for HRSC2016. Additionally, a linear warm-up is employed over 500 iterations, commencing at 0.001 of the initial learning rate to mitigate early gradient explosions. The models are trained for 18 epochs on DOTA and DIOR-R and 54 epochs on HRSC2016, using a batch size of 8. For the inpainting training stage, we employ a learning rate of 0.001 with a reduced batch size of 4.

C. Results

We employ mean average precision (mAP) as the performance metric used in comparative experiments. We use both the sampling-based interpolated average precision computed from the precision-recall curve (PASCAL VOC 2007 definition [40]) and the version that uses all data points on the curve (PASCAL VOC 2012 definition [41]). In particular, DOTA's evaluation server that implements the PASCAL VOC 2012 definition is used for the DOTA test results. Consistent with the literature, we use both PASCAL VOC 2007 and 2012 definitions for the HRSC2016 experiments.

The first set of experiments compares CODI with well-known one-stage and two-stage object detection methods on the DOTA dataset. Table I presents both class-specific and overall mAP scores for all methods. We present the results corresponding to a ResNet-50 backbone and multi-scale processing whenever applicable for fair comparison. We also present the results for both the baseline model (oriented R-CNN) and CODI using both single-scale and multi-scale data preparation settings. The results show that the proposed model outperforms the baseline model for both single-scale (with an mAP of 71.24% versus 70.57%) and multi-scale (with an mAP of 76.61% versus 76.01%) experiments according to the scores obtained from the official DOTA evaluation servers. The improvements for both settings show the effectiveness of the proposed integration of contextual information in the detection process. CODI also obtains the highest performance compared to all of the other detection methods in Table I.

When individual classes are considered, the results confirm that our model delivers its most significant gains over the baseline for both single- and multi-scale settings for the classes

TABLE I: Comparison of the proposed model CODI with various one-stage and two-stage object detectors in terms of both classwise and overall mean average precision (mAP) scores on the DOTA dataset. Horizontal bounding box (HBB) results are given when oriented bounding box (OBB) results are not available. Both single-scale and multi-scale results are provided for both the baseline oriented R-CNN (O-RCNN) model and our model CODI. The methods are ordered according to overall mAP.

Method	BBox	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
One-stage																	
SSD [37]	HBB	41.06	24.31	4.55	17.10	15.93	7.72	13.21	39.96	12.05	46.88	9.09	30.82	1.36	3.50	0.00	17.84
YOLO_v2 [37]	HBB	52.75	24.24	10.60	35.50	14.36	2.41	7.37	51.79	43.98	31.35	22.30	36.68	14.61	22.55	11.89	25.49
PIoU [42]	OBB	80.90	69.70	24.10	60.20	38.30	64.40	64.80	90.90	77.20	70.40	46.50	37.10	57.10	61.90	64.00	60.50
RetinaNet-0 [11]	OBB	88.67	77.62	41.81	58.17	74.58	71.64	79.11	90.29	82.18	74.32	54.75	60.60	62.57	69.67	60.64	68.43
DRN [43]	OBB	89.71	82.34	47.22	64.10	76.22	74.43	85.84	90.57	86.18	84.89	57.65	61.93	69.30	69.63	58.48	73.23
M2FE-YOLO [44]	OBB	94.40	77.50	55.70	49.20	69.60	83.80	90.2	97.50	70.60	84.10	69.50	47.60	88.00	86.00	45.40	73.90
DHRec [45]	OBB	88.58	77.90	53.84	72.93	78.45	78.84	87.64	90.88	88.78	85.46	56.11	66.74	67.58	70.25	57.53	74.57
SASM [46]	OBB	86.42	78.97	52.47	69.84	77.30	75.99	86.72	90.89	82.63	85.66	60.13	68.25	73.98	72.22	62.37	74.92
Two-stage																	
Faster RCNN [37]	HBB	79.42	77.13	17.70	64.05	35.30	38.02	37.16	89.41	69.64	59.28	50.30	52.91	47.89	47.40	46.30	54.13
Faster R-CNN-0 [11]	OBB	88.44	73.06	44.86	59.09	73.25	71.49	77.11	90.84	78.94	83.90	48.59	62.95	62.18	64.91	56.18	69.05
Faster R-CNN-0+OAN [15]	OBB	88.44	76.33	46.31	59.70	73.30	72.13	77.90	90.72	79.02	81.60	44.80	58.66	61.28	67.51	62.87	69.37
RoI Transformer [47]	OBB	88.64	78.52	43.44	75.92	68.81	73.68	83.59	90.74	77.27	81.46	58.39	53.54	62.83	58.93	47.67	69.56
CAD-Net [5]	OBB	87.80	82.40	49.40	73.50	71.10	63.50	76.70	90.90	79.20	73.30	48.40	60.90	62.00	67.00	62.20	69.90
ICN [48]	OBB	89.97	77.71	53.38	73.26	73.46	65.02	78.22	90.79	79.05	84.81	57.20	62.11	73.45	70.22	58.08	72.45
EAMSDet [49]	OBB	88.86	77.19	51.82	63.84	80.25	76.71	87.20	90.90	84.72	85.23	65.31	65.07	66.72	70.65	54.96	73.96
ARS-DETR [23]	OBB	86.97	75.56	48.32	69.20	77.92	77.94	87.69	90.50	77.31	82.86	60.28	64.58	74.88	71.76	66.62	74.16
DFDet [24]	OBB	88.92	79.25	48.40	70.00	80.22	78.85	87.21	90.90	83.13	83.98	60.07	66.49	68.27	76.78	58.11	74.71
OrientedFormer [22]	OBB	88.14	79.13	51.96	67.34	81.02	83.26	88.29	90.90	85.57	86.25	60.84	66.36	73.81	71.23	56.49	75.37
OSKDet [50]	OBB	89.98	86.99	53.13	75.55	72.87	76.97	87.63	90.74	78.87	86.97	60.13	70.68	75.70	71.53	65.60	76.22
COBB [51]	OBB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	76.53
O-RCNN [11] (single)	OBB	89.28	74.49	53.47	71.75	63.67	81.39	86.24	90.88	82.95	84.17	56.35	42.00	67.09	62.00	52.83	70.57
Our model (single)	OBB	89.19	75. 69	54.78	72.54	65.86	81.63	85.70	90.89	83.93	84.11	57.65	41.11	72.47	62.77	50.14	71.24
O-RCNN [11] (multi)	OBB	89.86	82.98	59.65	77.96	61.50	83.67	87.25	90.86	85.83	87.12	71.39	43.09	78.37	69.97	70.63	76.01
Our model (multi)	OBB	89.97	83.94	60.63	79.07	58.79	83.41	87.42	90.84	87.00	87.62	69.09	47.29	79.13	69.63	75.29	76.61

whose identities are reinforced by distinctive environments. Harbors, for example, are uniquely framed by docks, water, and quaysides, bridges are signaled by a road span over a river or valley, and sports venues such as baseball diamonds, ground track fields, and basketball courts present characteristic line markings, grass boundaries, and surrounding stands and open areas. When CODI can attend to these characteristic scene layouts, detection accuracy increases remarkably relative to the baseline. Conversely, classes whose instances appear in isolation or in visually cluttered backgrounds, such as small and large vehicles, planes, and storage tanks, derive limited benefit and, under multi-scale inference, may even lose accuracy over the baseline because the added context has the risk of diluting discriminative object cues and inflating false positives with respect to the class probability threshold.

