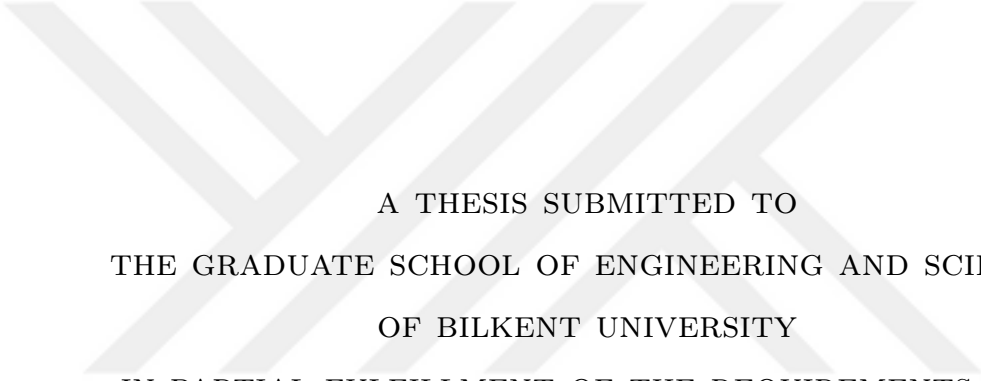


CONTEXTUAL OBJECT DETECTION VIA IMAGE INPAINTING



A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

By
Sinan Çavdar
July 2025

Contextual Object Detection via Image Inpainting

By Sinan avdar

July 2025

We certify that we have read this thesis and that in our opinion it is fully adequate,
in scope and in quality, as a thesis for the degree of Master of Science.



Selim Aksoy(Advisor)

Mehmet Erkut Erdem

Doruk ner

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

CONTEXTUAL OBJECT DETECTION VIA IMAGE INPAINTING

Sinan Cavdar
M.S. in Computer Engineering
Advisor: Selim Aksoy
July 2025

Object detection in aerial imagery is a critical task in computer vision with applications in urban monitoring, disaster management, and military surveillance. Current approaches often encounter challenges, such as a lack of representative features of objects with cluttered backgrounds. To address these challenges, we introduce Contextual Object Detection via Image Inpainting (CODI), a novel approach that extract the representative features of the image’s semantic context derived from the transformer units of the inpainting model, and our proposed fusion module learns how to inject these contextual features with the representative features of the objects obtained from the related Feature Pyramid Network (FPN) of the object detection model. The outputs, which we called Contextual Pyramid Network (CPN), of the fusion model are used to replace the original FPN layers for better localization and labeling accuracy over the objects. Our experiments were conducted on DOTA and HRSC2016 datasets. Our model achieved promising results on mAP by prevailing over Oriented RCNN, which is one of the best-performing models on the DOTA dataset. Specifically, by applying the same pre-processing and post-processing, CODI achieved an improvement of 0.67% in mean average precision (mAP) at single-scale and 0.6% at multi-scale over the Oriented RCNN on the test set. On the HRSC2016 dataset, CODI achieved an impressive mAP of 90.57%, outperforming Oriented RCNN by 0.27 percentage points. This result places CODI among the top-performing models using a ResNet-50 backbone. In that sense, our approach provides a foundation for future applications, paving the way for more precise object localization and identification in remote sensing.

Keywords: Image Inpainting, Transformers, Oriented Object Detection, Contextual Object Detection, Remote Sensing.

ÖZET

RESİM DOLDURMA İLE BAĞLAMSAAL NESNE TANIMA

Sinan Çavdar
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: Selim Aksoy
Temmuz 2025

Uydu görüntülerinde nesne tespiti, şehirlerin gözlemlenmesi, afet yönetimi ve askeri gözetim gibi uygulamalarla bilgisayarla görü alanında kritik bir yer almaktadır. Ancak, mevcut yaklaşımlarda genellikle karışık arka planlar gibi nedenlerden dolayı nesnelerin temsil edici özelliklerinin yeterli olmaması gibi zorluklarla karşılaşmaktadır. Bu çalışmada, Resim Doldurma ile Bağlamsal Nesne Tanıma modelimizi sunmaktayız. Bu yeni yaklaşımımızda resim doldurma modelinin dönüştürücüler kısmındaki öğrenilmiş parametrelerini kullanarak resimlerdeki genel bağlamı elde etmekteyiz. Geliştirdiğimiz füzyon modelimizi kullanarak elde ettiğimiz resimlerdeki bağlamlar ile nesne tanıma modelindeki karşılık gelen öznitelik piramit ağındaki nesneleri temsil eden verileri harmanlamaktayız. Böylelikle bağlam piramit ağı olarak adlandırdığımız tümleşim çıktılarımızı öznitelik piramit ağındaki ilgili katmanların yerine geçirerek daha iyi bir şekilde nesneler için konum ve etiket tespiti yapabilmekteyiz. Deneylerimiz DOTA ve HRSC2016 veri kümelerinde yapılmıştır. Modelimiz, yapılan deneylerde, literatürde en yüksek ortalama kesinlik (mAP) sonuçlarından birini veren OrientedRCNN modelini DOTA sınama veri kümesinde aynı ön işlem ve son işlem yöntemlerini kullanarak tek ölçekli resimlerde 0.67% ve çok ölçekli resimlerde ise 0.60% üstünlük göstermektedir. Modelimiz HRSC2016 veri kümesinde ise 90.57% mAP değeri elde ederek Oriented-RCNN'i 0.27% lik bir farkla geçmiş olup en yüksek değerlerden birini elde etmiştir. Bu bağlamda modelimiz, gelecekteki uydu görüntülerinde nesne tespiti uygulamaları için yeni bir yaklaşım sunmakta olup, nesnelerin daha doğru bir şekilde konumlandırılması ve tanımlanmasına öncülük etmektedir.

Anahtar sözcükler: Resim Doldurma, Dönüştürücüler, Yönelimli Nesne Tanıma, Bağlamsal Nesne Tanıma, Uzaktan Algılama.

Acknowledgement

To begin with, I am deeply grateful to my advisor, Prof. Selim Aksoy, whose insightful guidance, encouragement, and thoughtful mentorship have been fundamental for the duration of my graduation. His willingness to question my views while supporting my ideas has strengthened my research and profoundly influenced my personal and professional growth.

I also immensely appreciate the members of my thesis committee, Prof. Mehmet Erkut Erdem and Asst. Prof. Dr. Doruk Öner, who participated in partitioning their time and support. I am very thankful for the careful and thorough way my thesis committee handled this process, as their help has been crucial for the accuracy and importance of my research.

I am deeply thankful to my parents, who have been a steady source of love, support, and understanding throughout my academic journey. Their spiritual and financial support helped me to bring this respectable position today.

Lastly, I would like to thank the Retina Vision and Learning Group lab and my exceptional colleagues, Emir Türkölmez and Vahid Namakshenas. Collaborating with such a sincere and supportive group of individuals has been a privilege. In this respect, I hope these strong bonds with my valuable colleagues will come with me forever.

Contents

1	Introduction	1
2	Related Work	5
3	Methodology	11
3.1	Oriented Object Detector	12
3.2	Inpainting Model	14
3.2.1	Transformer Units	15
3.2.2	Inpainting Setting and Training	16
3.3	Overview of the Contextual Detection Mechanism	17
3.4	Contextual Feature Fusion	18
3.5	Loss Functions	20
3.5.1	Object Detection Losses	20
3.5.2	Image Inpainting Losses	21
3.6	Overall Model Training	22

4	Implementation Details	23
4.1	Dataset	23
4.2	The Design Choices	24
4.3	Training Procedure	25
4.4	Evaluation Metrics for Object Detection and Inpainting Model . .	26
5	Experiments	27
5.1	Evaluation in DOTA-v1	27
5.2	Evaluation in HRSC2016	29
5.3	Contextual Pyramid Network Visualization	35
5.4	Ablation Study	40
5.4.1	Score Threshold Analysis	40
5.4.2	Scene Level Prediction Analysis	51
5.4.3	Different Sizes of Chess-masks	52
5.4.4	Different Design Choices	62
5.5	Discussion	68
6	Conclusion	69

List of Figures

1.1	Detection results for the proposed and oriented R-CNN models at varying class probability thresholds	3
3.1	Visualization of the proposed model’s inference pipeline	11
3.2	Contextual Object Detection via Inpainting (CODI) architecture	15
3.3	Fusion architecture	18
5.1	Examples of the improved detection performance achieved by the proposed model compared to the baseline in HRSC2016 dataset .	31
5.2	Illustrating the precise detection performance of the proposed model against the baseline in HRSC2016 dataset.	32
5.3	Continue of figure 5.2	33
5.4	Illustrating the HRSC2016 dataset inpainting results obtained from MAT weights trained on the DOTA-v1 dataset	34
5.5	Visualization of the feature pyramid network (FPN) levels and the attention maps of the transformer layers	36
5.6	Continuation of Figure 5.5.	37

5.7	Continuation of Figure 5.6.	38
5.8	Continuation of Figure 5.7.	39
5.9	Effect of class probability threshold on performance metrics for the proposed model and oriented R-CNN	43
5.10	Effect of class probability threshold on DOTA detection results for recall	48
5.11	Effect of class probability threshold on DOTA detection results for precision	49
5.12	Continue of figure 5.11	50
5.13	Illustrating the distribution of object classes under each cluster, which presents different scenes.	51
5.14	Demonstrating the distribution of false positive predictions that belong to contextual or non-contextual classes.	53
5.15	Demonstration of MAT inpainting performance on different chess- mask settings	55
5.16	Continue of figure 5.15	56
5.17	Demonstration of the behavior of different mask settings under MAT architecture	57
5.18	Continue of figure 5.17	58
5.19	Demonstrating each transformer unit's behavior and individual contribution to the inpainted images	59
5.20	Continue of figure 5.19	60

5.21 Effect of partial convolution kernel size in the convolution head for inpainted images	61
5.22 Demonstrating our model's training monitor under 18 epochs. . .	62



List of Tables

5.1	Comparison of the proposed model CODI on the DOTA dataset. .	29
5.2	Comparison of the proposed model CODI on the HRSC2016 dataset	30
5.3	Comparison of the proposed model CODI with the Oriented RCNN in terms of classwise recall scores at various detection thresholds .	44
5.4	Comparison of the proposed model CODI with the Oriented RCNN in terms of classwise precision scores at various detection thresholds	45
5.5	Comparison of the proposed model CODI with the Oriented RCNN in terms of both classwise and overall mAP scores at various detec- tion thresholds	46
5.6	Comparison of the different chess mask settings	47
5.7	Analysis of different fusion models	66
5.8	Analysis of different hyperparameter tuning for the training . . .	67
5.9	Analysis of different learning strategies	67

Chapter 1

Introduction

In recent years, satellite technology advancements have driven numerous new applications, such as environmental monitoring, urban development, agriculture, and disaster management. One of the most common applications in this field is object detection in remote-sensing images. We can study object detection in two parts: localization and identification. However, object detection in remote sensing presents several unique challenges compared to traditional object detection applications.

Specific to remote sensing object detection, a complex and heterogeneous background is one of the well-known problems. For instance, detecting small vehicles parked along streets in densely built urban areas can be particularly challenging as they blend with the surrounding infrastructure, such as roads, buildings, and shadows, making it hard for detection models to distinguish the objects from the background. Another highly known issue is the high variability in scale among the types of objects, from small vehicles to baseball fields.

Lacking contextual information further complicates remote sensing object detection, as inter-similarity among different object classes can hinder model performance. For instance, baseball diamond, ground track field, tennis court, and

soccer ball field may share color, texture, or shape, making them difficult to distinguish. Intra-variability is another major issue, with significant differences within the same class. Vehicles, for example, can vary in color, size, and orientation or be partially occluded, adding complexity to detection tasks.

These difficulties highlight the importance of incorporating contextual information to improve object detection accuracy. Leveraging the relationships between objects and their surrounding helps for better localization and labeling for object detection. For instance, we expect small and large vehicles to be near or on the road. Similarly, buildings tend to form clusters. The ability to capture spatial and semantic relationships between objects and their environment is crucial for enhancing detection performance in remote sensing images.

In recent years, the general trend for object detection in remote sensing images has shifted towards deep learning techniques under single-stage and two-stage architectures. Traditional methods, which relied on handcrafted features and techniques like sliding window approaches, have mainly been replaced by deep learning models that automatically extract representative features and learn from them. Popular object detection algorithms for single- and two-stage algorithms such as Faster RCNN[1], YOLO[2], and Retina-Net[3] have been adopted for remote sensing images to overcome large-scale images, high-resolution data, and multispectral inputs. Recent improvements in this field focus more on improving small objects' detection, optimizing models for computational efficiency, and facilitating multi-scale feature learning to detect objects of varying sizes better. Also, there has been an increasing trend in integrating contextual information. This trend highlights the growing demand for improved overall performance in object detection, especially in applications requiring a low margin of error without emphasizing inference times. However, many recent papers that discuss the use of context in object detection implicitly claim to inject contextual information by modifying existing models and adding additional modules. These modules aim to learn from the highest receptive field layers of the backbone and connect this information to later stages, such as the bounding box regression head or classification head. Some approaches also propose ensembling the original model's heads with an additional model designed to store and leverage contextual information.

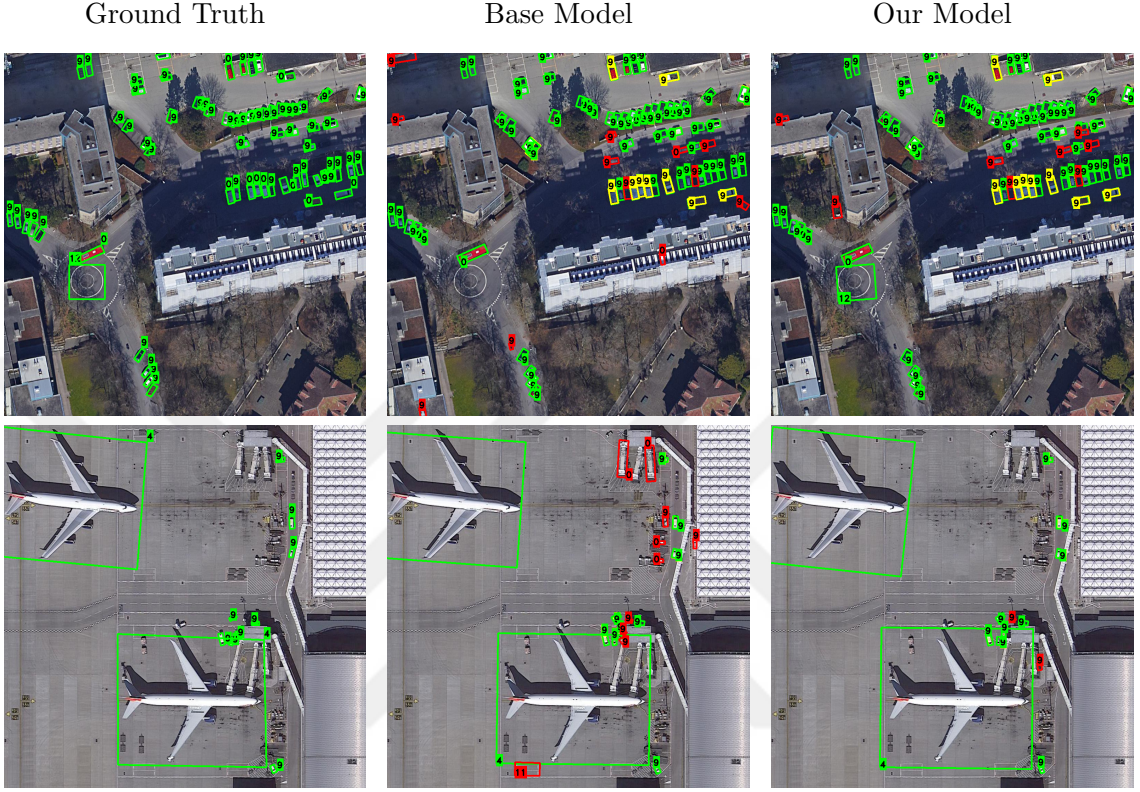


Figure 1.1: Detection results for the proposed and oriented R-CNN models at varying class probability thresholds. Green indicates correct detections, yellow indicates detections with wrong class labels, red indicates false positives. In the first row, similar precision is achieved using thresholds of 0.02 (our model) and 0.04 (base model). In the second row, similar recall is obtained at thresholds of 0.04 (our model) and 0.05 (base model).

This research obtains the semantic contextual information via the inpainting model’s transformer units. Custom fusion model helps to inject the obtained context properly into the Feature Pyramid Network (FPN) of the object detection model to increase objects’ localization and labeling accuracy in the given image. In that sense, we exploit one of the best-performing generative-based image inpainting model, which is Mask Aware Transformer inpainting (MAT)[4], to obtain representative contextual information for the given image. For object detection, we preferred to use the Oriented RCNN[5] because it is a lightweight oriented

model and one of the best-performing baseline two-stage models in remote sensing images. Another key motivation of this research is to surpass the baseline model in either recall or precision by adjusting the confidence score threshold during inference, while keeping the other metric at a comparable level. As illustrated in Figure 1.1, the top row demonstrates enhanced recall while keeping precision relatively constant, whereas the bottom row presents improved precision with recall held on a similar level for both base and our model.

