DEFORMABLE CONVOLUTION NETWORK AND DEFORMABLE ATTENTION IN YOLO11 FOR ORIENTED SMALL OBJECT DETECTION

A MASTER'S THESIS

Submitted to
Graduate School of Electrical Engineering

By

REYNALDHI TRYANA GRAHA

201012420030

In partial fulfillment of the requirements
for the Degree of Master of Engineering

TELKOM UNIVERSITY
BANDUNG
2025

APPROVAL PAGE

MASTER'S THESIS

DEFORMABLE CONVOLUTION NETWORK AND DEFORMABLE ATTENTION IN YOLO11 FOR ORIENTED SMALL OBJECT DETECTION

by

REYNALDHI TRYANA GRAHA
201012420030

Approved and authorized to fulfil one of the requirements of
Program of Master of Electrical-Telecommunication Engineering
School of Electrical Engineering
Telkom University
Bandung

Bandung, 2nd December, 2025

Supervisor	Co-Supervisor
Dr. Koredianto Usman, S.T., M.Sc.	**Suryo Adhi Wibowo, S.T., M.T., Ph.D.**
NIP. 02750053	NIP. 1087003

SELF DECLARATION AGAINST PLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct. I have full cited and referenced all materials and results that are not original to this work.

2nd December, 2025
REYNALDHI TRYANA GRAHA

Signature:_____

ABSTRACT

Object detection in aerial imagery faces critical challenges, primarily due to the prevalence of small, densely packed targets and arbitrary orientations that defy the fixed geometric assumptions of standard Convolutional Neural Networks (CNNs). This research proposes an enhanced single-stage detector, YOLO11-DCN-DA, designed to overcome these limitations by integrating adaptive feature extraction mechanisms. To address geometric rigidity, the standard C3k2 modules in the backbone and neck are replaced with C3k2_DCN blocks incorporating Modulated Deformable Convolution (DCNv2), enabling the network to dynamically adjust its receptive field to align with rotated objects. Additionally, to mitigate feature scarcity in small targets, a Deformable Attention block is introduced following the Spatial Pyramid Pooling - Fast (SPPF) module, allowing the model to sparsely attend to relevant key points while suppressing background clutter. The proposed framework is evaluated on the challenging DOTA and SODA-A datasets, focusing on Oriented Bounding Box (OBB) tasks. The study hypothesizes that this synergistic combination will yield superior detection accuracy (mAP) for small and oriented objects compared to the baseline YOLO11, while maintaining a viable computational cost (FLOPs) for practical deployment.

Keywords: Deformable Convolution, Deformable Attention, Oriented Bounding Box, Small Object Detection, YOLO11.

ACKNOWLEDGEMENTS

This thesis is compiled with the effort, help, and support from all supporting elements. The author would like to express the deepest gratitude and thanks to:

1. Allah SWT, for all the love, guidance and forgiveness in every mistake that the author has ever done and Rasulullah SAW, as role model who inspire writer in living life and trying to be better.

2. My beloved parents for the endless love, prayer, support, and motivation in finishing this thesis. The author could not imagine to be who he is today without their presence.

3. My thesis supervisors, Dr. Koredianto Usman, S.T., M.Sc. and Suryo Adhi Wibowo, S.T., M.T., Ph.D., for the invaluable guidance, advice, and support throughout the research and writing process of this thesis.

PREFACE

Alhamdu lillahi rabbil 'alamin, praise to Allah, the most gracious, the most merciful, with the mercy and guidance, the author has successfully finished this thesis with the title of "**DEFORMABLE CONVOLUTION NETWORK AND DEFORMABLE ATTENTION IN YOLO11 FOR ORIENTED SMALL OBJECT DETECTION**". The author compiled this thesis to be filled in the graduation requirements in Program of Master of Electrical-Telecommunication Engineering, School of Electrical Engineering, Telkom University.

The suggestions for improving this thesis are highly appreciated. Hopefully, this thesis is expected to be improved and provided contributions for the reader and Indonesia especially for education and research of telecommunication on the future.