We also compare our model against the oriented R-CNN baseline using two complementary statistical analyses on the DOTA dataset. Using the test set performance presented in Table I, we perform paired comparisons of per-class average precision values for 15 object categories, both for single-scale and multi-scale settings. In both cases, the mean difference in average precision is positive (approximately +0.66 AP for single-scale and +0.60 AP for multi-scale), indicating that, on average, our method consistently improves over the baseline across object categories. To obtain a more robust assessment, we also conduct a non-parametric bootstrap analysis on the validation set (containing 5297 images in the single-scale split), where ground truth annotations are available and statistics can be computed over thousands of images. Using 500 bootstrap repetitions at the image level with each repetition consisting of 5297 samples with replacement, we obtain a mean improvement of 0.97% in mAP in favor of our method. Importantly, the 95% confidence interval for this improvement, $[0.27, 1.76]$, lies entirely above zero, and the probability that the improvement is non-positive is only 0.2% (two-sided $p = 0.004 < 0.05$). This provides strong evidence that the observed performance gains over oriented R-CNN are statistically significant.

TABLE II: Comparison of the proposed model CODI with various object detectors with respect to mean average precision (mAP) (both PASCAL VOC 2007 and 2012 versions) on the HRSC2016 dataset. Information on the backbone architecture used is provided. The methods are ordered according to mAP VOC 2007 version and the best results are highlighted in bold.

Method	Backbone	mAP(07)	mAP(12)
Rotated RPN [52]	ResNet-101-FPN	79.08	85.64
RoI Transformer [47]	ResNet-101-FPN	86.20	-
Gliding Vertex [53]	ResNet-101-FPN	88.20	-
PIoU [42]	DLA-34	89.20	-
R3Det [54]	ResNet-101-FPN	89.26	96.01
DAL [55]	ResNet-101-FPN	89.77	-
CenterMap-Net [56]	ResNet-50-FPN	89.83	92.10
S^2ANet [57]	ResNet-101-FPN	90.17	95.01
OrientedFormer [22]	ResNet-50-FPN	90.17	96.48
DETR-ORD [58]	ResNet-50-FPN	90.21	96.80
DFDet [24]	ResNet-50-FPN	90.25	96.51
O-RCNN [11]	ResNet-50-FPN	90.30	96.52
AOPG [59]	ResNet-50-FPN	90.34	96.22
OASL [60]	ResNet-50-FPN	90.36	-
Oriented DETR [21]	ResNet-50-FPN	90.52	**97.73**
Our model	ResNet-50-FPN	**90.57**	96.73

The second set of experiments compares CODI with other object detection methods on the HRSC2016 dataset. Table II presents the mAP scores for all methods. The methods are ordered according to mAP VOC 2007 version as all methods used for comparison provide the scores for that version. We use the same hyperparameters as those used for the DOTA dataset. We also use the inpainting model trained on the DOTA dataset for these experiments. CODI outperforms the oriented R-CNN baseline when using a ResNet-50 backbone with mAP scores of 90.57% and 96.73% according to the PASCAL VOC 2007 and 2012 version of the performance metric, respectively. When all methods are considered, CODI achieves the best score under the 2007 metric. It also has the third highest score with a very competitive performance based on the 2012 metric among all methods with scores available in Table II.

The third set of experiments compares CODI with other object detection methods on the DIOR-R dataset. Table III presents the mAP scores for all methods. Both our model and the oriented R-CNN baseline are trained on the union of the training and validation sets, and evaluation is done on the test set. We use the same hyperparameters as those used for the DOTA dataset. We also use the inpainting model trained on the DOTA dataset for these experiments. CODI outperforms the oriented R-CNN baseline with an overall mAP difference of 1.97%. When individual classes are considered, CODI has a higher score than oriented R-CNN for 16 of the 20 categories. Consistent with the DOTA results, more significant gains are obtained for classes, such as dam, golf field, harbor, and train station, with characteristic scene layouts. CODI also obtains the highest performance compared to all of the other detection methods in Table III.

For qualitative evaluation and explainability of how fusion of contextual information captured by the transformer layers in the inpainting model enhances the object representations in the feature pyramid network, we visualize the corresponding feature and attention maps in Figure 4. These maps illustrate the parts of the scene that the model focuses on using different receptive fields. In particular, the upper layers correspond to higher-level semantics and larger objects whereas the lower layers capture smaller details in the scene. We observe that the FPN feature maps and the transformer attention maps in the same level have compatible resolutions in the corresponding heatmaps, confirming our decision for pairing these particular maps in the fusion process. When the FPN levels of the baseline model are considered, the focus moves from the taxiway on the upper levels to individual airplane details on the lower levels. When the FPN levels of the proposed model are considered, we can already see improved contrast due to the effect of backpropagation during learning the whole contextual model. Furthermore, when the attention maps of the inpainting model are considered, the focus moves from larger taxiway and apron regions on the upper layers to small details on the lower layers. Finally, when the fused levels in the contextual pyramid network are considered, we see more consistent activations and better localization of the taxiway and surrounding grass regions that correspond to the general context of the scene environment on the upper levels to rows of airplanes, taxiway markings, and individual object details

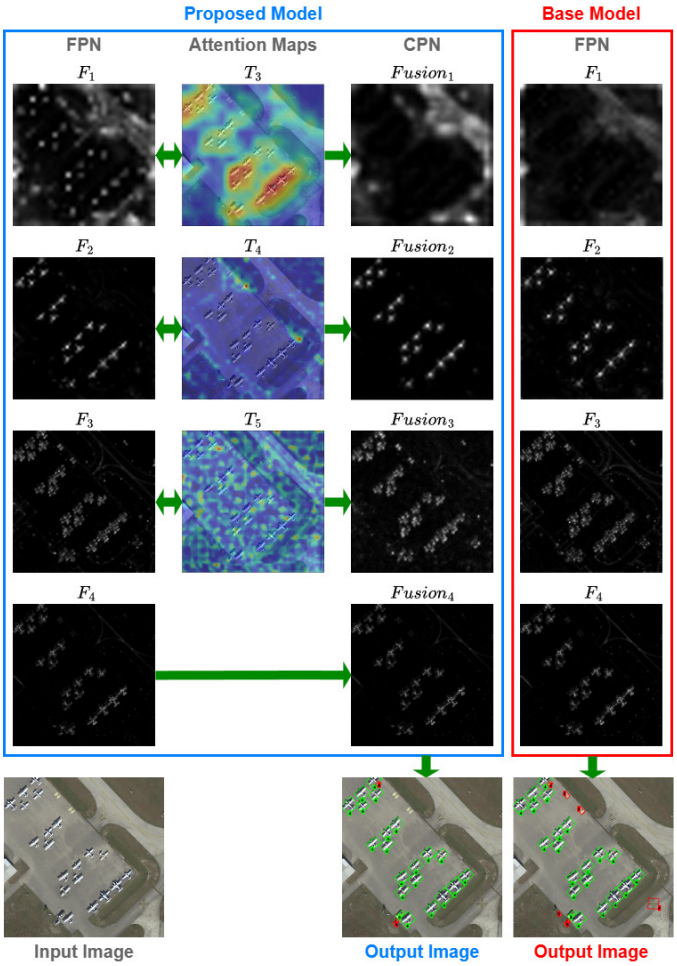


Fig. 4: Visualization of the feature pyramid network (FPN) levels of the oriented object detector and the attention maps of the transformer layers of the inpainting model as well as the proposed fusion result as the contextual pyramid network (CPN) levels. The FPN levels of the baseline oriented R-CNN model are also shown.

on the lower levels. These enhanced representations lead to improved precision in the final output image compared to that of the baseline model. Through finding an effective balance among both within-level and between-level relationships of the scene content in the contextual pyramid network, the proposed model improves both precision and recall with better object localization in the detection process.

We perform an additional experiment regarding scene-level prediction analysis to study how effectively our model captures the scene context. Since the images in the DOTA dataset do not have any scene label, we use the ground truth object class distribution in each image as a feature to cluster the images into different types of scenes. Figure 5 shows the cluster means corresponding to the normalized object frequency histograms when the images are grouped into 9 clusters by the k-means algorithm. The number of clusters is chosen empirically considering different values from 5 to 15. After obtaining the scene clusters, we analyze the individual object instances detected within each image. If a false positive object instance has a class

TABLE III: Comparison of the proposed model CODI with various object detectors in terms of both classwise and overall mean average precision (mAP) scores on the DIOR-R dataset. The methods are ordered according to overall mAP.