To the best of our knowledge, this study presents the first effective use of a generative model for object localization and classification. Unlike most of the previous implicit methods, our approach extracts contextual information from an independent architecture that is forced to learn both short and long-range dependencies between the parts of the given image. The results demonstrate promising potential for future research integrating inpainting models with object detection.

For the following chapters, we mention previous works in Chapter 2 on important object detection, contextual object detection, and inpainting studies in remote sensing images. Later, we discuss our proposed architecture in Chapter 3. Afterwards, we talk about the implementation details in Chapter 4. Our comprehensive experiments are presented in Chapter 5. Finally, we deduce important outcomes of our study in Chapter 6.

Chapter 2

Related Work

This section first discusses the important object detection papers on remote sensing images. Afterward, we mention the recent contextual object detection papers on aerial images. Finally, we discuss the recent advancements in image inpainting papers under remote sensing images.

In recent years, very important new concepts have been published in object detection under remote sensing images. Xie et al.[5] proposed Oriented RCNN, a two-stage detector designed to generate high-quality oriented proposals for the first time efficiently. In this way, they achieved remarkable results using oriented bounding boxes as a baseline object detector. Later, Pu et al.[6] introduced the Adaptive Rotated Convolution (ARC) module to enhance the ability of convolutional neural networks to handle objects with arbitrary orientations. Their proposed architecture provides convolution kernels to rotate adaptively based on the object's orientation in the image. In this way, they improve the feature extraction of objects in different orientations. Recently, DecoupleNet, released by Lu et al.[7], introduces a lightweight backbone network specifically designed for resource-constrained environments in remote sensing applications. Their provided architecture has a promising classification and detection performance among existing lightweight networks. These prominent studies released in recent years mainly focus on detecting objects based solely on object features,

by lack of contextual information, compared to our work. Although orientation-based methods outperform traditional horizontal-based detectors at capturing object features using orientation techniques, whether by applying them within the network by modifying convolutions or through explicit processing for bounding box predictions to differentiate objects from the background more easily, they are not capable of identifying false negatives. Therefore, false negative objects are mistakenly accepted as background. Using a semantic contextual information, our model gains more power to promote the representative features of the objects to better localization and correct identification, even if the objects lack representative features or are in a cluttered background in the image.

With the ultimate progression in large language models, transformer models have drawn attention to different research fields under deep learning. We observed a similar trend, followed by studies on object detection in remote-sensing images. The ViTAE, published by Wang et al.[8], is one of the first studies specifically designed for remote sensing images using transformers. Compared to Vision Transformer (ViT), Vitae uses different scales of window attention with different orientations, which helps with the computational cost and better object representation. Yu et al.[9] introduced the Spatial Transform Decoupling (STD) approach, which employs a separate network to predict the orientation of the bounding box of the object within the detection head. Compared to the ViTAE, they additionally implement Cascaded Activation Masks (Cams) into the self-attention modules to improve the features in the region of interest. Compared to previous studies, the Large Selective Kernel Network (LSKNet) proposed by Li et al.[10] focused more on contextual requirements for object detection. To capture appropriate context for different scales of objects, LSKNet dynamically adjusts the receptive fields of the selective kernels within its convolutional layers under the feature extraction backbone. Zhao et al.[11] introduced a Point-Axis representation to address challenges with traditional bounding box methods, such as loss discontinuity and abrupt rotation changes. Their approach decouples object location and rotation by representing objects with point sets for shape description and axes for orientation cues. By integrating this representation into the Detection Transformer (DETR) framework, they developed the Oriented DETR model.

The vast majority of these studies cannot obtain enough richness of object representative features, even if they use implicit contextual information obtained from transformer architectures under the backbone. So, their provided mechanisms' capability is more limited than consciously injecting the context as provided in our model.

Despite the huge potential, the integration of contextual information for object detection in remote sensing imagery remains relatively under-explored. Much of the existing research in this field implicitly integrates contextual information by modifying backbone architectures.

To improve detection performance, Ma et al. [12] proposed a multi-model decision fusion network based on VGG16 and faster R-CNN. Their approach integrates a contextual information fusion sub-network based on gated recurrent unit (GRU), combining local contextual features with object-object relationship features to improve recognition of complex objects. Additionally, they introduced a part-based multi-region fusion sub-network that merges multiple parts of an object to capture detailed spatial information. Compared to the previous study, Context-Aware Detection Network (CAD-Net)[13] directly obtains local and global contextual information from the representative features over specific layers under the backbone of the object detection model. The global context obtained by ResNet's top layer processed by Global Context Network (GCNet) and Pyramid Local Context Network (PLCNet) captures object-level context from the FPN levels. Later, those modules' outputs are combined in the second stage of the object detection model. More recently, Dong et al.[14] proposed a Gated Context-Aware Module (G-CAM), which obtained local and global contextual features from the model itself, like CAD-Net. The G-CAM replaces the standard RoIAlign in FPN with a module that adaptively integrates useful local context around each proposal and the global context of the entire image. This is achieved through a gated function that selectively passes relevant contextual features to the detection network. Afterward, Min et al.[15] released YOLO-DCTI, which, based on YOLOv7, has a similar strategy that GCAM has embraced. They integrated the Contextual Transformer module (CoT-I) into the backbone of the detection network to better capture spatial context and channel features. The

CoT-I module incorporates global residual connections and local fusion strategies, allowing the network to learn deeper-level representations. Additionally, they designed a decoupled detection head structure (DCTI) to mitigate feature conflicts between classification and regression tasks and improve the detection of small objects. Scene Contextualized Detection Network (SCDNet) is one of the latest studies in this research field released by Zhao et al.[16]. Their provided architecture consists of three main modules; those are scene classification sub-network (SCN), Scene Context Guided Fusion Module (SCGM), and Foreground Enhancement Module (FEM). SCN module, based on the ResNet18 backbone network, is used for capturing the scene contextual information to figure out the relationship between objects and their surrounding. SGGM is used to integrate scene context features with detection features. Additionally, FEM is a unique approach compared to previous works to enhance foreground object features while suppressing background interference. Most of these mentioned papers, which use contextual information, try to obtain contextual information from the parts of the backbone of object detection models and integrate these implicit contextual representative features into the later stages of the baseline architectures using extra modules. In this regard, our approach is one of the exceptional ones that obtains the context by benefiting from the generative mechanism to specifically force learning both short and long-range dependencies among different parts of the given image independently from the object detection architecture with more intuitive and straightforward design choices.

Applying image inpainting methods under different applications of remote sensing images is a new era, the capabilities of which have not yet been fully covered. The studies can be accumulated under object removal and painting of the occluded parts. Some prominent applications under these studies are cloud and shadow removal, removal of man-made objects and artifacts, and anonymization for privacy preservation to obscure sensitive locations.

One of the first studies in this field was released by Khan et al.[17] using a spatiotemporal inpainting mechanism to fill the missing surface information due to the consisting of artifacts, cloud occlusion, and limited camera aperture in the aerial images. They provided a two-stage mechanism to overcome the

lack of representative information for monitoring the changes in the terrain. In the first stage, image inpainting is used to fill in the missing spatial data with higher temporal resolution. The inpainted satellite image is used in the second stage, with learning region representation via convolution models for accurate monitoring. Compared to a previous study, Dong et al.[18] focused on cloud occlusion in remote sensing sea surface temperature (SST), facilitated by Deep Convolutional Generative Adversarial Network (DCGAN). Their aim is to learn the distribution of uncorrupted SST images from historical images by training DCGAN. Later, the trained DCGAN was used to reconstruct occluded regions caused by clouds. This study was one of the first promising results that highlight the potential of GAN-based models in remote sensing applications, particularly under restoration tasks. Similarly, Sun et al.[19] focused on removing clouds from single optical remote sensing images without relying on multitemporal information by proposing the Cloud-Aware Generative Network (CAGN). Their proposed architecture combined image inpainting and denoising techniques under two-stage architecture. In the first stage, they apply a recurrent convolution network with an attention mechanism to detect potential cloudy regions and generate a mask for those regions. For the second stage, they use an auto-encoder embedded with partial convolution layers to update the masked area gradually. More recently, Du et al.[20] pointed out the problem of missing information in high-definition remote sensing images caused by CCD splicing gaps by proposing a coarse-to-fine deep generative model with a novel Spatial Semantic Attention (SSA) architecture. Their approach consists of a two-stage network; a coarse network provides rough inpainting of missing areas, and a fine network refines the inpainting results with detailed operations. The SSA mechanism, deployed in both the encoder and decoder of the fine network, enhances local feature continuity and global semantic relevance by modeling the relationships between known and missing regions. Integrating SSA into their model helps reconstruct the missing regions while preserving structural integrity and texture details. Sha et al.[21] proposed an efficient pyramidal generative adversarial network (GAN), one of the first papers that focused on versatile missing data reconstruction in remote sensing images. Their architecture consists of a Mask Extraction Network (MEN) to obtain a unified soft mask that represents intrinsic priors under

various scenarios, such as a thin cloud, and a Unified Inpainting Network (UIN) that repairs diverse degraded images. By using dilated pyramidal convolutions (DPCs) and an attention fusion mechanism (AFM), they improved feature extraction. In this regard, our work focuses more on obtaining a semantic representation rather than reconstructing the specific parts of an image. Different from the mentioned used cases of the inpainting models under remote sensing images, Zhang et al.s'[22] Context-Enhanced Masked Image Modeling (CtxMIM) for remote sensing image understanding focuses more on learning representative features from masked-image reconstruction. By formulating the original image patches as reconstructive templates and employing a Siamese network architecture, their method provides contextual information through context consistency constraints during reconstruction. In this regard, we have a similar approach; instead of pushing Siamese network architecture to capture contextual information to reconstruct the images, we directly use an inpainting architecture to capture both short and long-range contextual relational understanding to fill the missing parts of the image. The majority of these inpainting studies under remote sensing apply common use cases of inpainting in different research areas, such as occlusion or artifact reduction. Neither of these studies directly points out the capability of image inpainting's contextual learning to adapt effectively to object detection.

Compared to the previous remote sensing studies under 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 using contextual information obtained from the inpainting model. In contrast to previous architectures, our model obtains semantic context independently from the backbone of the object detection model. Also, this is the first time that inpainting has been integrated with the object detection model to improve not just the performance of identifying objects but also the localization.

Chapter 3

Methodology

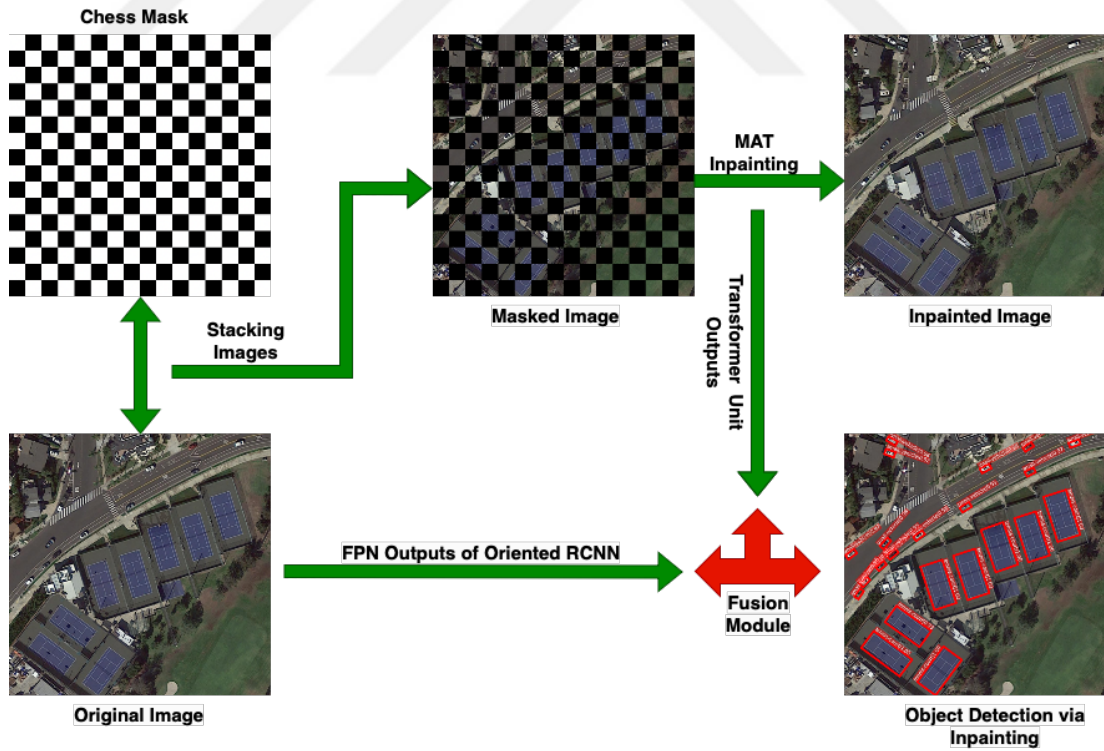


Figure 3.1: Visualization of the proposed model’s inference pipeline on a sample input image, demonstrating the sequential processing and resulting predictions.

The proposed methodology’s main goal is to improve the detection (localization) and classification (labeling) performance of the object detection model by injecting semantic contextual information into the feature representation pipeline. Our approach focuses on three key aspects: extracting representative features of objects through the backbone of the object detection model, obtaining contextual information using image inpainting, and employing a fusion module to combine these elements effectively. As shown in Figure 3.1, this method integrates contextual information from the inpainting model with object features from the backbone for enhanced performance. In later sections, we discuss design choices and use cases for object detection, inpainting, and fusion modules.

3.1 Oriented Object Detector

There are several highly well-known one-stage and two-stage models with specialized versions applied to remote sensing images, such as You Only Look Once[2] and Faster RCNN[1]. Even though many improvements have been made to one-stage algorithms in recent years, they cannot prevail over two-stage algorithms for several reasons. Localization under complex environments[23] and detection of small objects[24] are the most prominent problems of single-stage algorithms under satellite images. Due to hardware improvements, the duration margin under inference between one-stage and two-stage approaches gradually decreased. For these reasons, we constrain our baseline object detection research findings under two-stage algorithms. In addition, we find that oriented object detection models outperform horizontal object detection. The main reason behind this performance gap between the oriented and horizontal approaches is that the oriented bounding box prevents background noise when capturing the object. In contrast, the horizontal bounding box also captures representative features of objects with background noise, especially if it is not aligned with the actual orientation of objects. In this regard, we preferred to use the Oriented R-CNN[5] model as a baseline for our architecture.

The Oriented R-CNN model extends Faster R-CNN to handle arbitrarily oriented objects by adding key structural enhancements. The model begins with a feature extraction backbone followed by a Feature Pyramid Network (FPN), combining multi-scale feature maps to leverage low-level detail and high-level semantic features, enhancing detection across varied scales. The Oriented Region Proposal Network (RPN) works on each FPN level to generate bounding box proposals, predicting parameters such as each box’s center coordinates, width, height, and orientation. This allows it to produce proposals that align with the actual orientation of objects. Next, a Rotated Region of Interest (ROI) Align process extracts feature representations from the FPN feature maps, aligned to the proposed regions’ rotation. These aligned features are pooled into a fixed size and passed through fully connected layers in the second detection stage to classify objects and refine bounding box parameters. The final output includes refined oriented bounding boxes and object classifications, ensuring accurate localization and recognition.

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 size 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 shown in Figure 3.2. 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 fine 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 $F_i, i = 1, \dots, 4$ 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 features 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 will be used in the contextual feature fusion in Section 3.4.

3.2 Inpainting Model

The Mask Aware Transformer (MAT)[4] for Image Inpainting model addresses the task of reconstructing missing or damaged regions in images by taking advantage of transformers to capture both local and global context. Inpainting aims to fill in the missing parts of an image so that the result looks natural and consistent with the surrounding context. Traditional inpainting approaches generally struggle with large, complex missing regions. The MAT utilizes the self-attention mechanism in transformers to model long-range dependencies between different parts of the image, helping the model effectively understand relations among distant and neighboring regions. Due to this reason, MAT is differentiated from other inpainting models that generally use mask-aware and gated convolution-type approaches that cause natural restrictions to look only at limited receptive fields of the surrounding masked regions to fill in painted parts.

In the Mask Aware Transformer (MAT) inpainting model, the process begins by passing a masked image through a convolutional head composed of four layers. This head adjusts the input dimensions and downsamples the image, extracting features, particularly from the unmasked regions, to streamline the data for the transformer. The transformer then operates on tokens to capture long-range dependencies, using shifted windows and dynamic masking to process only valid tokens. Unlike traditional transformers, this version omits layer normalization and relies on feature concatenation instead of residual learning to accommodate incomplete tokens. Afterward, the transformer’s output is passed through a convolutional module to restore the original resolution, followed by a Conv-U-Net for refining details. A Style Manipulation Module (SMM) is incorporated to ensure that the inpainted areas align with the style of the original image.

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 $T_i, i = 1, \dots, 5$, uses an attention module with shifted windows and dynamic mask updating as depicted in Figure 3.2. The attention module uses the Swin-Transformer strategy [25] 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 at each layer is automatically updated 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.