Bandung, 2nd December, 2025

REYNALDHI TRYANA GRAHA

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

AI-TOD	:	AI-Task Oriented Dataset
AP	:	Average Precision
CNN	:	Convolutional Neural Network
CSP	:	Cross Stage Partial
DCN	:	Deformable Convolutional Network
DETR	:	DEtection TRansformer
DOTA	:	Dataset for Object Detection in Aerial Images
FFN	:	Feed-Forward Network
FLOPs	:	Floating Point Operations
FPN	:	Feature Pyramid Network
HBB	:	Horizontal Bounding Box
IoU	:	Intersection over Union
LayerNorm	:	Layer Normalization
mAP	:	Mean Average Precision
MHSA	:	Multi-Head Self-Attention
MS COCO	:	Microsoft Common Objects in Context
MSDeformAttn	:	Multi-Scale Deformable Attention
OBB	:	Oriented Bounding Box
PANet	:	Path Aggregation Network
RF	:	Receptive Field
SNR	:	Signal-to-Noise Ratio
SODA	:	Small Object Detection dAtaset
SPIE	:	Society of Photo-Optical Instrumentation Engineers
SPPF	:	Spatial Pyramid Pooling - Fast
SSD	:	Single Shot MultiBox Detector
ViT	:	Vision Transformer
WP	:	Work Package
YOLO	:	You Only Look Once

LIST OF SYMBOL

A_{mqk}	Attention weight for the sampled location (m, q, k)
AP_i	Average Precision for class i
B	Batch size
B_i	Number of biases in layer i
C	Number of channels in the feature map
d_k	Dimension of the key vector
Δm_n	Modulation scalar weighting each deformed sample
Δp_{mqk}	Attention offset for k^{th} key in head m
Δp_n	Deformable offset added to the sampling grid
F_i	Number of floating-point operations in layer i
FN	False Negatives
FP	False Positives
$G(q, p)$	Bilinear interpolation kernel
H	Height of the feature map
K	Number of sampling keys per attention head
L	Number of layers
M	Number of attention heads in the deformable module
N	Number of classes (in mAP context)
p_q	Reference point used by Deformable Attention for each query
Q	Query matrix in attention mechanism
$\mathcal{R}$	Regular sampling grid of a convolution kernel
S	Stride of the feature map
TP	True Positives
V	Value matrix in attention mechanism
W	Width of the feature map
W_i	Number of weights in layer i
W_m	Output projection matrix of attention head m
W'_m	Value projection matrix of attention head m
$w(p_n)$	Learned convolution weight at offset p_n
$x(p)$	Feature value sampled at position p (possibly fractional)
$y(p_0)$	Output feature at spatial location p_0
z_q	Content feature of the query element q

CHAPTER 1
INTRODUCTION

1.1 Background

Object detection is a fundamental task in computer vision with numerous applications across various domains, including medical imaging [1], autonomous driving [2], security surveillance [3], and aerial imaging [4]. The primary objective of object detection is to identify and localize objects within an image or video frame by predicting bounding boxes and class labels for each detected object. Over the years, significant advancements have been driven by the development of deep learning algorithms, specifically Convolutional Neural Networks (CNNs). These advancements have led into two main categories of object detection methods: two-stage detectors and single-stage detectors. Two-stage detectors, such as Faster R-CNN, prioritize accuracy by first generating region proposals before classification [5]. In contrast, single-stage detectors, like You Only Look Once (YOLO) and Single Shot MultiBox Detector (SSD), optimize for speed by directly predicting bounding boxes and class probabilities in a single pass [6, 7]. The YOLO family, in particular, has undergone rapid evolution leading to the recent YOLO11, which offers a superior balance between inference speed and detection accuracy [8].

Despite the impressive performance of single-stage detectors on general tasks, they frequently encounter difficulties when detecting small objects. Small objects typically defined as those occupying a minimal number of pixels present significant challenges due to limited visual information, susceptibility to occlusion, and background clutter [9]. This issue is particularly critical in aerial imagery analysis, where targets such as vehicles, buildings, and ships appear at varying scales [10]. The primary challenges include the loss of critical features during CNN downsampling, the extreme class imbalance between small and large objects, and the difficulty in distinguishing small objects from noise. To mitigate these issues, researchers have explored techniques such as multi-scale feature fusion, context-aware modeling, and specialized data augmentation [11]. Additionally, architectural enhancements like Feature Pyramid Networks (FPNs) and attention mechanisms have shown promise in recovering details necessary for small object detection [12].