Method	Backbone	APL	APO	BF	BC	BR	CH	DAM	ESA	ETS	GF	GTF	HA	OP	SH	STA	STO	TC	TS	VE	WM	mAP
RetinaNet-O [3]	R-50-FPN	61.49	28.52	73.57	81.17	23.98	72.54	19.94	58.20	72.39	69.25	79.54	32.14	44.87	77.71	67.57	61.09	81.46	47.33	38.01	60.24	57.55
Faster RCNN-O [1]	R-50-FPN	62.79	26.80	71.72	80.91	34.20	72.57	18.95	65.75	66.45	66.63	79.24	34.95	48.79	81.14	64.34	71.21	81.44	47.31	50.46	65.21	59.54
FCOS-O [61]	R-50-FPN	54.01	40.04	71.76	80.99	34.81	72.37	26.40	77.59	67.19	68.76	75.38	34.10	51.16	80.44	58.11	61.57	81.49	52.71	42.07	64.95	59.80
Gliding Vertex [53]	R-50-FPN	65.35	28.87	74.96	81.33	33.88	74.31	19.58	64.70	70.72	72.30	78.68	37.22	49.64	80.22	69.26	61.13	81.49	44.76	47.71	65.04	60.06
DFDet [24]	R-50-FPN	61.92	38.83	77.41	81.36	34.11	74.97	26.26	76.06	62.31	75.56	79.62	38.26	52.76	80.40	73.11	68.27	81.38	52.23	44.11	63.35	62.11
RoI Transformer [47]	R-50-FPN	63.34	37.88	71.78	87.53	40.68	72.60	26.86	68.09	78.71	68.96	82.74	47.71	55.61	81.21	78.23	70.26	81.61	54.86	43.27	65.52	63.87
AOPG [59]	R-50-FPN	62.39	37.79	71.62	87.63	40.90	72.47	31.08	77.99	65.42	73.20	81.94	42.32	54.45	81.17	72.69	71.31	81.49	60.04	52.38	69.99	64.41
OriMamba [62]	VMamba-T	72.08	35.74	80.55	81.30	36.73	72.61	32.62	79.65	65.48	78.11	84.06	43.65	50.04	72.25	81.61	70.61	81.55	64.70	41.58	65.73	64.53
DOdet [63]	R-50-FPN	63.40	43.35	72.11	81.32	43.12	72.59	33.32	70.84	78.77	74.15	75.47	48.00	59.31	85.41	74.04	71.56	81.52	55.47	51.86	66.40	65.10
O-RCNN [11]	R-50-FPN	65.38	37.81	78.14	88.83	43.21	77.67	25.68	84.51	67.81	74.26	85.42	41.17	58.39	87.40	75.31	67.12	88.54	55.11	45.14	68.37	65.76
EOOD [64]	R-50-FPN	66.44	39.67	73.19	87.51	43.25	77.45	35.33	79.46	70.62	77.38	76.42	45.49	56.30	87.47	62.97	71.81	85.20	56.22	53.75	70.04	65.80
ARS-DETR [23]	R-50-FPN	68.00	54.17	74.43	81.65	41.13	75.66	34.89	81.92	73.07	76.10	78.62	36.33	55.41	84.55	70.09	72.23	81.14	61.52	50.57	70.28	66.12
AFDR-Det [65]	R-50-FPN	71.92	42.63	80.68	88.46	45.65	72.41	39.28	79.83	71.26	77.37	83.81	44.04	58.75	81.26	76.94	70.05	81.46	56.02	49.56	66.72	66.90
TAOD [66]	R-50-FPN	64.39	44.73	72.74	88.57	42.71	73.65	35.72	80.54	65.12	74.76	83.44	47.65	58.81	82.73	76.81	73.79	89.74	62.33	53.68	70.58	67.12
M-O YOLOX [67]	CSP-Darknet	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	67.20
CR-HLDet [17]	R-50-FPN	71.83	41.32	79.75	89.97	43.85	77.47	36.46	84.79	70.15	76.62	84.62	45.55	58.67	81.17	79.49	70.95	81.48	61.30	48.30	66.17	67.50
Our model	R-50-FPN	68.55	40.47	79.40	89.38	44.94	78.33	31.97	84.97	69.95	79.38	85.34	46.88	60.08	86.83	74.65	66.07	89.19	62.18	45.45	68.59	67.63

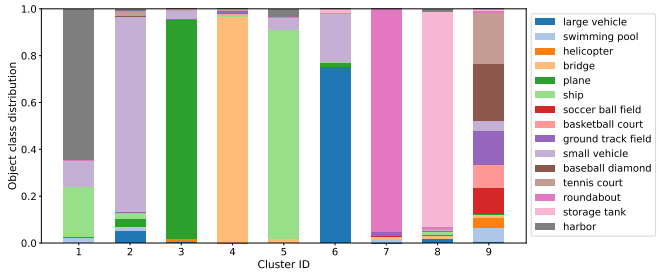


Fig. 5: Object class distributions for 9 clusters to study different scene types in the DOTA dataset. Different colors represent different object classes as shown in the legend.

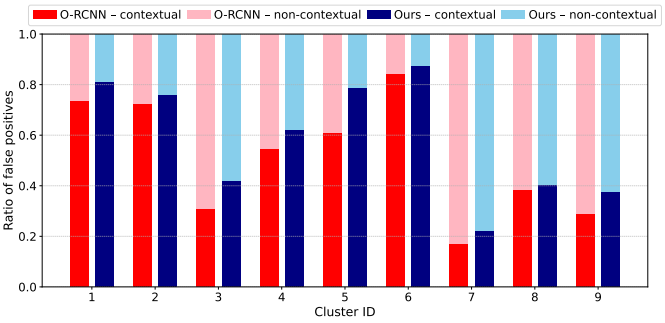


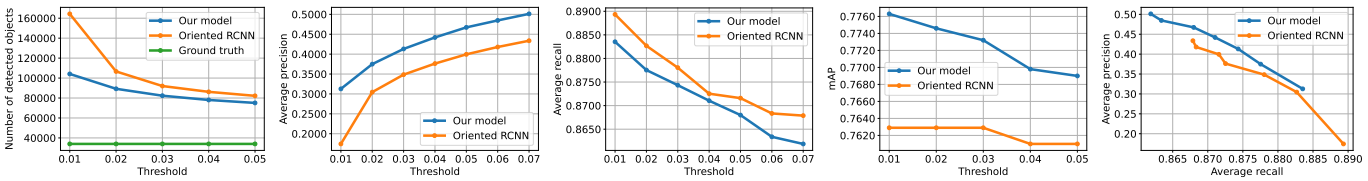
Fig. 6: Ratios of contextual and non-contextual detections among all false positive predictions for different scene types (illustrated in Figure 5) for both the oriented R-CNN baseline and the proposed model.

that is not among the ground truth classes in that image, that false positive instance is considered as out-of-context (non-contextual detection). Conversely, if the false positive object's class is among the ground truth annotations, it is considered as in-context (contextual detection). This analysis aims to identify how many false positive detections still have labels that are consistent with the ground truth object class annotations that exist in images belonging to different types of scenes. Figure 6 shows the ratios of contextual and non-contextual detections among all false positives for different scene types for both the proposed model and the oriented R-CNN baseline. Our model has better contextual predictions for every scene type. The highest improvements compared to the baseline model are obtained for the clusters 1, 3, 4, 5, and 9, corresponding to scenes with ships in harbor, airport, bridge, ships in open sea, and sports fields according to the distributions in Figure 5. All of these scene types have distinctive backgrounds that are exploited by contextual learning in the proposed model. Thus, even when our model produces false positives in such scenes, they are still consistent with the overall scene context. However, scenes with cluttered and non-distinctive background features do not have such significant performance difference between the proposed model and the baseline.