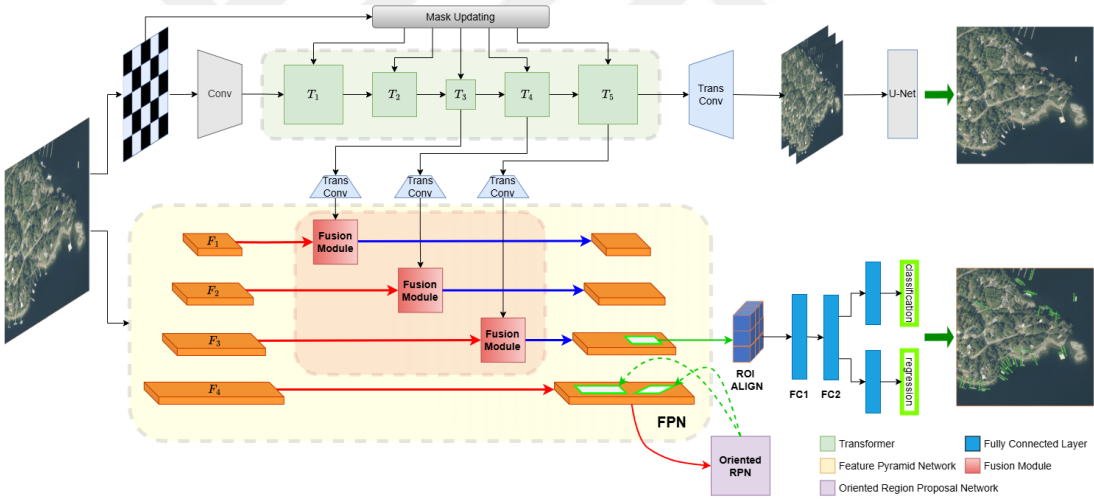


Figure 3.2: Contextual Object Detection via Inpainting (CODI) architecture

3.2.1 Transformer Units

The transformer layers follow an encoder-decoder design with the largest receptive field in the central layer. The outputs of the transformer layers become $X_i^T \in \mathbb{R}^{180 \times H_i^T \times W_i^T}$ 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 $H_5^T \times W_5^T = H/8 \times W/8$. 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. We also present a heat map under each transformer unit experiment in section 5.3 to visualize the behavior of each transformer layer. In this regard, we selected specific transformer units based on each one’s behavior.

3.2.2 Inpainting Setting and Training

An important design decision in adapting the MAT architecture as our contextual backbone is the choice of 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. Also, we do not prefer to use the MAT inpainting model’s random masks because the masked areas’ random strokes can be large and lose some important contextual information, especially for small objects. In addition, the proposed model’s performance will not be consistent due to the uncontrolled environment. Therefore, we designed 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 cell size in the chessboard grid is a hyperparameter that we investigate in the ablation study in Section 5.4. We adjust the cell size in coordination with the number of tokens used in different transformer layers.

We used the split images in the training and inference phase of the inpainting model. One of the main reasons for applying split images for training and inference is that the inpainting model only takes 512×512 low-resolution images. Therefore, resizing the original image to fit it into the inpainting model causes information loss. In other words, low resolution causes poor representative contextual information to fill the masked region under the original scale of satellite images. Another advantage of the split images strategy is limiting the contextual information the inpainting model can process. In general, high-resolution satellite images encapsulate large surfaces on Earth. For that reason, context can change

regionally even in the same image. Lastly, applying the same split images for both FPN levels with transformer unit layers covers a matched earth surface area for processing. Putting full-scale images in the inpainting model, contrary to the object detection model for the second stage of training and inference, causes the same representative contextual information to be used even if the representative features changed in the FPN levels due to using different parts of the same image.

3.3 Overview of the Contextual Detection Mechanism

In this section, we mention the model’s overall detection mechanism. First, we start by splitting images into smaller pieces and sending them to the object detection model. Before the image tensor is processed by the backbone of the object detection model, we send the same preprocessed image, with the applied chess mask over it, to the trained inpainting model to obtain the representative contextual features of the output of the transformer module for the given image. Later, representative contextual information is combined with the representative features of the objects gathered from the Feature Pyramid Network’s (FPN) levels from the object detection model as shown in Figure 3.2. We applied our custom fusion model depicted in Figure 3.3 to inject contextual information with the FPN levels’ object-representative features. Later, we directly continued the output of the fusion model as a new FPN level, which also includes contextual information with object representative features for further processing on the original object detection architecture. So, our fusion module works under a backbone of Oriented RCNN.

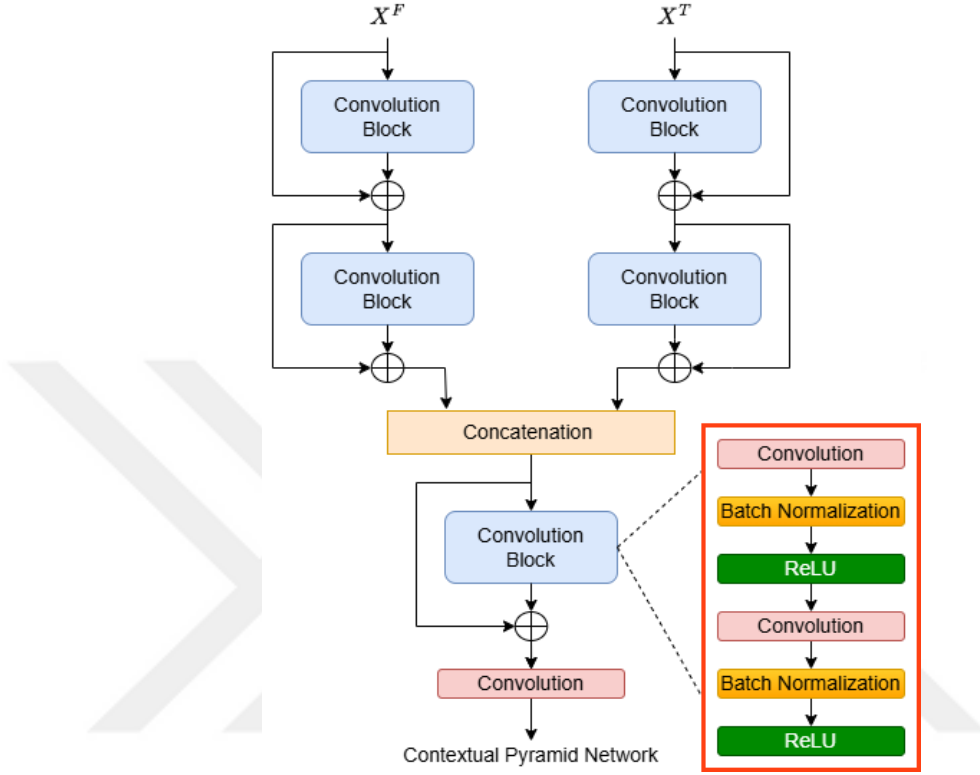


Figure 3.3: Fusion architecture

3.4 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 $H/32 \times W/32$, $H/16 \times W/16$, and $H/8 \times W/8$ pixels, respectively. Since there is element wise feature 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 3.3. The input tensor pair (X^F, X^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 region proposal network (RPN) in the oriented object detector.

Compared to the traditional residual approaches [26] that apply the skip connections before the activation functions, we connect these connections after the activations. Let x be the input to the convolutional block, $f(x)$ be the mapping learned until the ReLU activation, and y be the result of adding the skip connection as

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

In the forward pass, this design ensures that only the activated features from $f(x)$ contribute to the output. This promotes sparsity and achieves an implicit regularization effect. 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

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

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 that is defined as

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

yields the derivative

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

where the gradient passing through the skip connection is adjusted by the derivative of the ReLU activation. Our proposed fusion module provides an unaltered gradient path through the residual connection and helps mitigating vanishing gradient issues. This leads to more stable and efficient training dynamics over the traditional approach.

3.5 Loss Functions

In this section, we talk about losses under two sections: object detection and inpainting model. For both models, we used the default loss functions provided by the original implementations for both object detection and inpainting models.

3.5.1 Object Detection Losses

The loss function that is used for the oriented object detector [5] 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 [5], 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

$$\mathcal{L}_{\text{det}} = \frac{1}{M} \sum_{i=1}^M \mathcal{L}_{\text{cls}}(p_i, p_i^*) + \frac{p_i^*}{M} \sum_{i=1}^M \mathcal{L}_{\text{reg}}(b_i, b_i^*) \quad (3.1)$$

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.

3.5.2 Image Inpainting Losses

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

$$\mathcal{L}_{\text{inp}} = \mathcal{L}_{\text{g}} + \gamma R_1 + \lambda \mathcal{L}_{\text{p}} \quad (3.2)$$

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

3.6 Overall Model Training

In the first stage of the training, images are split into smaller pieces. These pieces are then resized to 512 by 512. Later, A fixed-size chess mask is applied for all the training images, and the MAT inpainting model is trained on these masked images.

In the second stage of training, we operate the MAT inpainting model in inference mode while running both the object detection model and the fusion model in training mode. Similar to the first stage, the original images are divided into smaller parts and then resized for the object detection model, depending on the dataset. For the inpainting model’s inference process, the same split images undergo additional resizing to fit into the inpainting model as in the first stage and are combined with a chess mask to obtain representative contextual information.

Chapter 4

Implementation Details

4.1 Dataset

For this study, we employ DOTA v1[27] and HRSC2016[28], a widely recognized benchmark that supports oriented annotations for remote sensing object detection. We use DOTA v1 for its diversity, high-quality annotations, and multi-class detection support. It includes 15 object classes: 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). DOTA v1 has 188,282 object instances with rotated bounding box annotations in 2806 images with sizes around 4000×4000 pixels, enabling precise localization and orientation handling. Its widespread adoption as a benchmark dataset in research justifies its use in this study.

We prefer to use HRSC2016 for its challenging maritime scenarios and detailed annotations. The dataset comprises high-resolution remote sensing images capturing a wide range of maritime scenes, including coastal areas, harbors, and open seas. There are 1061 images in total, and each image is annotated with

rotated bounding boxes that precisely capture ship orientations and scale variations, ensuring accurate localization.

We split high-resolution images into smaller 1024x1024 chunks for object detection in DOTA. Meanwhile, for HRSC2016, we resized the original images to 800x1333 dimensions without any splitting. These images are directly used in training and inference by the object detection model. We resized images to 512x512 for the inpainting model during training and inference. Consistent with original Oriented RCNN[5] and MAT[4] implementations, we applied similar pre-processing techniques, including data augmentation (horizontal and vertical flipping) and normalization. Additionally, MAT converts grayscale images to RGB format and applies mirroring for further data augmentation.

4.2 The Design Choices

In the preprocessing, we used both single and multi-scale split approaches for the input images. We used the sliding window technique with padding in the image-splitting process. Consistent with [5], we split the images into 1024×1024 pixel tiles with 200-pixel strides for single-scale experiments. The main reason we chose a single scale with 1024 pixel tiles over smaller ones is that, we need to capture contextual information even if we sacrifice the representation of small objects. Another reason is that bigger window sizes have faster training times than smaller window sizes, which are accompanied by smaller strides. In that sense, our chosen hyper-parameters for a single scale are the sweet spot to observe the benefit of injecting context over the Oriented RCNN. 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. Applying multiscale splitting, we capture contextual information on a larger scale and obtain more representative features for small objects on a smaller scale, boosting the performance of our model. Due to hardware limitations, our recall and precision experiments were done in single-scale images.

In the backbone of our proposed model, we only applied ResNet-50 due to similar limitations as in multi-scale. Using the ResNet-50 backbone, we obtained faster training and inference times with a convenient batch size. Moreover, our model’s backbone is deeper due to the integration of the fusion model; we do not want to encounter any training instability, such as a vanishing gradient problem. From our observations, Oriented RCNN with ResNet-50 or ResNet-101 also performs similarly.

In the second training stage of our model, we first applied default object detection preprocessing for the batch of images. Before processing the input in the ResNet backbone part, send the images to the inpainting architecture with additional preprocessing, such as resizing 512 to 512 images with the batch number of chess masks. The main reason for fetching a batch of split images in the ResNet backbone for the inpainting part is the stochasticity of iteration in training. In training and inference, we choose masks with 16 squares for both width and height. Therefore, each square in our mask consists of 32 by 32 pixels, and the resolution of the mask is 512 by 512 pixels. We further discuss the main reason for applying the specific box number as a hyperparameter for the chess mask in section 5.4.3.

4.3 Training Procedure

During the training process for both DOTA and HRSC2016, we utilize Stochastic Gradient Descent (SGD) 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 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 54 epochs on HRSC2016, using a batch size of 8, primarily due to hardware limitations. For the inpainting

training stage, we employ a learning rate of 0.001 with a reduced batch size of 4.

4.4 Evaluation Metrics for Object Detection and Inpainting Model

We preferred to use mean Average Precision (mAP) to evaluate our proposed architecture’s performance due to its wide range of popularity studies under object detection. In addition, the DOTA benchmark dataset prefers to use the mAP evaluation metric for comparison. Another reason for choosing mAP is to observe the performance of our model by accounting for both precision and recall under different score thresholds for detected objects under inference configuration. We used PASCAL VOC_07[29] when we used the validation set as a test set in our DOTA dataset experiments. The DOTA remote server test set evaluation uses PASCAL VOC_12[30]. For the HRSC dataset, we used both PASCAL VOC_07 and PASCAL VOC_12 evaluation protocol. For all our mAP evaluations, we use the polygon Intersection over the Union (IoU) because of working with oriented bounding boxes. The Fréchet inception distance (FID) evaluation metric is used for the inpainting part.

Chapter 5

Experiments

In all experiments, we used the DOTA dataset to demonstrate the impact of context on object detection over the base model. Moreover, DOTA is among the few datasets that provide oriented, multiclass annotations. Additionally, we use the HRSC2016 dataset to demonstrate our model’s generalizability.

5.1 Evaluation in DOTA-v1

The first set of experiments compares CODI with well-known one-stage and two-stage object detection methods on the DOTA dataset. Table 5.1 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 achieves a better performance compared to all of the other detection methods in Table 5.1.

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 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, and storage tanks, derive limited benefit and, under multi-scale inference, may even lose accuracy over the baseline because the added context dilutes discriminative object cues and inflates false positives with respect to the class probability threshold.

Table 5.1: 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.

Method	BBox	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
One-stage																	
SSD[27]	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[27]	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[31]	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[5]	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[32]	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
DHRec[33]	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[34]	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[27]	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[5]	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
RoI Transformer[35]	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[13]	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[36]	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
OSKDet[37]	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[38]	OBB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	76.53
O-RCNN[5] (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[5] (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

5.2 Evaluation in HRSC2016

The second set of experiments compares CODI with other object detection methods on the HRSC2016 dataset. Table 5.2 presents the mAP scores for all methods. 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, which is demonstrated in Figure 5.4. 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. According to Table 5.2, CODI achieves the highest score under the 2007 metric and shows a very competitive performance based on the 2012 metric among all presented models. We also provide visual examples in Figures 5.1 and 5.2, which demonstrate our model’s improved performance over the baseline in terms of recall and precision, respectively.

Table 5.2: 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 also provided.

Method	Backbone	mAP(07)	mAP(12)
Rotated RPN[39]	ResNet-101-FPN	79.08	85.64
RoI Transformer[35]	ResNet-101-FPN	86.20	-
Gliding Vertex[40]	ResNet-101-FPN	88.20	-
PloU[31]	DLA-34	89.20	-
R3Det[41]	ResNet-101-FPN	89.26	96.01
DAL[42]	ResNet-101-FPN	89.77	-
CenterMap-Net[43]	ResNet-50-FPN	89.83	92.10
DRN[32]	H-34	-	92.70
S ² ANet[44]	ResNet-101-FPN	90.17	95.01
OrientedFormer[45]	ResNet-50-FPN	90.17	96.48
DETR-ORD[46]	ResNet-50-FPN	90.21	96.80
Oriented R-CNN[5]	ResNet-50-FPN	90.30	96.52
AOPG[47]	ResNet-50-FPN	90.34	96.22
Oriented DETR[11]	ResNet-50-FPN	90.52	97.73
Our model	ResNet-50-FPN	90.57	96.73

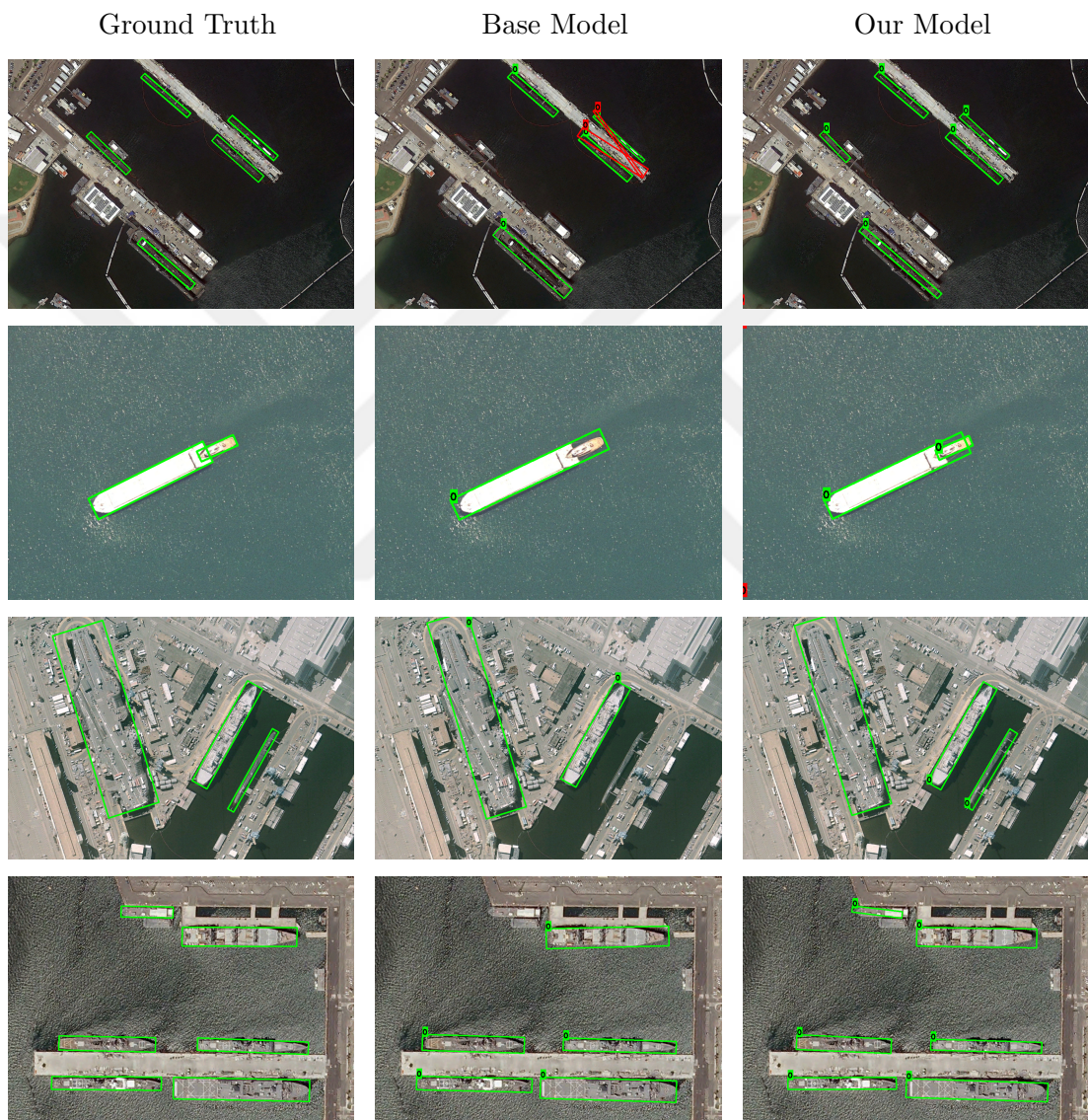


Figure 5.1: Examples of the improved detection performance achieved by the proposed model compared to the baseline in HRSC2016 dataset. At a confidence threshold of 0.01 for the proposed and default setting for the based model, the proposed model identifies more true positives without introducing additional false positives, thereby achieving higher recall while maintaining precision.