D. Ablation study

The ablation experiments use the training and validation subsets of the DOTA dataset for training and testing the proposed detection pipeline, respectively. We use the PASCAL VOC 2007 definition for mean average precision for performance evaluation. We also use individual precision and recall values across various class probability thresholds to thoroughly evaluate the differences in performance between our model and the baseline. An IoU threshold of 0.5 is used for both models.

The first set of ablation experiments evaluates the effect of the class probability threshold on different performance metrics during inference. Figure 7 shows the number of detected objects and the resulting precision, recall, and mAP values at different threshold levels for both the proposed model and the oriented R-CNN baseline. When the class probability threshold is decreased from the default value of 0.05 to 0.01, the baseline model produces significantly more predictions (an increase of 100%) that lead to an increase (from 0.8716 to 0.8893) in its recall performance. However, it also leads to a significant decrease (from 0.3995 to 0.1744) in precision because many of the newly detected objects are false positives. On the other hand, when the threshold is decreased from 0.05 to 0.01, the proposed model predicts 39% more objects with a similar increase in recall (from 0.8680 to 0.8835) but with a much smaller decrease (from 0.4672 to 0.3127) in precision. In particular, the recall score difference between the baseline and our model is 0.005 or less in favor of the former when the threshold is smaller than 0.05, but the difference in precision increases to almost 0.14 to our model's advantage



(a) (b) (c) (d) (e)

Fig. 7: Effect of the class probability threshold on the number of detected objects and the resulting performance metrics (precision, recall, and mAP) for the proposed model (blue) and the oriented R-CNN baseline (orange) on the DOTA dataset. (a) Number of detected objects vs threshold. (b) Average precision vs threshold. (c) Average recall vs threshold. (d) Mean average precision vs threshold. (e) Average precision vs average recall.

in the same setting. Furthermore, the proposed model has a higher mAP score than that of the baseline model for all threshold values, and achieves a higher precision score for all recall levels. These experiments show that the proposed fusion of the contextual information captured by the inpainting model with the object embeddings computed by the feature pyramid network leads to more selective and precise object representations that help the second stage of the object detector to identify the objects in the scene more accurately.

Tables IV, V, and VI present the details of the classwise precision, recall, and mAP values at different thresholds, respectively. Lowering the threshold reveals a notable difference between the proposed model and the baseline, particularly for classes such as plane, ship, storage tank, and harbor that appear in relatively more consistent contexts compared to other classes. With its effective utilization of this contextual information, the proposed model achieves additional true detections towards an increased recall when the threshold is relaxed. Although this relaxation slightly reduces precision, the proposed model's precision and mAP scores remain high compared to the baseline. In contrast, under identical conditions, the oriented R-CNN baseline admits a significantly higher number of false positives, resulting in a sharp deterioration in precision despite an increase in recall. For example, for classes such as helicopter, roundabout, basketball court, soccer ball field, and bridge that are more rare compared to others, the proposed model achieves a better precision-recall combination compared to the baseline at higher threshold levels. Consequently, lowering the threshold further detects a few additional object instances and increases recall with only a moderate decrease in precision. However, the baseline model cannot achieve a similar precision-recall balance for such classes.

Qualitative illustration of the detection performance and the improvement in both precision and recall for the proposed model are presented in Figures 8, 9, 10, and 11. In Figure 8, decreasing the threshold from 0.05 to 0.01 results in a similar recall performance with many more false positives that lead to loss in precision for the baseline model. However, the proposed model detects more objects with minimal false positives, resulting in an improvement in recall at a very similar level of precision. In Figure 9, the recall performances of both models are similar at a threshold of 0.01 but the proposed model achieves higher precision with fewer false positives. Finally, in Figures 10 and 11, we see that the proposed model achieves higher recall or precision while preserving precision or recall, respectively, whereas the baseline model has missing or false detections. Overall, by calibrating the class probability threshold, we can have effective control on enhancing the recall performance with minimal precision trade-offs while filtering low-scoring predictions prior to non-maximum suppression during inference.

The second set of ablation experiments evaluates the effect of the chessboard mask sizes within the inpainting model. The experiments presented earlier use 32×32 pixel grid cells in the default setting. We also consider 8×8 and 64×64 pixel cells as alternatives. When 8×8 cells are used, the inpainting model tries to fill in the masked image regions by looking at more local parts of the image. Even though this works well for small objects, it is ineffective for larger-scale targets. On the contrary, the inpainting model cannot capture the local details when the larger 64×64 cells are used for learning the context. This is consistent with the observations in [10] where the model cannot effectively capture the local details when larger masks are used. These observations are also supported by quantitative results obtained in experiments done on a smaller subset of the DOTA dataset. In these experiments, using 8×8 pixel, 32×32 pixel, and 64×64 pixel grid cells achieve 0.7422, 0.7482, and 0.7294 mAP scores, respectively. For 10 of the 15 object classes, the 32×32 pixel setting achieves the highest average precision, and this setting has the second highest score for the remaining 5 classes. Thus, the 32×32 pixel setting emerges as a good balance regarding the scale of the details used for capturing the relationships between the valid (available) and the invalid (masked) image regions.

The final set of ablation experiments illustrates how each transformer layer in the inpainting model contributes to fill in the missing regions. To emphasize the details, we employ a larger mask as shown in Figure 12. The output of each transformer block is illustrated by bypassing the remaining transformer blocks and applying only upsampling before the convolutional reconstruction tail. We observe that, after the convolution head with 3×3 partial convolutions fills the general background as part of the inpainting task, the transformer blocks fill in the masked regions in an increasingly detailed fashion.

TABLE IV: Comparison of the proposed model CODI with the baseline oriented R-CNN (O-RCNN) model in terms of classwise precision scores at various detection thresholds on the DOTA dataset.

Method	Threshold	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC
O-RCNN	0.01	0.2671	0.1405	0.0293	0.0784	0.2219	0.1473	0.4115	0.5099	0.0981	0.2108	0.0628	0.0912	0.1532	0.1802	0.0128
Our Model	0.01	0.7021	0.2284	0.0865	0.1271	0.2973	0.2130	0.6057	0.6490	0.1824	0.5311	0.0958	0.1746	0.3758	0.2723	0.1492
O-RCNN	0.02	0.7054	0.2218	0.0791	0.1166	0.2652	0.2364	0.6131	0.6252	0.1830	0.4782	0.0918	0.1306	0.3924	0.2827	0.1462
Our Model	0.02	0.7715	0.2985	0.1207	0.1796	0.3382	0.2790	0.6838	0.7302	0.2432	0.5925	0.1347	0.2415	0.4604	0.3363	0.2125
O-RCNN	0.03	0.7653	0.2719	0.0986	0.1438	0.3304	0.2799	0.6671	0.6707	0.2279	0.5218	0.1159	0.1591	0.4671	0.3302	0.1770
Our Model	0.03	0.8038	0.3448	0.1467	0.2159	0.3648	0.3228	0.7225	0.7683	0.2903	0.6261	0.1706	0.2800	0.5025	0.3795	0.2571
O-RCNN	0.04	0.7846	0.3088	0.1148	0.1650	0.3524	0.3108	0.6964	0.7028	0.2632	0.5522	0.1362	0.1835	0.5023	0.3648	0.2054
Our Model	0.04	0.8245	0.3877	0.1665	0.2443	0.3837	0.3570	0.7488	0.7895	0.3213	0.6569	0.1990	0.3156	0.5361	0.4099	0.2880
O-RCNN	0.05	0.8011	0.3351	0.1276	0.1876	0.3691	0.3369	0.7192	0.7257	0.3002	0.5809	0.1565	0.2013	0.5278	0.3869	0.2355
Our Model	0.05	0.8417	0.4244	0.1826	0.2683	0.3990	0.3838	0.7681	0.8112	0.3569	0.6749	0.2251	0.3449	0.5631	0.4426	0.3212

TABLE V: Comparison of the proposed model CODI with the baseline oriented R-CNN (O-RCNN) model in terms of classwise recall scores at various detection thresholds on the DOTA dataset.