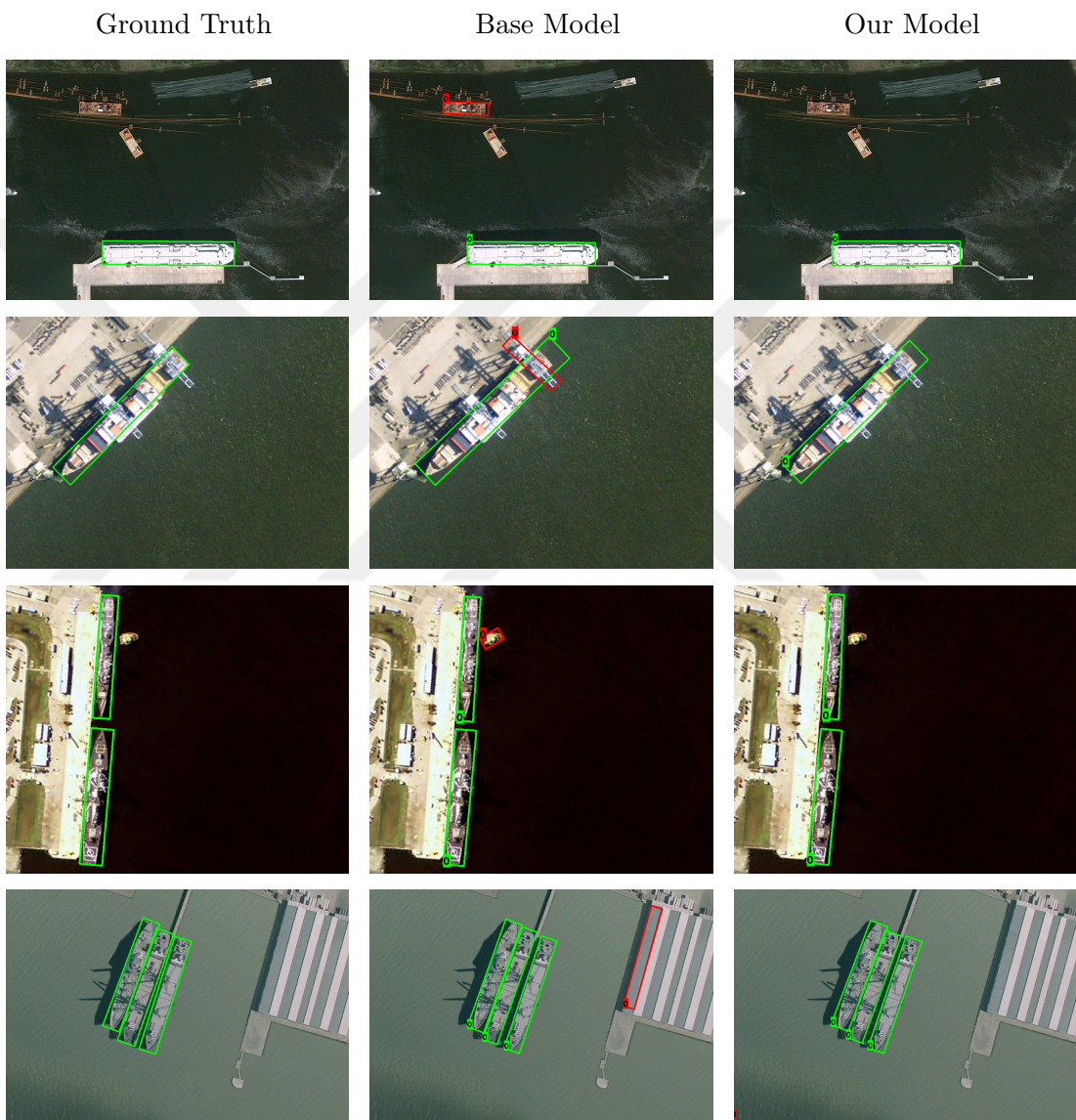


Figure 5.2: Illustrating the precise detection performance of the proposed model against the baseline in HRSC2016 dataset. At a confidence threshold of 0.01 for the proposed and default setting for the based model, the proposed model identifies a higher precision while maintaining recall, effectively reducing false positives without compromising detection sensitivity.

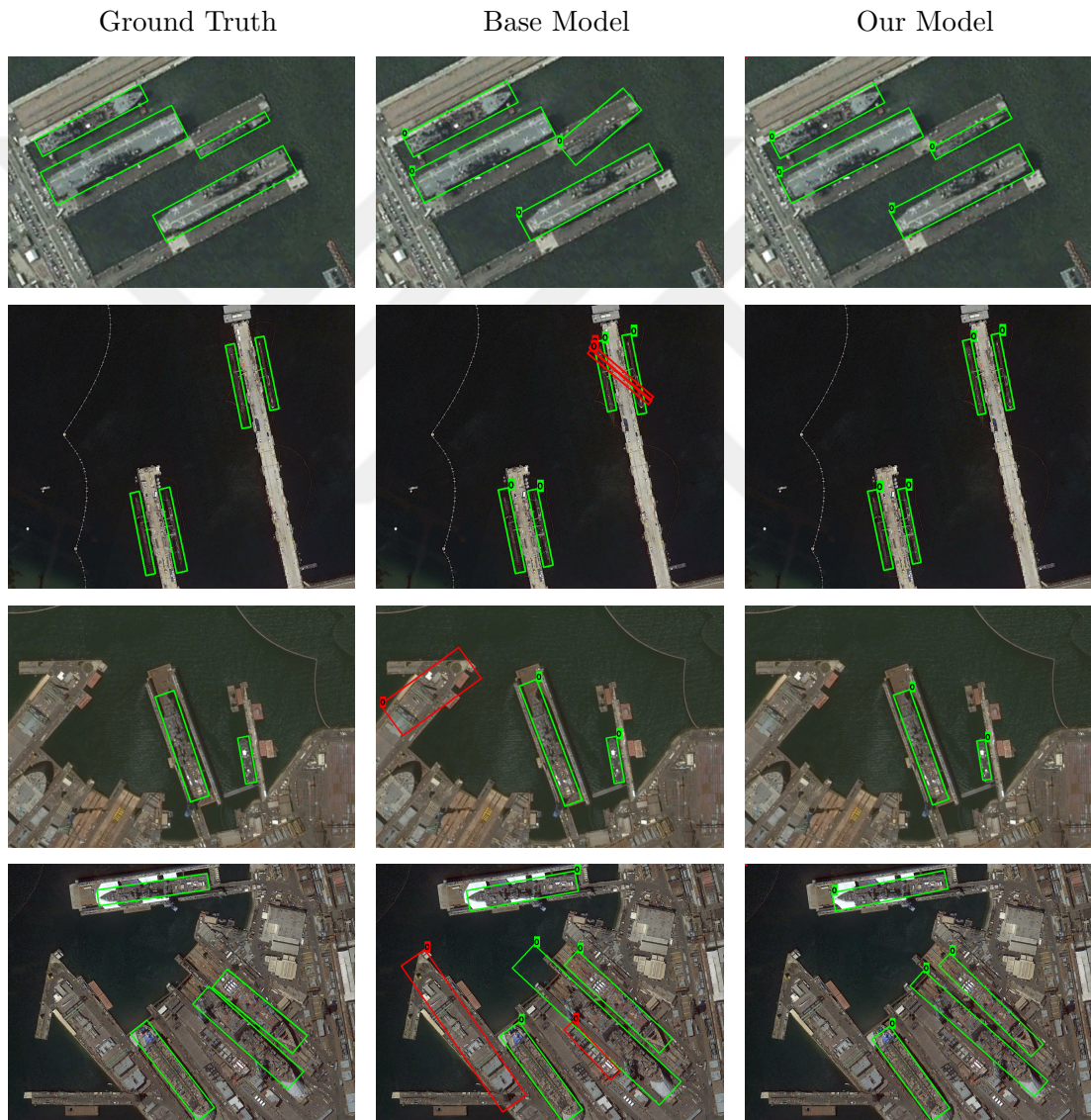
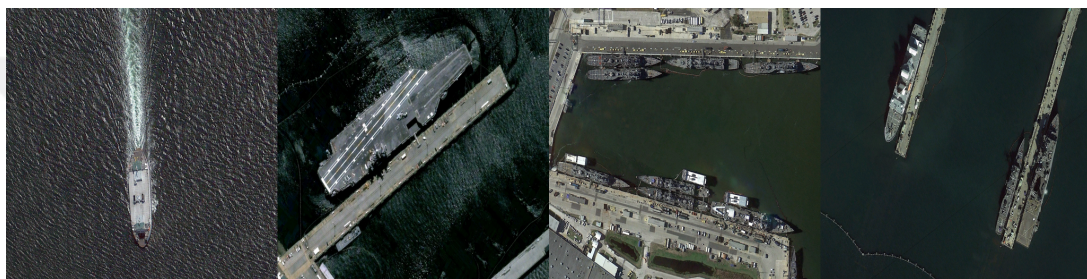
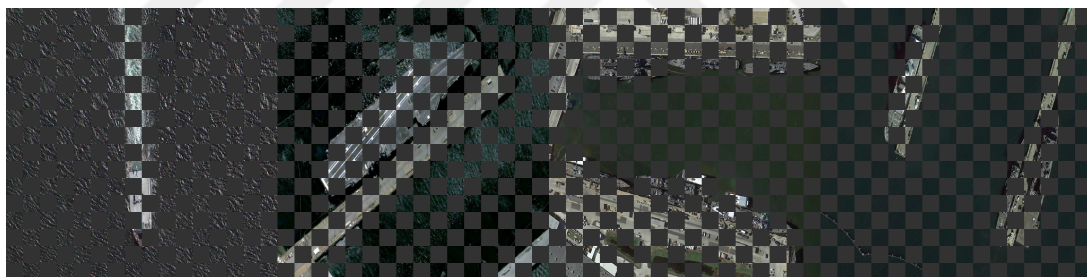


Figure 5.3: Continue of figure 5.2

Real Images



Masked Images



Inpainted Images

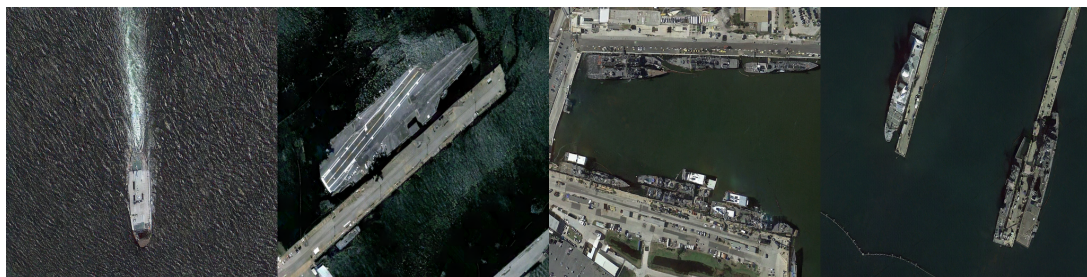


Figure 5.4: Illustrating the HRSC2016 dataset inpainting results obtained from MAT weights trained on the DOTA-v1 dataset applied on Chess Mask 32×32 pixel setting.

5.3 Contextual Pyramid Network Visualization

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 5.5. 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 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.

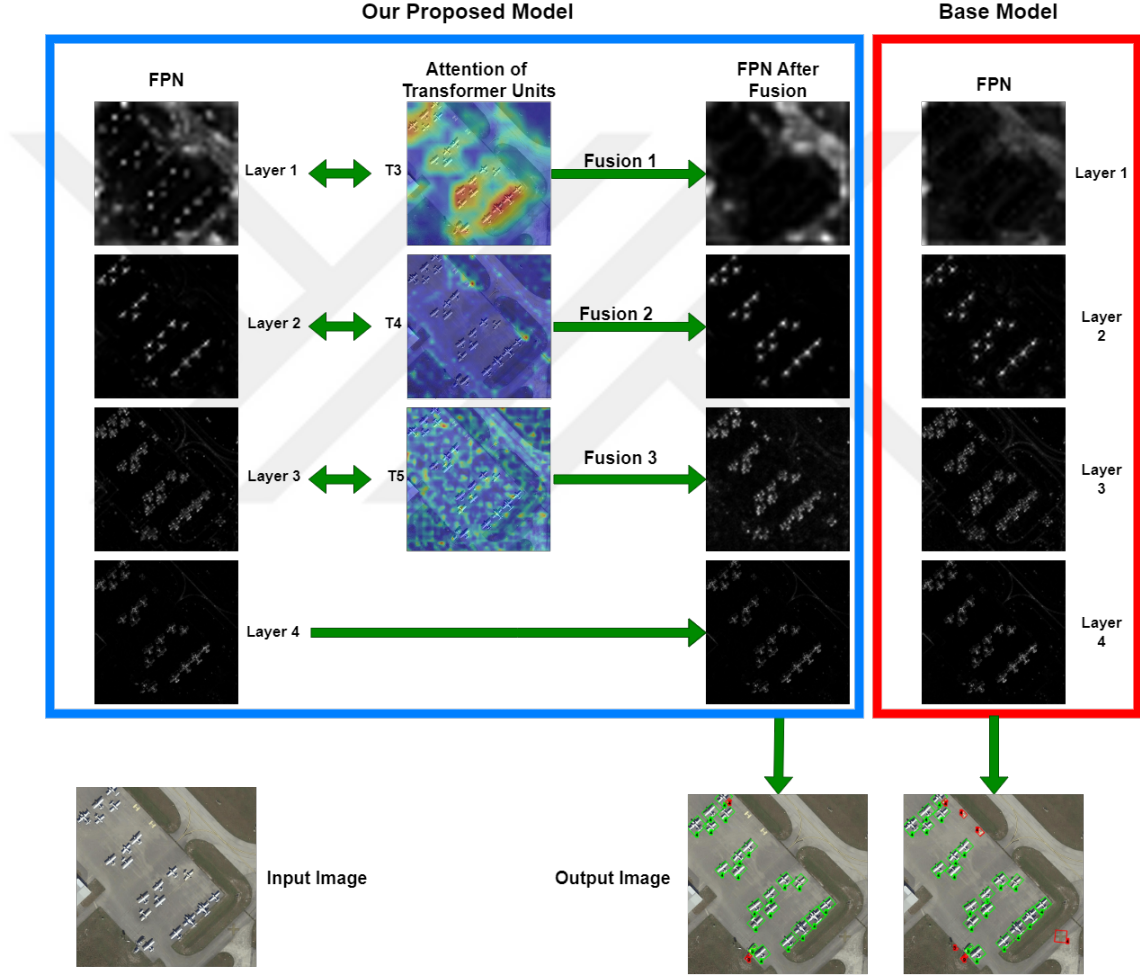


Figure 5.5: Visualization of the feature pyramid network (FPN) levels of the oriented object detector and the attention maps of the transformer layers of the inpainting model, as well as the proposed fusion result as the contextual pyramid network levels. The FPN levels of the baseline-oriented R-CNN model are also shown.

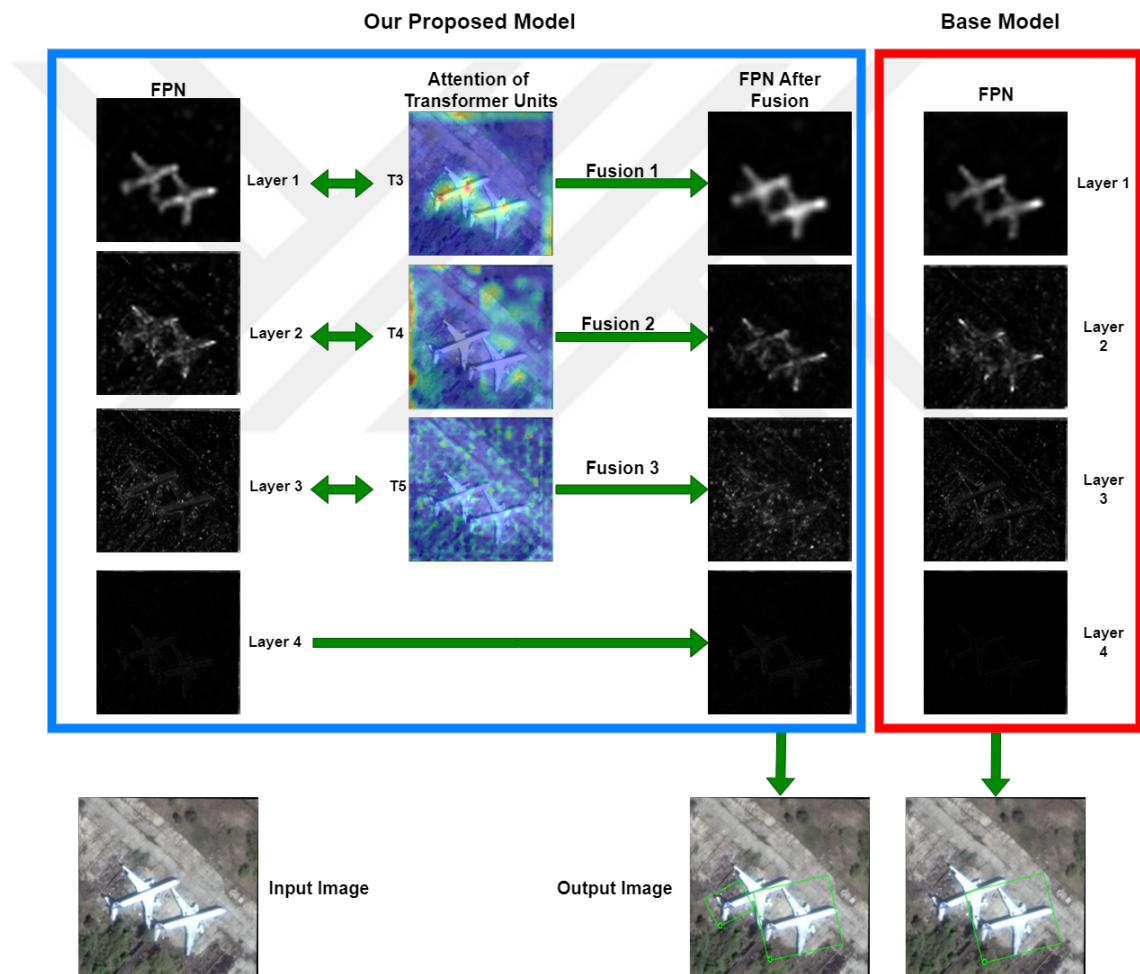


Figure 5.6: Continuation of Figure 5.5.

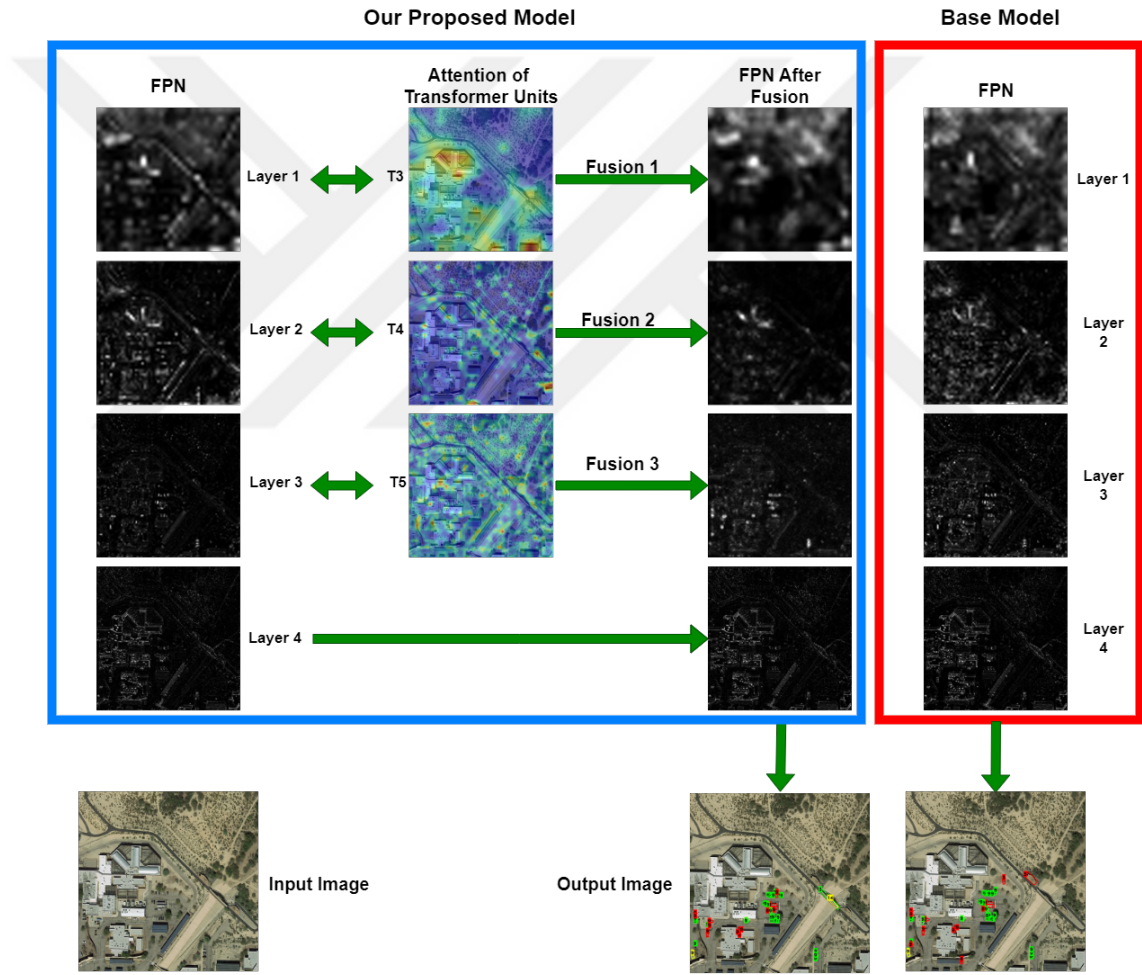


Figure 5.7: Continuation of Figure 5.6.

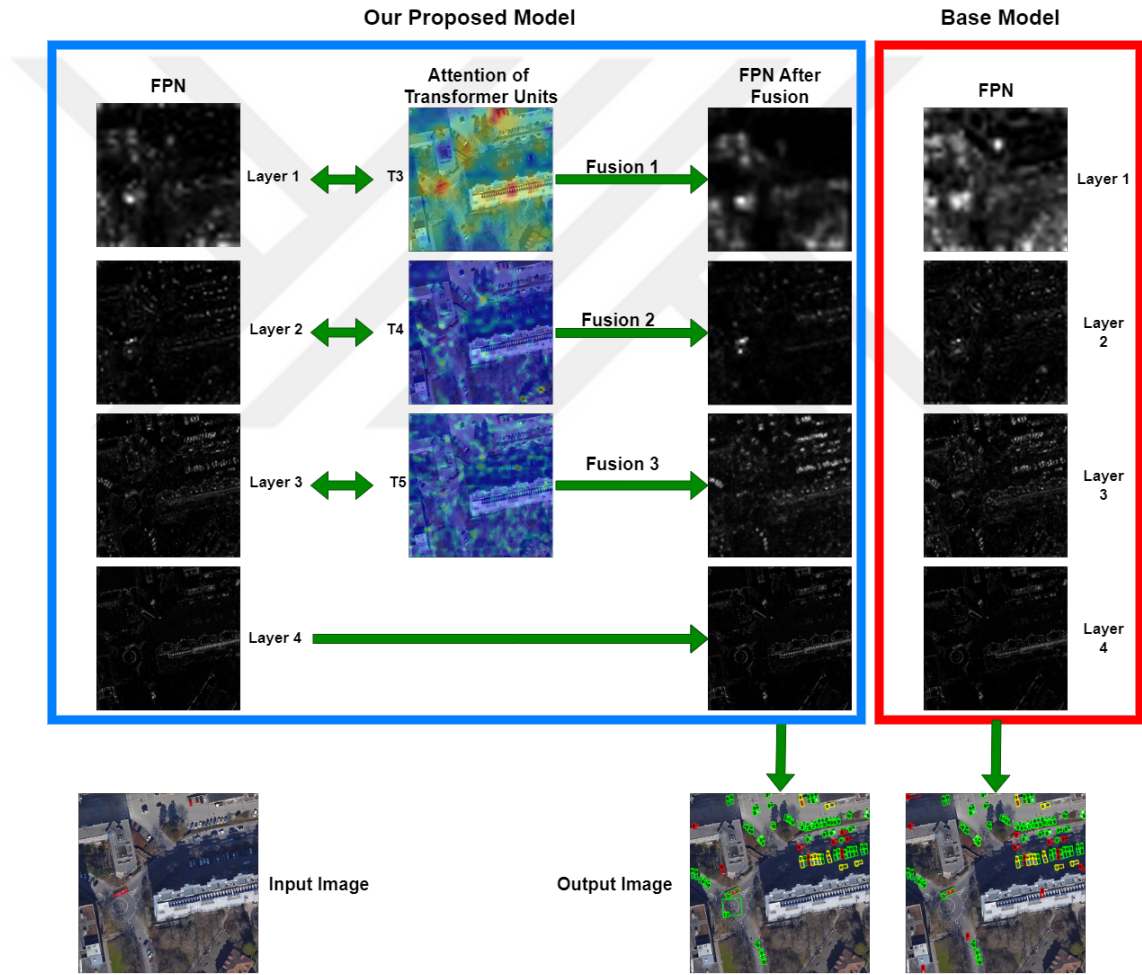


Figure 5.8: Continuation of Figure 5.7.

5.4 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.

5.4.1 Score Threshold Analysis

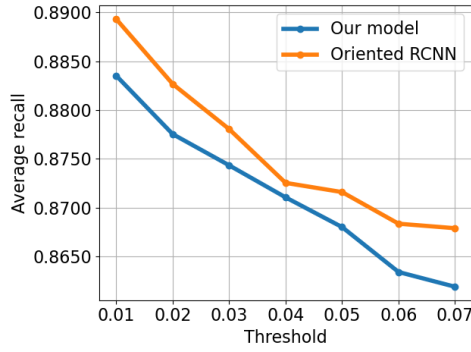
The first set of ablation experiments evaluates the effect of the class probability threshold on different performance metrics during inference. Figure 5.9 shows the number of detected objects and the resulting precision, recall, and mAP values at different threshold levels for both the proposed model 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 in our model’s favor 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.

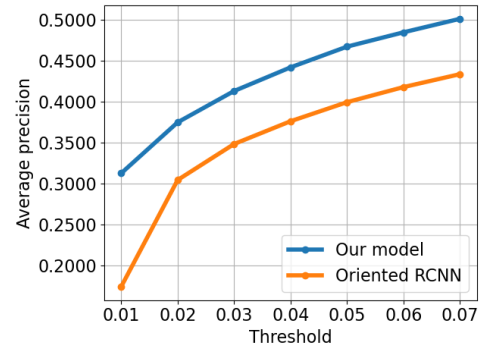
Tables 5.4, 5.3, and 5.5 present the details of the classwise precision, recall, and mAP values at different thresholds, respectively. From the profound classwise analysis of these Tables, we can conclude that lowering the confidence-score threshold reveals a notable difference between CODI and the baseline Oriented RCNN, particularly for classes that share many visual similarities with their surroundings. Specifically, for classes such as Storage-tank, Harbor, Ship, and Plane, CODI outperforms the baseline model due to its superior utilization of contextual information. When the threshold is decreased, CODI gains additional true detections, resulting in increased recall. Although this degradation slightly reduces precision, CODI’s precision remains relatively high compared to the baseline. In contrast, under identical conditions, Oriented RCNN admits a significantly higher number of false positives, causing a sharp deterioration in its precision despite an increase in recall. For rarer or smaller classes, including Helicopter, Bridge, and Soccer-ball-field, CODI already benefited from spatial context effectively at the original threshold 0.05, thus achieving a better precision-recall correlation compared to Oriented RCNN. Lowering the threshold for CODI further recovers a few additional true instances, thereby increasing recall with a moderate decrease in precision. On the contrary, decreasing the threshold for Oriented RCNN yields a small margin of improvements in the overall precision-recall relationship relative to CODI. Overall, integrating semantic contextual information enables CODI to improve recall with only a moderate precision trade-off. Meanwhile, the baseline model achieves higher recall predominantly at the cost of a significant increase in false positives when the confidence-score threshold is lowered.

Qualitative illustration of the detection performance and the improvement in both precision and recall for the proposed model are presented in Figures 5.1, 5.2, 5.10 and 5.11. In Figure 5.10, decreasing the threshold from 0.05 to 0.01 resulted in a similar recall performance with many more false positives that led to a 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 5.11, the recall performances of both

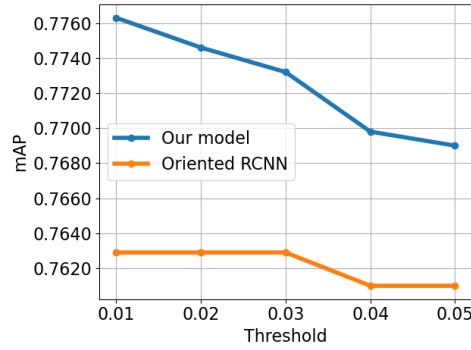
models are similar at a threshold of 0.01, but the proposed model achieves higher precision with fewer false positives. Finally, in Figure 5.1 and 5.2, we see that the proposed model achieves higher recall or precision while preserving precision or recall, respectively, whereas the baseline model has missing or false detections at a default assigned threshold of 0.05. Overall, by calibrating the class probability threshold, we can have effective control over enhancing the recall performance with minimal precision trade-offs while filtering low-scoring predictions prior to non-maximum suppression during inference.



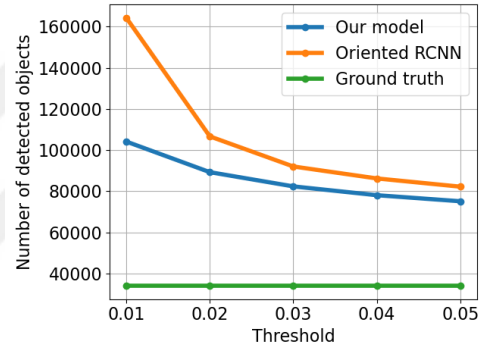
(a)



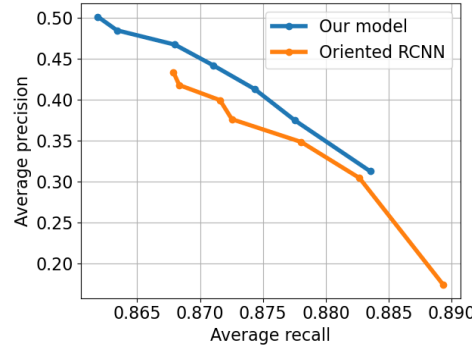
(b)



(c)



(d)



(e)

Figure 5.9: Effect of class probability threshold on detected object count and performance metrics (precision, recall, mAP) for the proposed model (blue) and 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.

Table 5.3: 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.

Model Type	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	Threshold
O-RCNN	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	0.01
Our Model	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	0.01
O-RCNN	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	0.02
Our Model	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	0.02
O-RCNN	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	0.03
Our Model	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	0.03
O-RCNN	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	0.04
Our Model	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	0.04
O-RCNN	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	0.05
Our Model	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	0.05

Table 5.4: 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

Model Type	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	Threshold
O-RCNN	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	0.01
Our Model	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	0.01
O-RCNN	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	0.02
Our Model	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	0.02
O-RCNN	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	0.03
Our Model	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	0.03
O-RCNN	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	0.04
Our Model	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	0.04
O-RCNN	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	0.05
Our Model	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	0.05

Table 5.5: 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.

Model Type	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	Threshold	mAP
O-RCNN	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	0.01	0.7629
Our Model	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	0.01	0.7763
O-RCNN	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	0.02	0.7629
Our Model	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	0.02	0.7746
O-RCNN	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	0.03	0.7629
Our Model	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	0.03	0.7732
O-RCNN	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	0.04	0.7610
Our Model	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	0.04	0.7698
O-RCNN	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	0.05	0.7610
Our Model	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	0.05	0.7690

Table 5.6: Comparison of the different chess mask settings. A lightweight convolution block is applied only to the trans-former units token in the fusion model before channel-wise concatenation with 12 epoch training in total.