Method	Threshold	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC
O-RCNN	0.01	0.9599	0.8846	0.7377	0.9023	0.9146	0.9532	0.9286	0.9610	0.8370	0.9306	0.9888	0.8824	0.8315	0.8618	0.7662
Our Model	0.01	0.9558	0.8798	0.7236	0.9022	0.9122	0.9421	0.9269	0.9569	0.8000	0.9131	0.9662	0.9058	0.8463	0.8812	0.7402
O-RCNN	0.02	0.9583	0.8798	0.7283	0.9023	0.9127	0.9482	0.9266	0.9597	0.8296	0.9274	0.9775	0.8706	0.8301	0.8618	0.7273
Our Model	0.02	0.9546	0.8653	0.7236	0.9022	0.9115	0.9384	0.9242	0.9569	0.8000	0.9104	0.9662	0.8823	0.8463	0.8790	0.7012
O-RCNN	0.03	0.9563	0.8798	0.7190	0.9023	0.9112	0.9445	0.9260	0.9583	0.8222	0.9243	0.9663	0.8706	0.8291	0.8596	0.7013
Our Model	0.03	0.9546	0.8605	0.7096	0.8947	0.9103	0.9338	0.9237	0.9543	0.8000	0.9094	0.9662	0.8764	0.8449	0.8747	0.7012
O-RCNN	0.04	0.9550	0.8702	0.7119	0.8947	0.9107	0.9412	0.9252	0.9570	0.8074	0.9206	0.9551	0.8647	0.8262	0.8596	0.6883
Our Model	0.04	0.9518	0.8557	0.7049	0.8947	0.9088	0.9284	0.9219	0.9529	0.7925	0.9067	0.9662	0.8764	0.8410	0.8747	0.6883
O-RCNN	0.05	0.9546	0.8702	0.7096	0.8947	0.9094	0.9369	0.9246	0.9570	0.8074	0.9195	0.9551	0.8647	0.8243	0.8575	0.6883
Our Model	0.05	0.9522	0.8509	0.6908	0.8796	0.9078	0.9251	0.9218	0.9529	0.7851	0.9062	0.9662	0.8764	0.8410	0.8747	0.6883

TABLE VI: Comparison of the proposed model CODI with the baseline oriented R-CNN (O-RCNN) model in terms of both classwise and overall mean average precision (mAP) scores at various detection thresholds on the DOTA dataset.

Method	Threshold	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
O-RCNN	0.01	0.9037	0.7333	0.4966	0.7435	0.7651	0.8346	0.8969	0.9083	0.7094	0.8778	0.7789	0.7245	0.7479	0.7392	0.5845	0.7629
Our Model	0.01	0.9023	0.7495	0.5304	0.7598	0.7701	0.8336	0.8971	0.9086	0.7038	0.8769	0.7929	0.7628	0.7577	0.7484	0.6506	0.7763
O-RCNN	0.02	0.9037	0.7333	0.4966	0.7435	0.7651	0.8346	0.8969	0.9083	0.7094	0.8778	0.7789	0.7245	0.7479	0.7392	0.5845	0.7629
Our Model	0.02	0.9023	0.7500	0.5300	0.7597	0.7705	0.8340	0.8969	0.9086	0.7035	0.8769	0.7919	0.7589	0.7582	0.7485	0.6497	0.7746
O-RCNN	0.03	0.9037	0.7333	0.4966	0.7435	0.7651	0.8346	0.8969	0.9083	0.7094	0.8778	0.7789	0.7245	0.7479	0.7392	0.5845	0.7629
Our Model	0.03	0.9022	0.7495	0.5309	0.7405	0.7698	0.8337	0.8970	0.9086	0.7023	0.8773	0.7908	0.7394	0.7578	0.7488	0.6496	0.7732
O-RCNN	0.04	0.9037	0.7333	0.4966	0.7302	0.7651	0.8346	0.8969	0.9083	0.7094	0.8778	0.7789	0.7245	0.7479	0.7392	0.5684	0.7610
Our Model	0.04	0.9023	0.7503	0.5303	0.7394	0.7704	0.8336	0.8971	0.9086	0.6767	0.8769	0.7912	0.7402	0.7567	0.7477	0.6255	0.7698
O-RCNN	0.05	0.9037	0.7333	0.4966	0.7302	0.7651	0.8346	0.8969	0.9083	0.7094	0.8778	0.7789	0.7245	0.7479	0.7392	0.5684	0.7610
Our Model	0.05	0.9023	0.7500	0.5147	0.7389	0.7704	0.8336	0.8970	0.9086	0.6741	0.8770	0.7910	0.7441	0.7553	0.7493	0.6270	0.7690

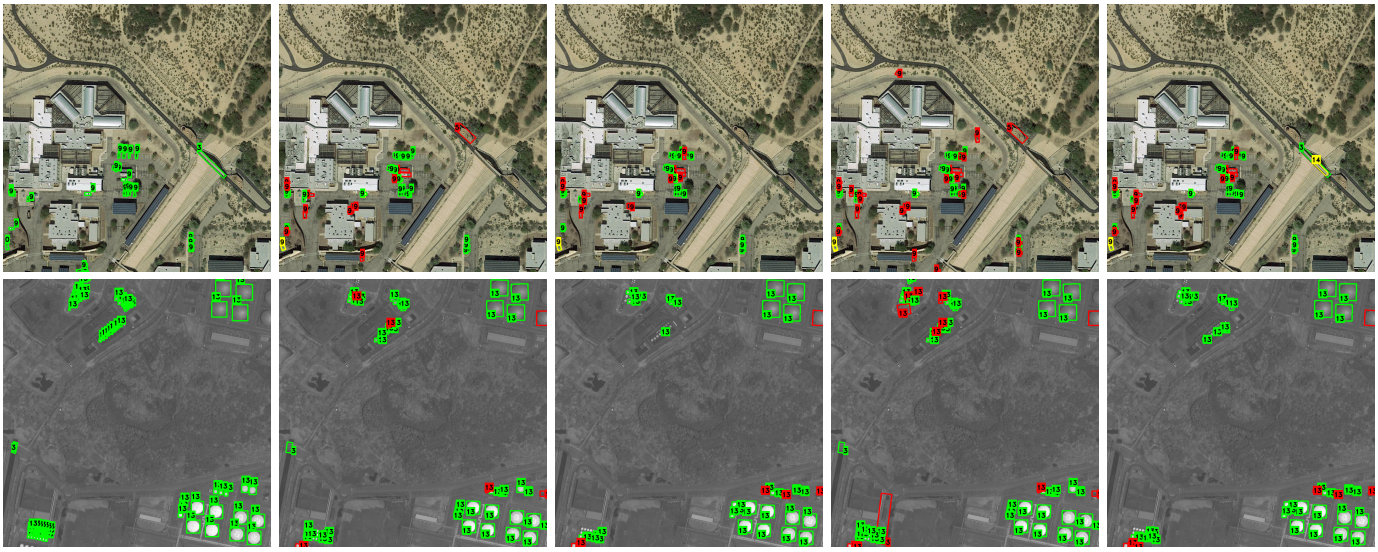


Fig. 8: Effect of the class probability threshold on the detection results on the DOTA dataset. Each row shows a different scene. Green indicates correct detections, yellow indicates detections with wrong class labels, red indicates false positives. Left to right: ground truth, detections of the oriented R-CNN baseline when threshold is 0.05, detections of the proposed model when threshold is 0.05, detections of the oriented R-CNN baseline when threshold is 0.01, detections of the proposed model when threshold is 0.01. The proposed model has higher recall at a similar precision level when the threshold is lowered but the baseline suffers from increased false positives without any improvement in recall.