Chessmask pixel settings	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
8×8 square grid cells	0.9014	0.7422	0.4673	0.6946	0.7520	0.8223	0.8942	0.9075	0.6736	0.8767	0.7027	0.7210	0.7235	0.7124	0.5409	0.7422
32×32 square grid cells	0.9017	0.7471	0.4706	0.6985	0.7502	0.8235	0.8937	0.9082	0.6851	0.8764	0.7157	0.7281	0.7256	0.7336	0.5648	0.7482
64×64 square grid cells	0.9008	0.7315	0.4870	0.6564	0.7483	0.8217	0.8926	0.9072	0.6303	0.8716	0.7197	0.7032	0.6639	0.7164	0.4910	0.7294

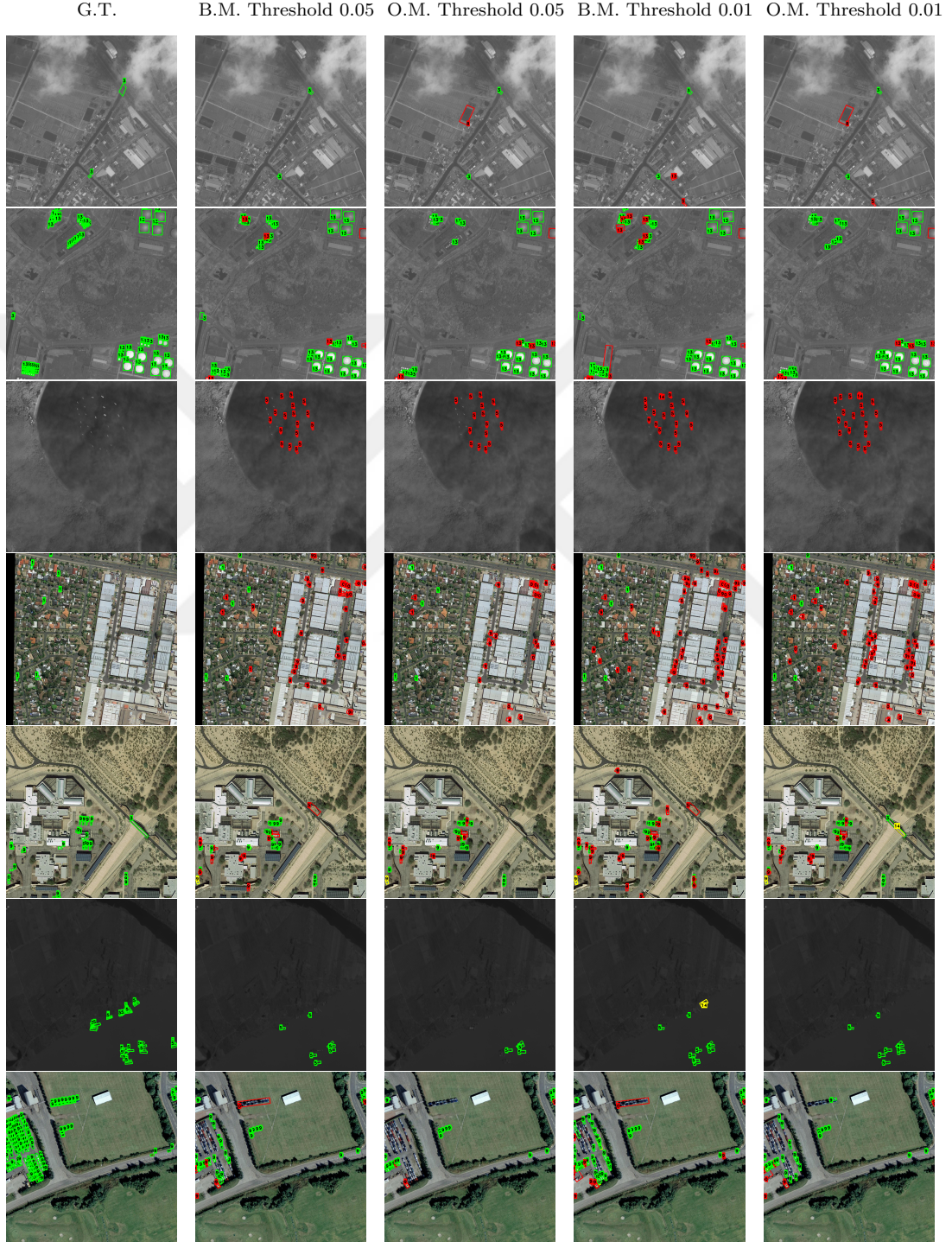


Figure 5.10: Effect of class probability threshold on DOTA detection results. Rows represent different scenes. Colors indicate green (correct), yellow (incorrect class), and red (false positives). Columns (left to right): ground truth; oriented R-CNN baseline at threshold 0.05; proposed model at 0.05; oriented R-CNN baseline at 0.01; proposed model at 0.01. Lowering the threshold increases recall for the proposed model without precision loss, while the baseline shows more false positives without improved recall.



Figure 5.11: Effect of class probability threshold on DOTA detection results. Each row represents a scene. Colors: green (correct), yellow (wrong class), red (false positives). Columns (left to right): ground truth, oriented R-CNN baseline (threshold 0.01), proposed model (threshold 0.01). The proposed model achieves similar recall but improved precision over the baseline.



Figure 5.12: Continue of figure 5.11

5.4.2 Scene Level Prediction Analysis

In the scene-level contextual experiments, our aim is to observe both the base and our model’s prediction behaviors more profoundly. In the first part, we differentiated each image by looking at its object class distribution and stacking the similar distributions under the same cluster. In this regard, a total of 9 clusters shown in Figure 5.13 are appropriate to capture different scenes. When choosing the cluster number, we focused on observing similar classes under the same group of images. In this respect, more than 9 clusters divide the similar class distribution images into two different groups. On the other hand, decreasing the cluster number results in unrelated classes being accommodated under the same scene.

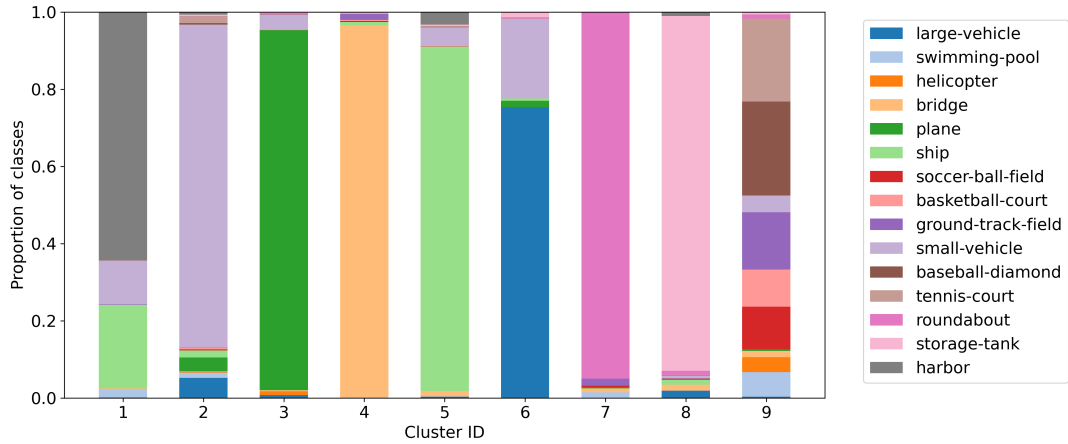


Figure 5.13: Illustrating the distribution of object classes under each cluster, which presents different scenes.

In the second part, we aim to observe the prediction performance of objects’ identities for both the CODI and the base model in different scenes. For this reason, we extract a false positive vector, which includes the number of predictions for the given class, and we have a corresponding ground truth vector obtained from the annotations for each image. Both ground truth and false positive vectors have a length of 15, which is the number of classes provided by the DOTA dataset. If a false positive vector has an object for a given class, but the ground truth annotation does not include that class, then we call it a ”non-contextual

class". However, if both the ground truth vector and the false positive vector have an object for the given class, we call it a "contextual class". To calculate the ratio of contextual and non-contextual classes for different scenes in the given plots in Figure 5.14, first, we calculate the ratio of the total number of predictions in contextual classes over the total number of predictions under the false positive vector in each image. Afterwards, we take the overall cumulative of these ratios of images under the same clusters. In Figure 5.14, our model has better contextual predictions in every scene, even when we increase the threshold to 0.05. Also, when the threshold is 0.01, clusters 1, 3, 4, 5, and 9, those corresponding to the identical cluster ids under the Figure 5.13, have better contextual class identification performance among all clusters. Specified clusters capture scenes that have a strong bond among classes' identities with their backgrounds. Additionally, these scenes have specific background features with limited classes. For instance, we only expect ships and harbors in the blueish background. Therefore, our model effectively uses these semantic contextual clues for better detection under these specific scenes. We can differentiate these scenes under the harbor, airport, open sea, and sports activity. However, scenes that include classes with cluttered and non-distinctive background features have presented a lower gap between base and our model in contextual false positive ratios.

5.4.3 Different Sizes of Chess-masks

The third set of ablation experiments evaluates the effect of the chessboard mask sizes within the inpainting model demonstrated in Figure 5.15. We evaluated the three different chess-mask settings by applying a lightweight fusion model with a shorter period of training session and presented in Table 5.6. The experiments presented earlier use 32×32 pixel square 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 such as a small vehicle presented in Table 5.6, it is ineffective for larger-scale targets. The MAT inpainting model [4] also supports our experiments through evaluations on the

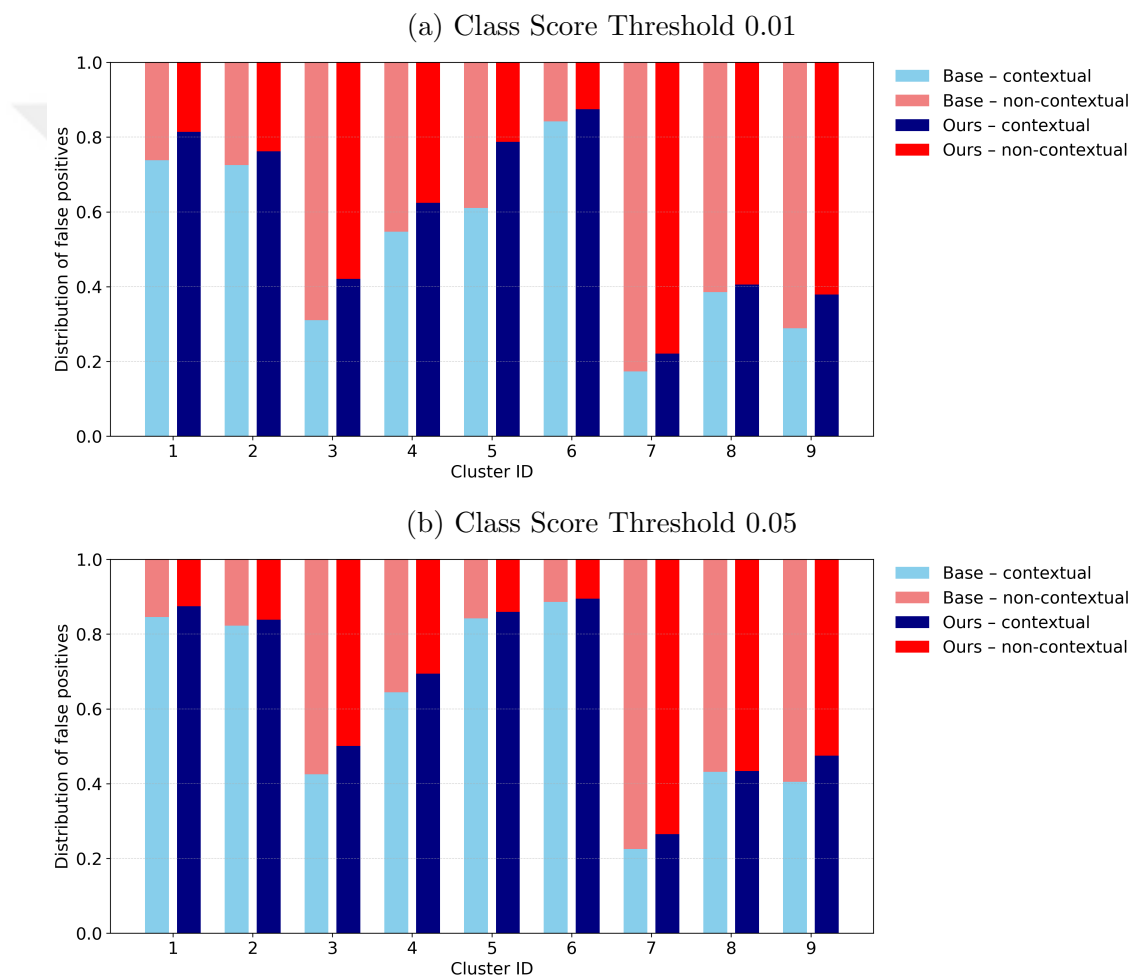


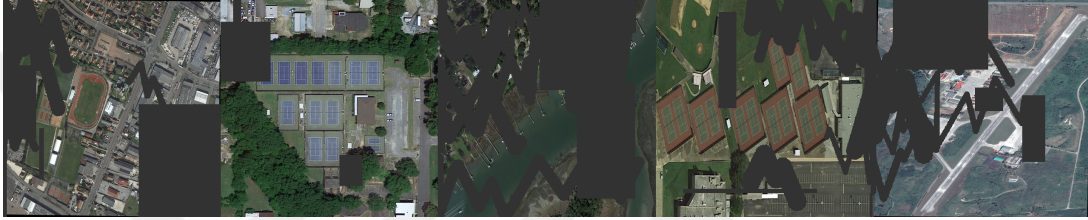
Figure 5.14: Demonstrating the distribution of false positive predictions that belong to contextual or non-contextual classes. These plots show that our model has better prediction performance depending on the scene than the base model.

CelebA-HQ [48] and Places [49] datasets using small and large masks. For both datasets, the FID score, which measures how closely the distribution of features from generated images matches that of original images, is significantly lower for small masks, indicating that the model fills smaller regions more consistently and less aggressively. In other words, lower FID score predictions have better alignment in terms of structural and semantic with the original images. Therefore, the inpainting model cannot capture the local details when the larger 64×64 cells are used for learning the context. 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. Additionally, we conduct further experiments to understand the behavior of the mask at each layer in MAT inpainting, from the convolutional head up to the output of the second transformer unit shown in Figure 5.17. We find that only 256×256 pixel setting grid cell masks are not filled until the second transformer unit. In this regard, transformer units behave like restoration in smaller masked regions, as demonstrated in Figure 5.19. Finally, to better understand the transformer units' role in filling the missing areas in the given image, we applied 1×1 partial convolutions in the convolution head. The further experiments done in Figure 5.21 support our previous experiment that transformer units mainly restore the warped image rather than inpainting the masked areas, and most of the inpainting duty is undertaken by the convolution head with 3×3 partial convolutions.

Original Image



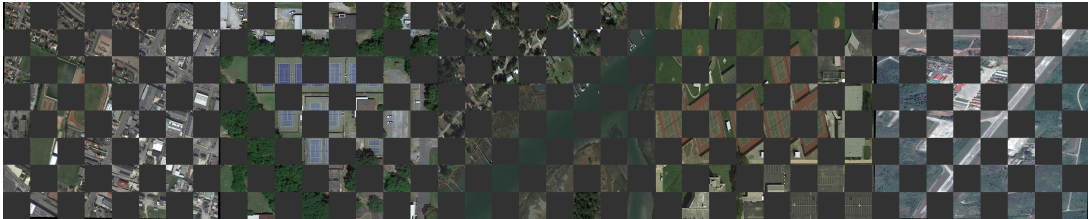
Random Mask



Random Masks Inpainted



Chess Mask 64×64 pixel setting

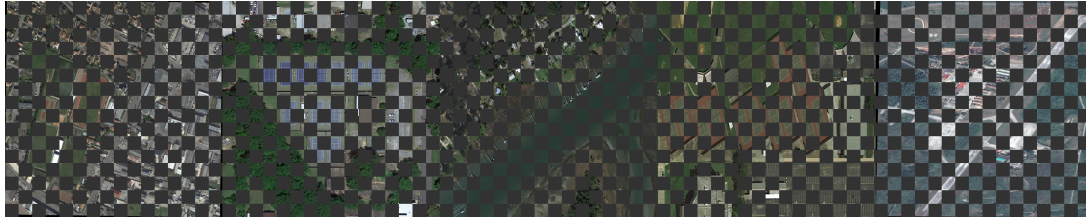


Chess Mask 64×64 pixel setting Inpainted



Figure 5.15: Demonstration of MAT inpainting performance on remote sensing images, showing results from the inpainting model trained and tested with different chessmask settings.

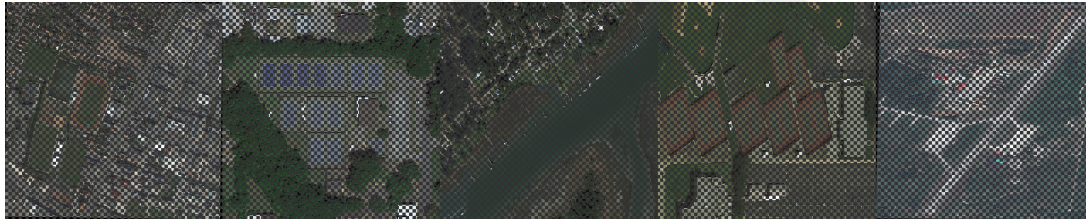
Chess Mask 32×32 pixel setting



Chess Mask 32×32 pixel setting Inpainted



Chess Mask 8×8 pixel setting



Chess Mask 8×8 pixel setting Inpainted



Figure 5.16: Continue of figure 5.15

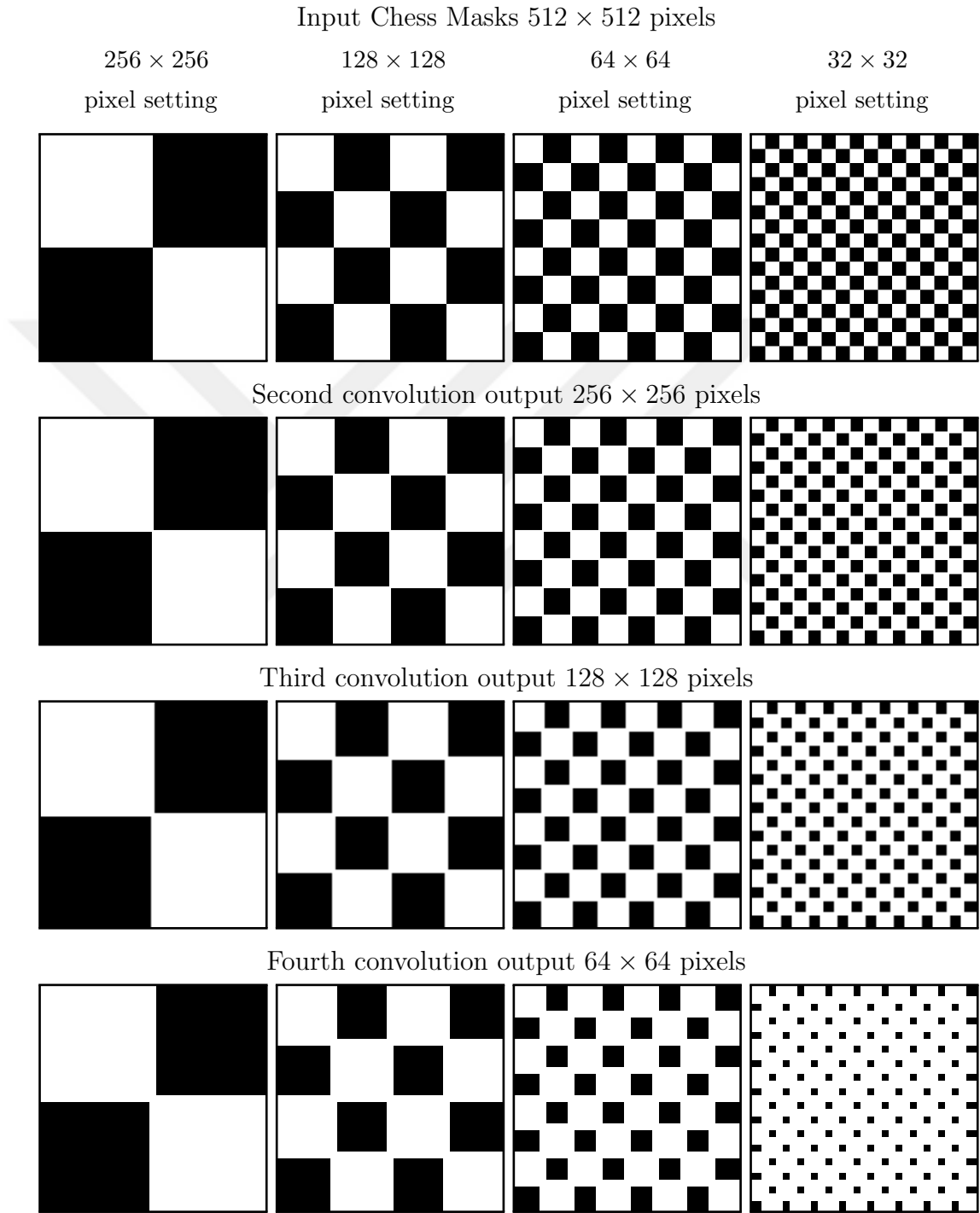
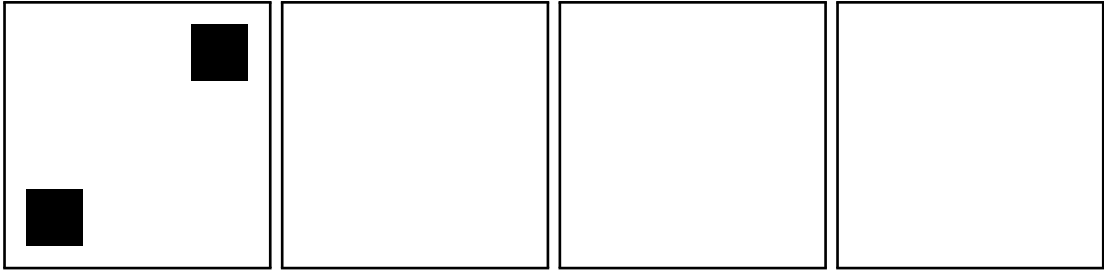


Figure 5.17: Demonstration of the behavior of different mask settings, with outputs observed after the convolution head and transformer unit stages in the MAT architecture.

First transformer unit output 64×64 pixels



Second transformer unit output 32×32 pixels

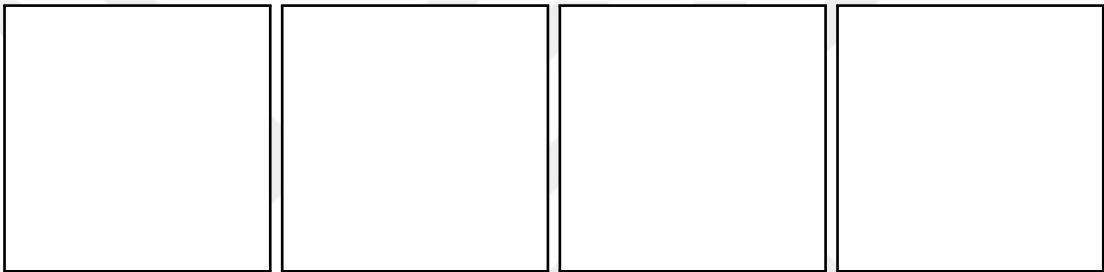


Figure 5.18: Continue of figure 5.17

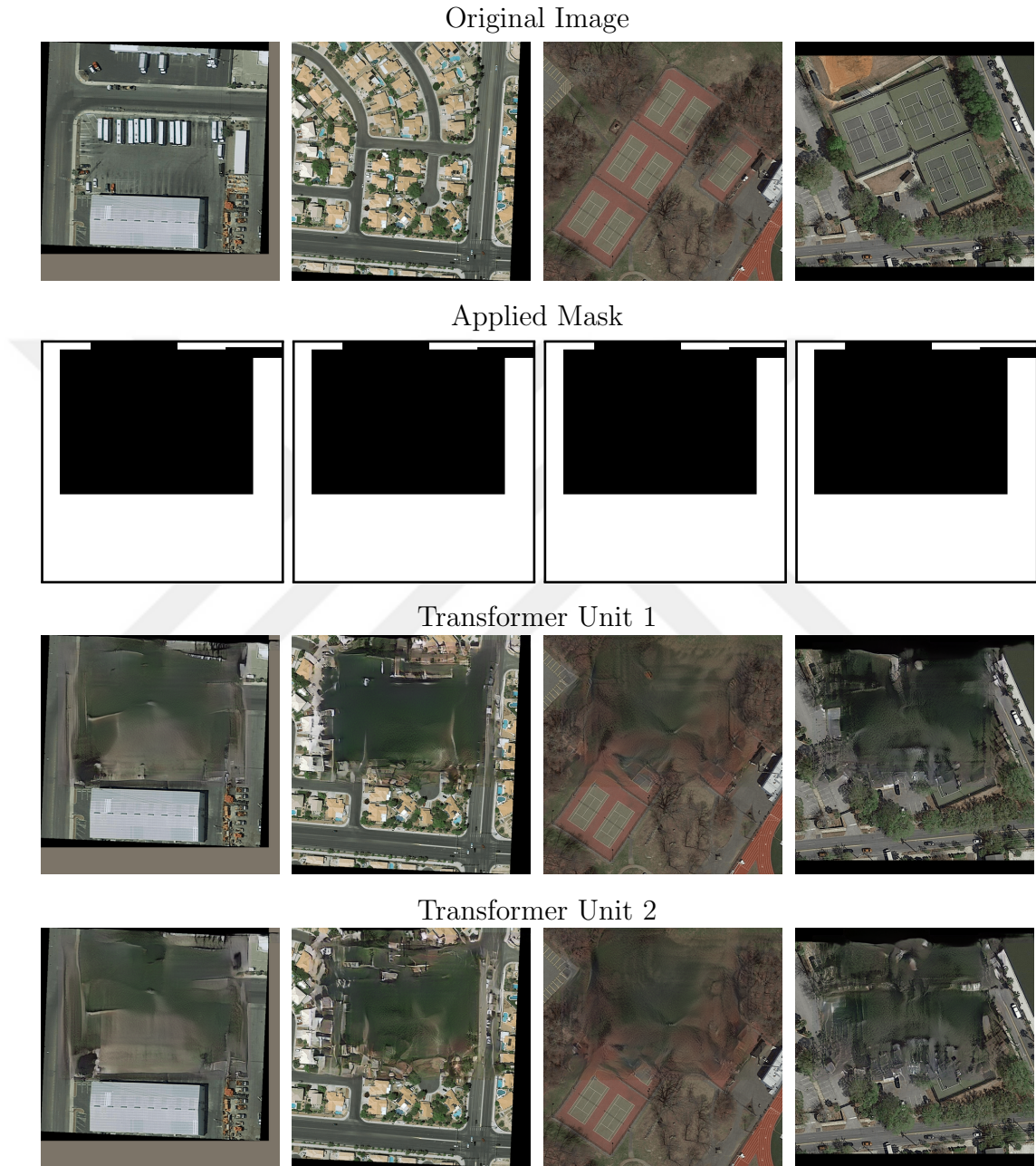


Figure 5.19: Demonstrating each transformer unit’s behavior and individual contribution to the inpainted images. The output of each transformer unit is illustrated by bypassing the remaining transformer units and applying only upsampling before the convolutional reconstruction tail.

Transformer Unit 3



Transformer Unit 4



Transformer Unit 5



Figure 5.20: Continue of figure 5.19

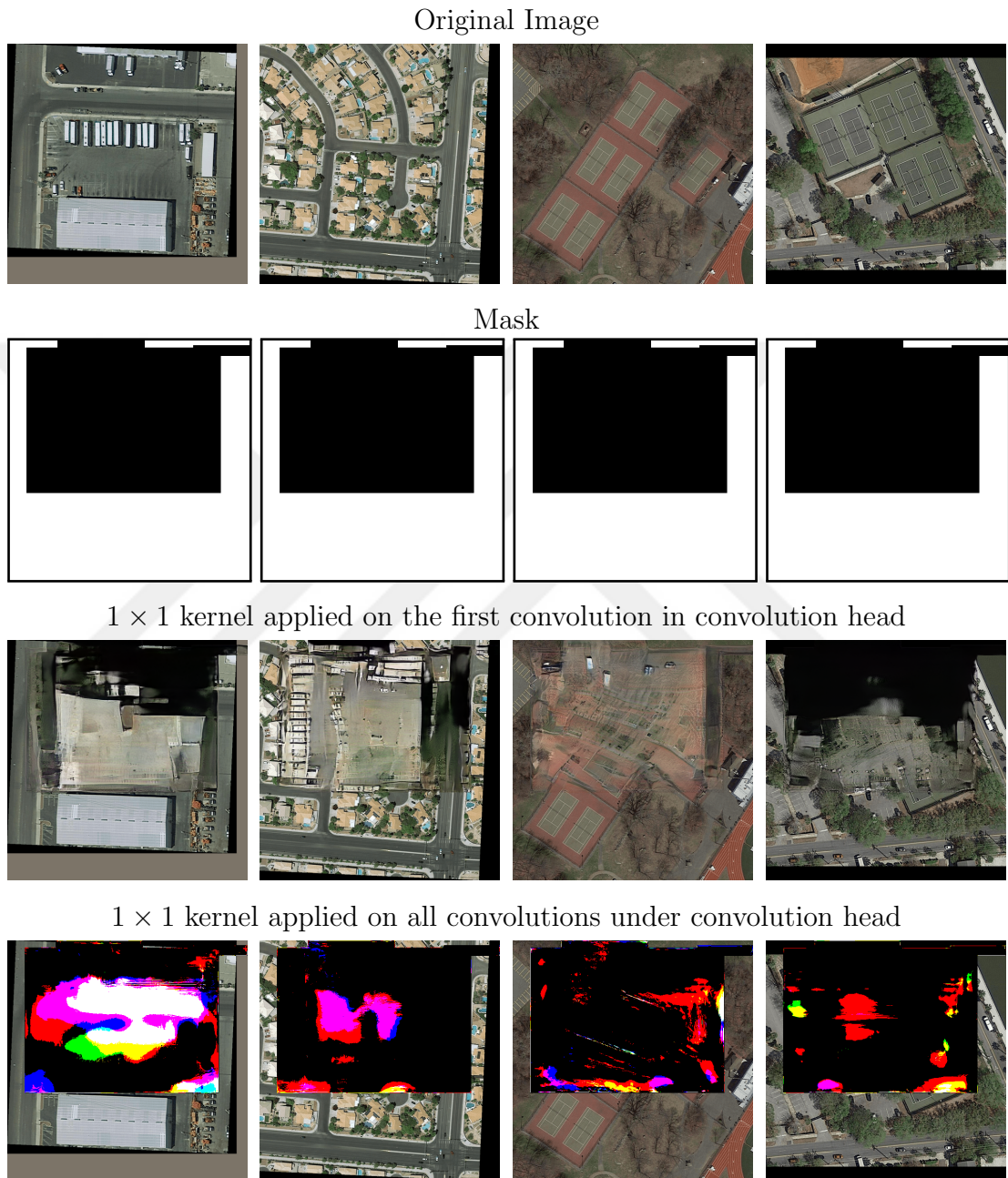


Figure 5.21: Effect of partial convolution kernel size in the convolution head for inpainted images. Here, a 1×1 kernel is used to analyze transformer unit behavior, whereas the original MAT inpainting uses a default 3×3 partial convolution kernel.

5.4.4 Different Design Choices

This section mentions different architectural choices with training strategies that were experimented with and thoroughly presented in Tables 5.7, 5.8, and 5.9.

We comprehensively monitored the training phase of each model before deciding on architectural design choices and hyperparameter tuning. Our training phase monitoring plots in Figure 5.22 show us that the 32×32 pixel setting mask with best fusion model and hyperparameter settings does not show any overfitting signs due to validation set mAP is stabled at approximately %80 after the 14'th epoch. Moreover, we do not see any gradient explosion indication when we monitor the gradient norm graph. Our accuracy graph and average loss curve increase and decrease in a stable way, respectively.

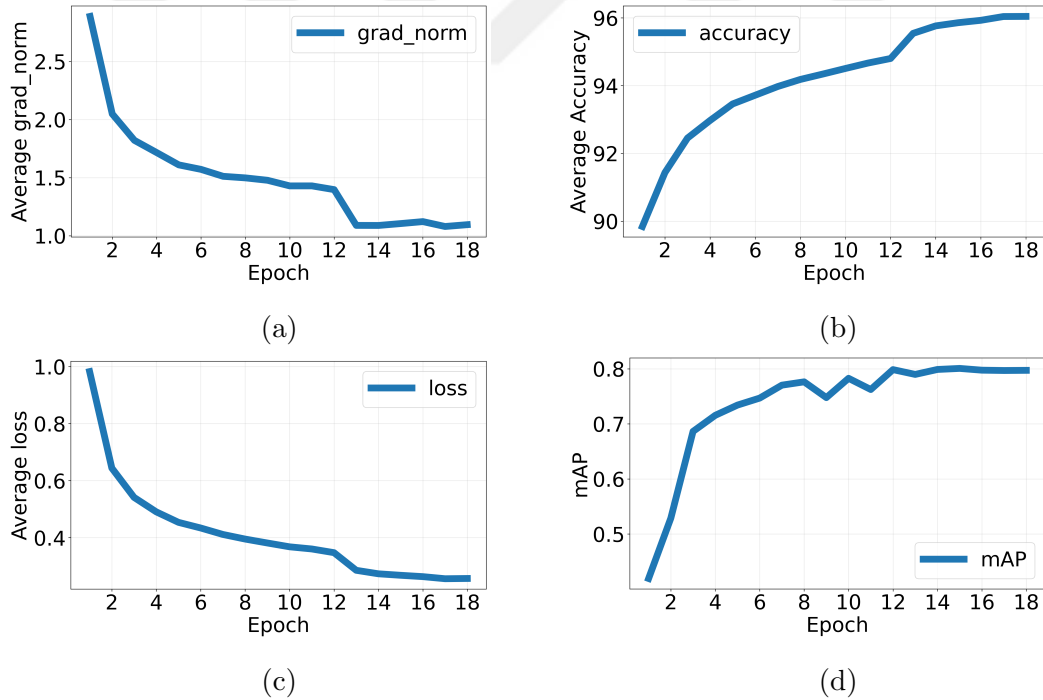


Figure 5.22: Demonstrating our model’s training monitor under 18 epochs. (a) Gradient norm per epoch. (b) Accuracy per epoch. (c) Average loss per epoch. (d) Validation set mAP per epoch. The validation set is evaluated on the randomly chosen subsample of 10% of the original training set.