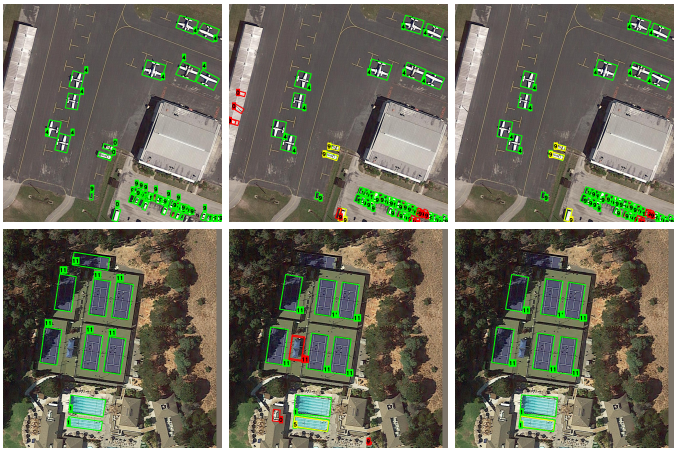


Fig. 9: Effect of the class probability threshold on the detection results on the DOTA dataset. Each row shows a different scene. Green indicates correct detections, yellow indicates detections with wrong class labels, red indicates false positives. Left to right: ground truth, detections of the oriented R-CNN baseline when threshold is 0.01, detections of the proposed model when threshold is 0.01. The proposed model has similar recall but better precision compared to the baseline.

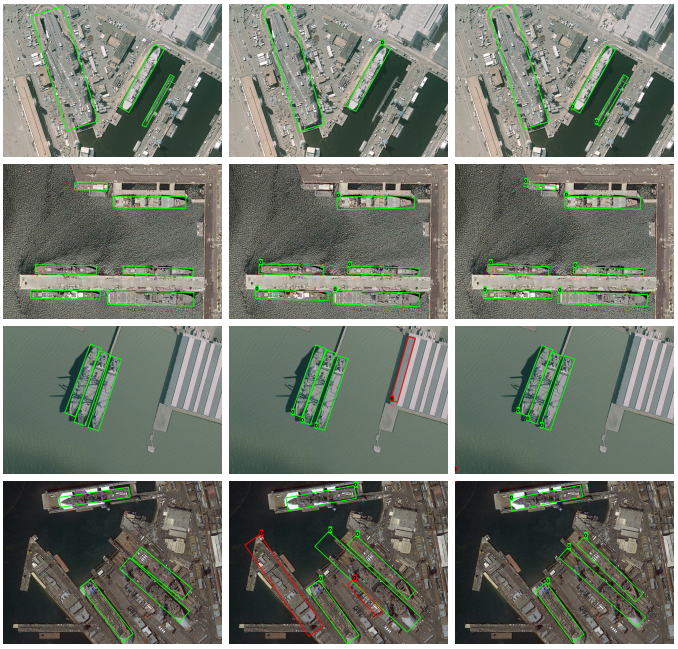


Fig. 10: Effect of the class probability threshold on the detection results on the HRSC2016 dataset. Each row shows a different scene. Green indicates correct detections, red indicates false positives. Left to right: ground truth, detections of the oriented R-CNN baseline when threshold is 0.05, detections of the proposed model when threshold is 0.01. On the top two rows, the proposed model shows higher recall while preserving precision. On the bottom two rows, the proposed model shows higher precision while preserving recall.

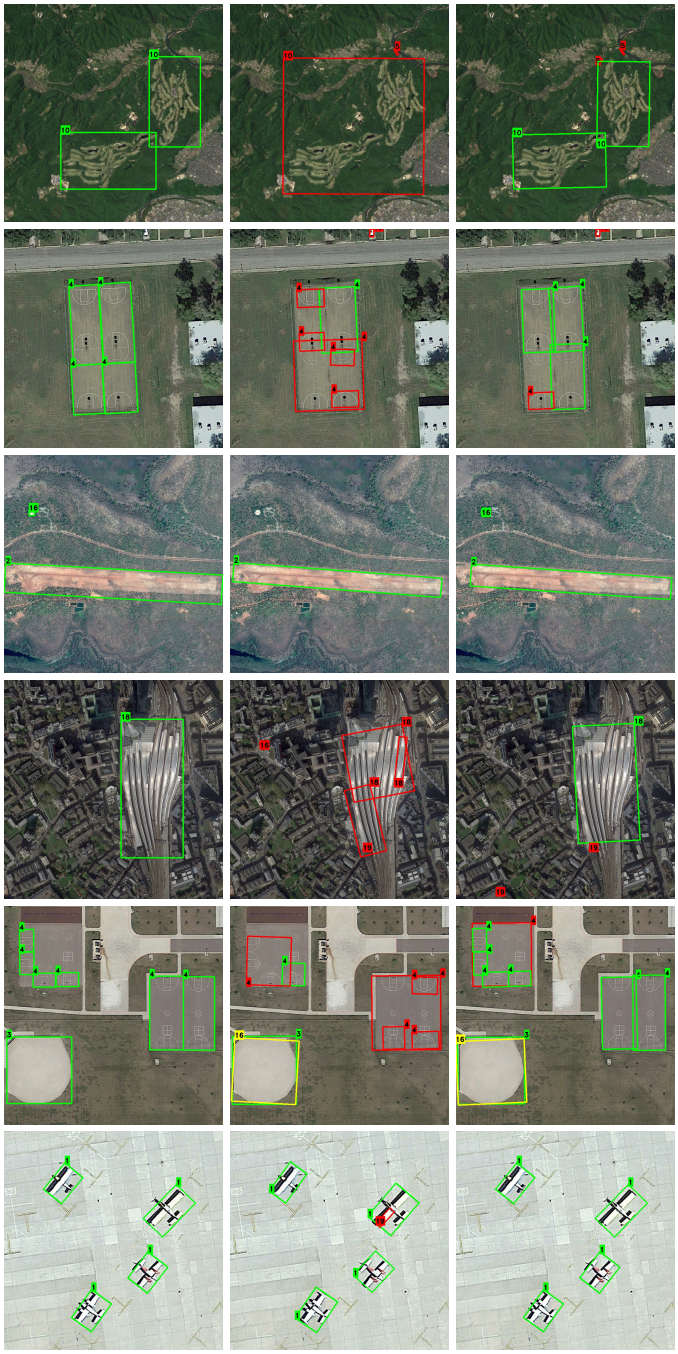


Fig. 11: Effect of the class probability threshold on the detection results on the DIOR-R dataset. Each row shows a different scene. Green indicates correct detections, yellow indicates detections with wrong class labels, red indicates false positives. Left to right: ground truth, detections of the oriented R-CNN baseline when threshold is 0.05, detections of the proposed model when threshold is 0.01. On the top three rows, the proposed model shows higher recall while preserving precision. On the bottom three rows, the proposed model shows higher precision while preserving recall.

Fig. 12: Contribution of different transformer units in the inpainting task. Each row shows a different scene. Left to right: original image, mask, outputs of transformer units 1 to 5.

E. Discussion

The experiments show that the proposed architecture that fuses contextual information captured by the inpainting model and the object features computed in the feature pyramid network layers of the oriented detector outperforms the base model for each of the DOTA, HRSC2016, and DIOR-R datasets. The most significant improvements are obtained for the classes whose identities can be consistently associated with specific environments, whereas the classes that can appear in isolation or in a wide variety of cluttered backgrounds do not receive such performance gains. Comparative experiments with recent one-stage and two-stage detection models also show that the proposed architecture achieves better overall mAP scores for all datasets. We hypothesize that modeling both local and global contextual information in the transformer-based inpainting model is more effective than competitor models that derive the scene-level contextual information using extra modules attached to the main feature extraction backbone. Visualization of the feature pyramid network and the contextual pyramid network show better emphasis of the global scene features as well as higher contrast for important local object details after fusion. Furthermore, even when the proposed model produces false positives, their labels are still more consistent with the overall scene content compared to the out-of-context predictions made by the baseline model. Finally, ablation experiments show that the proposed model has a higher mAP score compared to the baseline model for all values of the score probability threshold, and achieves a higher precision score for all recall levels. This enables a reliable control of the performance by calibrating the class probability threshold to achieve an effective trade-off between precision and recall. Of course, these improvements come at an increase in the computational overhead where it takes 3.6 times long for the proposed architecture to produce an output compared to the baseline model. However, the overall architecture can benefit from a leaner model that focuses only on the contextual feature extraction stage as opposed to the currently implemented full inpainting pipeline.

In terms of the design choices regarding the fusion module, one consideration is which particular transformer layers are matched with which FPN levels. When we use only the output of the last transformer layer X_5^T with each of the FPN levels X_1^F, X_2^F, and X_3^F, the mAP score is reduced by 6.23%. Thus, using different transformer layers according to their compatibility with FPN levels is more effective as shown in both quantitative and qualitative results in Section IV-C. Another set of design choices includes the selected activation functions, normalization types, skip connection strategies, number of convolution layers, and feature map fusion strategies in the fusion module in Figure 3. First, we consider using Smish activation [68] instead of ReLU. This necessitates the use of smaller batch sizes due to memory requirements. Even with group normalization [69] and gradient accumulation, the lightweight ReLU activation and batch normalization perform better using the settings in this paper. Furthermore, quantitative results support the discussion on Section III-C regarding different skip connection strategies similar to the original ResNet model [35] where the proposed design results in trainings that are more stable and more consistent under several runs. Next, we consider varying the number of convolution blocks before and after concatenation. Results show that applying more than two convolution blocks before concatenation and one convolution block after concatenation lead to only minor changes in the evaluation scores, and the proposed setting achieves the best performance with a lightweight design. Finally, we evaluate three different strategies, element-wise addition, channel-wise concatenation, and cross-attention, to fuse the tensors coming from the transformer units and FPN levels. Channel-wise concatenation performs better than element-wise addition, where the latter convolutes the individual contributions of the two feature representations with different characteristics. The third alternative, cross-attention, necessitates reduction in batch sizes due to memory constraints, and leads to inferior performance compared to the proposed design. Overall, the proposed fusion module is empirically shown to be more effective and efficient among different alternatives.

V. Conclusions

This paper presented CODI, a contextual object detection framework that integrates image inpainting-based context modeling into oriented object detection for remote sensing images. By fusing object-specific features from a two-stage

detector with semantic contextual representations learned independently by an inpainting model, CODI provides a contextual pyramid network that benefits both proposal generation and classification/regression stages. Experimental results on the DOTA, HRSC2016, and DIOR-R datasets demonstrated that CODI achieves state-of-the-art performance, outperforming existing methods in mean average precision. Ablation studies further confirmed the capability of the model in enhancing precision and recall with effective control of the score probability threshold. These findings highlight the potential of improving object detection by exploiting contextual information via inpainting in challenging remote sensing scenes. Future work will investigate alternative inpainting architectures and fusion strategies.

REFERENCES

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[3] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017, pp. 2980–2988.

[4] W. Ma, Q. Guo, Y. Wu, W. Zhao, X. Zhang, and L. Jiao, "A novel multi-model decision fusion network for object detection in remote sensing images," *Remote Sensing*, vol. 11, no. 7, p. 737, 2019.

[5] G. Zhang, S. Lu, and W. Zhang, "CAD-Net: A context-aware detection network for objects in remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 10 015–10 024, 2019.

[6] X. Dong, Y. Qin, R. Fu, Y. Gao, S. Liu, and Y. Ye, "Remote sensing object detection based on gated context-aware module," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[7] L. Min, Z. Fan, Q. Lv, M. Reda, L. Shen, and B. Wang, "YOLO-DCTI: Small object detection in remote sensing base on contextual transformer enhancement," *Remote Sensing*, vol. 15, no. 16, p. 3970, 2023.

[8] Z. Zhao, J. Du, C. Li, X. Fang, Y. Xiao, and J. Tang, "Dense tiny object detection: A scene context guided approach and a unified benchmark," *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[9] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536–2544.

[10] W. Li, Z. Lin, K. Zhou, L. Qi, Y. Wang, and J. Jia, "MAT: Mask-aware transformer for large hole image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 758–10 768.

[11] X. Xie, G. Cheng, J. Wang, X. Yao, and J. Han, "Oriented R-CNN for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3520–3529.

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

[13] W. Lu, S.-B. Chen, Q.-L. Shu, J. Tang, and B. Luo, "Decouplenet: A lightweight backbone network with efficient feature decoupling for remote sensing visual tasks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, 2024.

[14] X. Xie, C. Lang, S. Miao, G. Cheng, K. Li, and J. Han, "Mutual-assistance learning for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 12, pp. 15 171–15 184, 2023.

[15] X. Xie, G. Cheng, Q. Li, S. Miao, K. Li, and J. Han, "Fewer is more: Efficient object detection in large aerial images," *Science China Information Sciences*, vol. 67, no. 1, p. 112106, 2024.

[16] X. Xie, G. Cheng, W. Li, C. Lang, P. Zhang, Y. Yao, and J. Han, "Learning discriminative representation for fine-grained object detection in remote sensing images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 8, pp. 8197–8208, 2025.

[17] Q. Wu, D. Zhang, Y. Pan, and H. Zhou, "Center-symmetry representation-based high-quality localization detector for oriented object detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 63, pp. 1–14, 2025.

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

[19] H. Yu, Y. Tian, Q. Ye, and Y. Liu, "Spatial transform decoupling for oriented object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 7, 2024, pp. 6782–6790.

[20] Y. Li, Q. Hou, Z. Zheng, M.-M. Cheng, J. Yang, and X. Li, "Large selective kernel network for remote sensing object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 16 794–16 805.

[21] Z. Zhao, Q. Xue, Y. He, Y. Bai, X. Wei, and Y. Gong, "Projecting points to axes: Oriented object detection via point-axis representation," in *Proceedings of the European Conference on Computer Vision*, 2025, pp. 161–179.

[22] J. Zhao, Z. Ding, Y. Zhou, H. Zhu, W.-L. Du, R. Yao, and A. El Saddik, "OrientedFormer: An end-to-end transformer-based oriented object detector in remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[23] Y. Zeng, Y. Chen, X. Yang, Q. Li, and J. Yan, "ARS-DETR: Aspect ratio-sensitive detection transformer for aerial oriented object detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–15, 2024.

[24] X. Xie, G. Cheng, C. Rao, C. Lang, and J. Han, "Oriented object detection via contextual dependence mining and penalty-incentive allocation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–10, 2024.

[25] S. H. Khan, X. He, F. Porikli, and M. Bennamoun, "Forest change detection in incomplete satellite images with deep neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 9, pp. 5407–5423, 2017.

[26] J. Dong, R. Yin, X. Sun, Q. Li, Y. Yang, and X. Qin, "Inpainting of remote sensing sst images with deep convolutional generative adversarial network," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 2, pp. 173–177, 2018.

[27] L. Sun, Y. Zhang, X. Chang, Y. Wang, and J. Xu, "Cloud-aware generative network: Removing cloud from optical remote sensing images," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 4, pp. 691–695, 2019.

[28] Y. Du, J. He, Q. Huang, Q. Sheng, and G. Tian, "A coarse-to-fine deep generative model with spatial semantic attention for high-resolution remote sensing image inpainting," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[29] M. Shao, C. Wang, W. Zuo, and D. Meng, "Efficient pyramidal GAN for versatile missing data reconstruction in remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[30] M. Zhang, Q. Liu, and Y. Wang, "CtxMIM: Context-enhanced masked image modeling for remote sensing image understanding," 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2310.00022

[31] X. Lu, Q. Li, B. Li, and J. Yan, "Mimicdet: Bridging the gap between one-stage and two-stage object detection," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 541–557.