Initially, our experiments were done in the last transformer unit X_5^T 's output. We applied the representative contextual information of output tokens of X_5^T for all the Feature Pyramid Network's (FPN) levels. However, the results were 6.23% lower than the base model when the DOTA validation set was used as a test set. In that respect, injecting a single transformer unit's output as a context for each FPN level behaves as noise, and only the roundabout average precision score surpasses the base model by a small margin. Therefore, we prefer to use X_3^T , X_4^T , and X_5^T with the corresponding FPN level by their dimensional compatibility.

Our custom fusion model includes too many variables under the chosen activation functions, normalization types, skip connection strategy, convolution layer numbers, and a combination of two different feature maps as presented in table 5.7. We applied the smish [50] activation function presented as 'M33 to M35' in table 5.7. Compared to the lightweight ReLU activation function, Smish is a burden on the memory, causing a decrease in the batch size to half. Even if we applied group norm[51] with gradient accumulation when using smish, we cannot reach the performance of applying ReLU with batch normalization. Similarly, we applied group norm with different gradient accumulation hyperparameters using the ReLU activation function and could not reach the performance we obtained with batch normalization with the ReLU activation function. For the skip connection, we applied the same strategy on the Residual Network[26] paper, and our monitoring under training presents oscillations for gradient normalizations under iterations, which cause unstable training. Because of the caveats, we mentioned profoundly in section 3.4, obtained a dramatically lower score on the skip connection applied before the activation function in models from $M05$ to $M09$ in table 5.7 compared to the base model. We also observed a similar training stability issue under Oriented RCNN training. So, when we train the base model several times, we have a small margin of difference in testing on the DOTA's validation set as a test set. Our novel approach for modifying the skip connection arrangement helps our model to be more robust to oscillations in training. For this reason, we obtained more robust and higher test results with minimal marginal change even after being trained several times.

We applied three different strategies, element-wise addition, channel-wise addition, and cross-attention, to fuse two different tensors obtained from the transformer units and FPN levels. First, we applied element-wise addition. However, element-wise addition injected into the fusion model in *M03* and *M04* shown in table 5.7 could not keep up with any of the channel-wise addition with similar settings. From the scores, we interpret that directly applying element-wise addition with two different feature representation maps can damage each other’s representative features. Therefore, channel-wise addition prevails over this difficulty by not losing any features coming from both tensors and learning harmony between representative object features with the related contextual features under the final convolution block before the tensor compression in channel-wise for use as the corresponding FPN level. We also tried cross-attention before concatenation presented as *M36* in table 5.7 to highlight the important features for both tensors coming from the FPN level’s output and the transformer unit’s output. Because of the GPU limitations, we can only train it under two batches, which performs worse than our current proposed fusion model, with too much overload on the hardware.

Finally, we applied different convolution block numbers before and after concatenation. Our experiments in table 5.7 show us that applying more than two convolution blocks before concatenation and one convolution block after concatenation changes our evaluation results with small margins. So, our proposed fusion model gives the optimum result with a lightweight design.

We applied a greedy search strategy to find out an optimal hyperparameter and training strategies over a fixed fusion model, including a single convolution block before and after the concatenation, presented in table 5.8 and table 5.9. We start by applying the default hyperparameter setting used in the base model[5], dictated as ‘Original’ in table 5.8, and obtain a low score compared to the base model. This is because even though our provided model has more learnable weights, we used the same weight decay and epoch number, causing our model to overfit the dataset and not fully train due to complexity. Therefore, for the later hyperparameter searches, we increase the weight decay to decrease high variation by penalizing the high weight values and also increase the epoch number for our

model to learn better due to more learnable parameters over the base model.

We also tried Adam with stepwise linear learning shown as *v1_1* in table 5.9 rather than stochastic gradient descent with momentum. Even though we applied a twenty times smaller learning rate in Adam, gradients exploded in the training phase. Additionally, rather than applying stepwise learning in stochastic gradient descent with momentum, we applied cosine annealing presented as *v1_2* in 5.9, and even though the training interval is 1.5 times higher than stepwise linear learning, our model cannot obtain a remarkable mAP score in the DOTA dataset.

Table 5.7: Analysis of different fusion models with varying hyperparameters, emphasizing their effect on mAP performance in the DOTA-v1 validation dataset used as a test set.

ID	Hyperparameter type	Lr	Weight decay	Step size	Total epochs	Score thr.	mAP
M01	sgd with linear	0.02	0.00012	[12, 16]	18	0.02	77.34
M02	sgd with linear	0.02	0.00012	[11, 15]	16	0.03	76.91
M03	sgd with linear	0.02	0.00012	[11, 15]	16	0.03	76.42
M04	sgd with linear	0.02	0.00012	[11, 15]	16	0.02	76.90
M05	sgd with linear	0.02	0.00012	[11, 15]	16	0.05	76.18
M06	sgd with linear	0.02	0.00012	[12, 16]	18	0.03	76.47
M07	sgd with linear	0.02	0.00012	[14, 19]	20	0.02	76.04
M08	sgd with linear	0.02	0.00013	[14, 19]	20	0.01	76.45
M09	sgd with linear	0.02	0.00014	[14, 19]	20	0.02	76.54
M10	sgd with linear	0.02	0.00012	[11, 15]	16	0.05	75.31
M11	sgd with linear	0.02	0.00012	[12, 16]	18	0.03	76.89
M12	sgd with linear	0.02	0.00013	[12, 16]	18	0.01	76.59
M13	sgd with linear	0.02	0.00013	[12, 16]	18	0.01	77.63
M14	sgd with linear	0.02	0.00012	[12, 16]	18	0.01	77.26
M15	sgd with linear	0.02	0.00014	[12, 16]	18	0.01	76.31
M16	sgd with linear	0.02	0.00013	[14, 19]	20	0.01	76.78
M17	sgd with linear	0.02	0.00013	[12, 16]	18	0.01	76.67
M18	sgd with linear	0.02	0.00014	[12, 16]	18	0.01	76.95
M19	sgd with linear	0.02	0.00015	[12, 16]	18	0.02	76.79
M20	sgd with linear	0.02	0.00014	[13, 18]	20	0.01	77.00
M21	sgd with linear	0.02	0.00013	[12, 16]	18	0.01	77.13
M22	sgd with linear	0.02	0.00014	[12, 16]	18	0.01	76.84
M23	sgd with linear	0.02	0.00014	[13, 18]	20	0.01	76.88
M24	sgd with linear	0.015	0.00015	[14, 19]	20	0.02	77.21
M25	sgd with linear	0.02	0.000135	[12, 16]	18	0.02	76.56
M26	sgd with linear	0.02	0.00012	[12, 16]	18	0.01	77.13
M27	sgd with linear	0.02	0.00013	[13, 18]	20	0.02	77.41
M28	sgd with linear	0.15	0.00013	[14, 19]	20	0.00	77.75
M29	sgd with linear	0.15	0.00013	[13, 18]	20	0.05	76.88
M30	sgd with linear	0.01	0.00013	[14, 19]	20	0.01	76.45
M31	sgd with linear	0.015	0.00013	[15, 20]	22	0.02	77.04
M32	sgd with linear	0.01	0.00013	[12, 16]	18	0.01	77.46
M33	sgd with linear	0.02	0.00013	[12, 16]	18	0.01	76.95
M34	sgd with linear	0.02	0.00013	[12, 16]	18	0.01	76.59
M35	sgd with linear	0.02	0.00013	[12, 16]	18	0.01	76.06
M36	sgd with linear	0.005	0.00013	[12, 16]	18	0.01	76.63

Model-ID: **M01** – Bridge_v1 (classic_concat); **M02** – Bridge (added ReLU to transcovs forward function); **M03** – Elementwise_simple (1 conv for elementwise addition); **M04** – Elementwise_complex (1 conv block before and after elementwise addition); **M05** – Channel-wise concat, skip before ReLU, Hyperparam v8; **M06** – Channel-wise concat, skip before ReLU, Hyperparam v9; **M07** – Channel-wise concat, skip before ReLU, Hyperparam v11; **M08** – Channel-wise concat, 2 conv block before concat, skip before ReLU, Hyperparam v11; **M09** – Channel-wise concat, 2 conv block before & after concat, skip before ReLU, Hyperparam v11; **M10** – Channel-wise concat *without* skip; **M11–M12** – Traditional channel-wise concat (skip after ReLU, same channels) with two weight-decay settings; **M13–M16** – Traditional concat, 2 conv block before concat (four weight-decay / epoch combos); **M17–M20** – Traditional concat, 3 conv block before concat (four variants); **M21–M31** – Traditional concat, 2 conv block before *and* after concat (eleven hyper-parameter sweeps); **M32** – Best model (M13) + Smish; **M33–M35** – Best model (M13) + Smish + gradient accumulation (2/4/8 iters); **M36** – Best model (M13) + cross-attention module.

Table 5.8: Analysis of different hyperparameter tuning applied to the same fusion architecture, including a single convolutional block before and after the concatenation step, emphasizing their effect on mAP performance in the DOTA-v1 validation dataset used as a test set.

Model version	Hyperparameter type	Lr	Weight decay	Step size	Total epoch	Score thr.	mAP
Original	sgd with linear	0.02	0.0001	[8, 11]	12	0.05	75.99
v1	sgd with linear	0.02	0.0001	[12,24]	29	0.05	76.13
v2	sgd with linear	0.01	0.0005	[12, 17]	18	0.05	74.20
v3	sgd with linear	0.02	0.0005	[12, 16]	18	0.05	73.48
v4	sgd with linear	0.02	0.0001	[10, 14]	16	0.05	76.18
v5	sgd with linear	0.02	0.00012	[9, 13]	14	0.05	76.56
v6	sgd with linear	0.02	0.00011	[9, 13]	14	0.05	76.47
v6	sgd with linear	0.02	0.00011	[9, 13]	14	0.02	76.81
v6	sgd with linear	0.02	0.00011	[9, 13]	14	0.01	76.75
v7	sgd with linear	0.02	0.00012	[9, 15]	16	0.05	76.49
v8	sgd with linear	0.02	0.00012	[11, 15]	16	0.05	76.84
v9	sgd with linear	0.02	0.00012	[12, 16]	18	0.05	76.37
v9	sgd with linear	0.02	0.00012	[12, 16]	18	0.02	77.34
v9	sgd with linear	0.02	0.00012	[12, 16]	18	0.03	77.36
v9	sgd with linear	0.02	0.00012	[12, 16]	18	0.04	77.17
v10	sgd with linear	0.02	0.00013	[12, 16]	18	0.05	76.72
v10	sgd with linear	0.02	0.00013	[12, 16]	18	0.01	77.34
v10	sgd with linear	0.02	0.00013	[12, 16]	18	0.02	77.31
v10	sgd with linear	0.02	0.00013	[12, 16]	18	0.03	77.27

Table 5.9: Analysis of different learning strategies applied to the same fusion architecture, including a single convolutional block before and after the concatenation step, emphasizing their effect on mAP performance in the DOTA-v1 validation dataset used as a test set.

Model version	Hyperparameter type	Lr	Weight decay	Step size	Total epoch	Score thr.	mAP
v1_1	Adam with linear	0.001	0.0005	[12, 17]	20	0.05	grad explode
v1_2	Sgd with cosineannealing	0.02	0.0001	-	27	0.05	74.74

5.5 Discussion

In this study, we experimented the effect of injecting semantic contextual information obtained from the inpainting model into the FPN layers of the object detection architecture. In this respect, in both the DOTA and HRSC2016 datasets, our model outperforms the base model in terms of mAP score in both the Pascal VOC2007 and Pascal VOC2012 protocols. Further observations indicate that classes whose identities are braced by their specific environments perform better than the base model in all our experiments related to the DOTA evaluations. However, our model failed to identify classes that appear in isolation or visually cluttered backgrounds. In other words, applying semantic contextual information behaves like noise in these classes and degrades their performance. As a result, our model’s best use case occurred in classes; those identities have been strongly supported by their distinctive environment. We also provided our model’s Contextual Pyramid Network visualization to present how our model pronounced objects’ representative features in corresponding FPN layers compared to the base model. We also encountered the effect of this representative feature enrichment in precision and recall, so our model surpasses marginally in the other metric when we equalize the precision or recall. Experiments done in transformer units show us that these units mainly restore the warped image by looking at both short and long-range dependencies among different parts of the image. In this respect, chess mask experiments indicate that applying a specific mask size is a sweet spot for capturing both local and global contextual information for restoration. We have also done experiments with different context injection techniques. The related experiments suggest that using channel-wise concatenation with a specific number of convolution blocks gives the most effective efficiency with a minimum performance trade-off. Even though we designed a lightweight fusion architecture, another limitation of our model is inference time due to the bottleneck of the inpainting model.

Chapter 6

Conclusion

We introduced Contextual Object Detection via Image Inpainting (CODI) architecture for accurate detection even in challenging remote sensing tasks. As far as we know, this is the first time that inpainting has been effectively used directly in the object detection. We used the Feature Pyramid Network (FPN) layers of the object detection model to obtain representative features of the objects. We obtained the semantic contextual information for the image by benefiting from the transformer units of the inpainting model, which learns the relations for both short and long-range dependencies among different parts of the warped image for restoration. We proposed a custom fusion module that learns how to inject the contextual information into the object detection’s related FPN layer. Outputs of the fusion modules give new FPN layers, which we call these new layers as Contextual Pyramid Network (CPN). CPN layers have richer object representative features than former FPN layers due to the support of the semantic context. For this reason, the Region Proposal Network (RPN) performs better localization and labeling accuracy over the objects. The extensive experiments in DOTA and HRSC2016 datasets demonstrated that our approach outperforms the base model, which is one of the best-performing models for detecting objects in aerial images, by effectively using contextual clues to detect objects correctly in the most challenging environments. Specifically, the model effectively uses the semantic contextual information for mostly the classes whose identities are

reinforced by distinctive environments. In addition, our model surpasses the base model in either precision or recall when fixing the other parameter. These outcomes highlight the potential of object detection by applying contextual information via inpainting for the most complex remote sensing tasks. For future work, incorporating diffusion networks could be a viable approach for extracting contextual information in conjunction with an object detection model.



Bibliography

- [1] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” 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 conference on Computer Vision and Pattern Recognition*, pp. 779–788, 2016.
- [3] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017.
- [4] 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*, pp. 10758–10768, 2022.
- [5] 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*, pp. 3520–3529, 2021.
- [6] 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*, pp. 6589–6600, 2023.
- [7] 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*, 2024.
- [8] 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.
 - [9] 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, pp. 6782–6790, 2024.
 - [10] Y. Li, Q. Hou, Z. Zheng, M.-M. Cheng, J. Yang, and X. Li, “Large selective kernel network for remote sensing object detection,” *arXiv preprint arXiv:2303.09030*, 2023.
 - [11] Z. Zhao, Q. Xue, Y. He, Y. Bai, X. Wei, and Y. Gong, “Projecting points to axes: Oriented object detection via point-axis representation,” *arXiv preprint arXiv:2407.08489*, 2024.
 - [12] 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.
 - [13] 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. 10015–10024, 2019.
 - [14] 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.
 - [15] 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.

- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] M. Zhang, Q. Liu, and Y. Wang, “Ctxmim: Context-enhanced masked image modeling for remote sensing image understanding,” *arXiv preprint arXiv:2310.00022*, 2023.
- [23] 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*, pp. 541–557, Springer, 2020.
- [24] 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*, p. 1–10, IEEE, Oct. 2019.

- [25] 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.
- [26] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
- [27] 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 Conference on Computer Vision and Pattern Recognition*, pp. 3974–3983, 2018.
- [28] 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, pp. 324–331, SciTePress, 2017.
- [29] 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.
- [30] 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.
- [31] 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 *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pp. 195–211, Springer, 2020.
- [32] 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*, pp. 11207–11216, 2020.

- [33] 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.
- [34] 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, pp. 923–932, 2022.
- [35] 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*, pp. 2849–2858, 2019.
- [36] 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*, pp. 150–165, Springer, 2018.
- [37] D. Lu, “Oskdet: Towards orientation-sensitive keypoint localization for rotated object detection,” *CoRR*, vol. abs/2104.08697, 2021.
- [38] 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*, pp. 16912–16922, 2024.
- [39] 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.
- [40] 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.
- [41] 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, pp. 3163–3171, 2021.

- [42] 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, pp. 2355–2363, 2021.
- [43] 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.
- [44] 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, 2022.
- [45] 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*, vol. 62, pp. 1–16, 2024.
- [46] 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.
- [47] 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.
- [48] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of gans for improved quality, stability, and variation,” 2018.
- [49] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million image database for scene recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, 2018.
- [50] X. Wang, H. Ren, and A. Wang, “Smish: A novel activation function for deep learning methods,” *Electronics*, vol. 11, no. 4, p. 540, 2022.
- [51] Y. Wu and K. He, “Group normalization,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.