[32] A. Groener, G. Chern, and M. Pritt, "A comparison of deep learning object detection models for satellite imagery," in *2019 IEEE Applied Imagery Pattern Recognition Workshop*, Oct. 2019, pp. 1–10.

[33] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.

[34] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[36] ——, "Identity mappings in deep residual networks," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 630–645.

[37] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "DOTA: A large-scale dataset for object detection in aerial images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3974–3983.

[38] Z. Liu, L. Yuan, L. Weng, and Y. Yang, "A high resolution optical satellite image dataset for ship recognition and some new baselines," in *International Conference on Pattern Recognition Applications and Methods*, vol. 2, 2017, pp. 324–331.

[39] G. Cheng, J. Wang, K. Li, X. Xie, C. Lang, Y. Yao, and J. Han, "Anchor-free oriented proposal generator for object detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2022.

[40] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2010.

[41] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, pp. 98–136, 2015.

[42] Z. Chen, K. Chen, W. Lin, J. See, H. Yu, Y. Ke, and C. Yang, "Piou loss: Towards accurate oriented object detection in complex environments," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 195–211.

[43] X. Pan, Y. Ren, K. Sheng, W. Dong, H. Yuan, X. Guo, C. Ma, and C. Xu, "Dynamic refinement network for oriented and densely packed object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 207–11 216.

[44] Q. Wu, X. You, W. Huang, L. Sun, Y. Xu, and X. Wang, "M2FE-YOLO: Multi-branch and multi-level feature enhancement network for remote sensing object detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 63, pp. 1–19, 2025.

[45] G. Nie and H. Huang, "Multi-oriented object detection in aerial images with double horizontal rectangles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4932–4944, 2022.

[46] L. Hou, K. Lu, J. Xue, and Y. Li, "Shape-adaptive selection and measurement for oriented object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, 2022, pp. 923–932.

[47] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning RoI transformer for oriented object detection in aerial images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2849–2858.

[48] S. M. Azimi, E. Vig, R. Bahmanyar, M. Körner, and P. Reinartz, "Towards multi-class object detection in unconstrained remote sensing imagery," in *Asian Conference on Computer Vision*, 2018, pp. 150–165.

[49] Z. Xu, Y. Xu, H. Wang, Z. Wei, and Z. Wu, "Horizontal box supervised oriented object detection based on edge feature self-attention and multi-task synthetic visual patterns," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 18, pp. 21 883–21 899, 2025.

[50] D. Lu, "OSKDet: Towards orientation-sensitive keypoint localization for rotated object detection," 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2104.08697

[51] Z. Xiao, G. Yang, X. Yang, T. Mu, J. Yan, and S. Hu, "Theoretically achieving continuous representation of oriented bounding boxes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16 912–16 922.

[52] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3111–3122, 2018.

[53] Y. Xu, M. Fu, Q. Wang, Y. Wang, K. Chen, G.-S. Xia, and X. Bai, "Gliding vertex on the horizontal bounding box for multi-oriented object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 4, pp. 1452–1459, 2020.

[54] X. Yang, J. Yan, Z. Feng, and T. He, "R3det: Refined single-stage detector with feature refinement for rotating object," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, 2021, pp. 3163–3171.

[55] Q. Ming, Z. Zhou, L. Miao, H. Zhang, and L. Li, "Dynamic anchor learning for arbitrary-oriented object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, 2021, pp. 2355–2363.

[56] J. Wang, W. Yang, H.-C. Li, H. Zhang, and G.-S. Xia, "Learning center probability map for detecting objects in aerial images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 4307–4323, 2020.

[57] J. Han, J. Ding, J. Li, and G.-S. Xia, "Align deep features for oriented object detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2021.

[58] X. He, K. Liang, W. Zhang, F. Li, Z. Jiang, Z. Zuo, and X. Tan, "DETR-ORD: An improved DETR detector for oriented remote sensing object detection with feature reconstruction and dynamic query," *Remote Sensing*, vol. 16, no. 18, p. 3516, 2024.

[59] G. Cheng, J. Wang, K. Li, X. Xie, C. Lang, Y. Yao, and J. Han, "Anchor-free oriented proposal generator for object detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2022.

[60] Z. Zhao and S. Li, "OASL: Orientation-aware adaptive sampling learning for arbitrary oriented object detection," *International Journal of Applied Earth Observation and Geoinformation*, vol. 128, p. 103740, 2024.

[61] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9627–9636.

[62] Z. Xiao, Z. Li, J. Cao, X. Liu, Y. Kong, and Z. Du, "OriMamba: Remote sensing oriented object detection with state space models," *International Journal of Applied Earth Observation and Geoinformation*, vol. 143, p. 104731, 2025.

[63] G. Cheng, Y. Yao, S. Li, K. Li, X. Xie, J. Wang, X. Yao, and J. Han, "Dual-aligned oriented detector," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2022.

[64] C. Zhang, Z. Chen, B. Xiong, K. Ji, and G. Kuang, "EOOD: End-to-end oriented object detection," *Neurocomputing*, vol. 621, p. 129251, 2025.

[65] J. Yang, L. Zhou, and Y. Ju, "AFDR-Det: Adaptive feature dual-refinement oriented detector for remote sensing object detection," *IEEE Access*, vol. 13, pp. 32 901–32 917, 2025.

[66] X. Qian, J. Zhao, B. Wu, Z. Chen, W. Wang, and H. Kong, "Task-aligned oriented object detection in remote sensing images," *Electronics*, vol. 13, no. 7, p. 1301, 2024.

[67] Y. Li, J. Shen, R. Liu, X. Guo, Y. Chen, R. Shang, and L. Jiao, "A scalable target orientation detection method for remote sensing images based on improved yolox algorithm," *IEEE Geoscience and Remote Sensing Letters*, vol. 21, pp. 1–5, 2024.

[68] X. Wang, H. Ren, and A. Wang, "Smish: A novel activation function for deep learning methods," *Electronics*, vol. 11, no. 4, p. 540, 2022.

[69] Y. Wu and K. He, "Group normalization," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 3–19.

Sinan Çavdar received the B.Sc. degree in Computer Engineering from TED University, Ankara, Türkiye, in 2021, and the M.Sc. degree in Computer Engineering from Bilkent University, Ankara, in 2025. He is currently pursuing the Ph.D. degree in Computer Engineering at Bilkent University. His research interests include computer vision and generative learning, with a particular focus on remote sensing and medical imaging.

Selim Aksoy (S'96-M'01-SM'11) received the B.S. degree from Middle East Technical University, Ankara, Türkiye, in 1996 and the M.S. and Ph.D. degrees from the University of Washington, Seattle, in 1998 and 2001, respectively. During 2001–2003, he was a Research Scientist at Insightful Corporation in Seattle. He joined the Department of Computer Engineering, Bilkent University, Ankara in 2004 where he is currently a Professor and the Department Chair. He was a Visiting Associate Professor at the Department of Computer Science and Engineering, University of Washington in 2013. His research interests include computer vision, pattern recognition, and machine learning with applications to remote sensing and medical imaging.

Dr. Aksoy is a member of the IEEE Geoscience and Remote Sensing Society, the IEEE Computer Society, and the International Association for Pattern Recognition (IAPR). He was one of the Guest Editors of the special issues on Pattern Recognition in Remote Sensing of IEEE Transactions on Geoscience and Remote Sensing, Pattern Recognition Letters, and IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing in 2007, 2009, and 2012, respectively. He served as the Vice Chair of the IAPR Technical Committee 7 on Remote Sensing during 2004–2006, and as the Chair of the same committee during 2006–2010. He also served as an Associate Editor of Pattern Recognition Letters during 2009–2013.