Beyond scale, another critical aspect in aerial object detection is orientation. Unlike traditional detection methods that utilize axis-aligned Horizontal Bounding

Boxes (HBB), oriented object detection employs Oriented Bounding Boxes (OBB) or polygons to capture objects more precisely. This capability is essential for aerial imagery, where objects often appear with arbitrary rotations due to the top-down perspective of the imaging sensor [10]. Standard HBBs often introduce excessive background noise when enclosing rotated objects, confusing the classifier. Consequently, oriented detection methods incorporate angle regression, rotation-invariant feature extraction, and specialized loss functions to effectively handle these geometric variations [13].

However, standard CNNs backbone of most modern detectors including YOLO possess inherent limitations when simultaneously addressing orientation and small scale. Standard convolution operations sample the input feature map using a fixed, regular grid. This rigid geometric structure lacks invariance to large geometric transformations such as rotation, scaling, or deformation [14]. Consequently, when a small object is rotated or appears in a non-standard pose, the fixed receptive field of a standard convolution may fail to cover the object of interest effectively [15, 16]. This limitation is particularly detrimental for small oriented objects, where semantic information is already scarce, and any misalignment in feature extraction can lead to detection failures [17].

To overcome these geometric and spatial limitations, advanced mechanisms such as Deformable Convolution Networks (DCN) and Attention Mechanisms have been proposed. Deformable convolution introduces learnable offsets to the regular sampling grid, allowing the receptive field to adaptively deform and align with the object's actual shape and orientation [14, 18]. Furthermore, attention mechanisms, specifically Deformable Attention, enable the network to focus dynamically on the most relevant features while suppressing irrelevant background information [19]. By adjusting the importance of different spatial locations, these mechanisms enhance the representation of small objects that might otherwise be overwhelmed by background clutter [15].

Therefore, this research proposes the integration of Deformable Convolution and Deformable Attention mechanisms into the YOLO11 architecture to enhance feature extraction capabilities for oriented small object detection. By leveraging the adaptive sampling of deformable convolution and the context-aware focusing of deformable attention, the proposed method aims to address the geometric variations and feature scarcity inherent in aerial imagery. This thesis explores the synergistic effect of these components within the YOLO11 framework, aiming to achieve superior detection performance compared to existing state-of-the-art methods on challenging aerial and remote sensing datasets.

1.2 Problem Identification

The primary problem addressed in this research is the main two challenges of detecting objects that are both tiny in scale and arbitrarily oriented within aerial imagery, which modern detectors struggle to handle due to the fixed geometric structure of standard Convolutional Neural Networks (CNNs).

Specifically, the technical problems are identified as follows:

1. **Geometric Rigidity of Standard Convolution:** Standard convolution operations rely on kernels with a fixed geometric shape. This rigid structure inherently restricts the network's ability to model complex geometric transformations. When objects in aerial imagery appear at arbitrary angles, the fixed receptive field of the convolution kernel fails to adapt, preventing the network from effectively capturing the features of rotated objects.

2. **Feature Loss in Low-Resolution Targets:** Small objects in aerial imagery typically possess low resolution and limited feature information. As these images pass through the downsampling layers of standard CNN architectures, the already scarce spatial details are often lost or 'washed out'. Consequently, the detector fails to preserve the fine-grained information necessary to recognize these very small targets in deeper network layers.

3. **Interference from Background Noise:** Due to their small scale and limited features, small objects are difficult to distinguish from environmental background noise. Standard feature extraction mechanisms often lack the ability to focus exclusively on the object of interest, causing the weak feature signals of small targets to be overwhelmed by stronger signals from complex backgrounds, leading to missed detections.

4. **Inefficiency of Horizontal Bounding Boxes:** Traditional object detection relies on Horizontal Bounding Boxes (HBB). For oriented objects, HBBs are inefficient because they inevitably capture excessive background information along with the object. This inclusion of irrelevant background noise within the object proposal confuses the classification process and hinders the precise localization required for aerial targets.

1.3 Research Objective

The primary objective of this research is to develop an enhanced hybrid single-stage object detection framework based on YOLO11, specifically optimized for the

two main challenges of oriented small object detection. This is achieved by addressing the geometric rigidity of standard convolutions and the lack of context-aware focusing in current baselines through the integration of adaptive feature extraction mechanisms.

The specific objectives of this research are detailed as follows:

1. **Integrate Deformable Convolution Modules for Geometric Adaptation:** Strategically modify the standard convolutional layers within the YOLO11 architecture by integrating Deformable Convolution modules. This modification aims to enable the network to adaptively adjust its receptive field based on the scale and orientation of target objects. By learning offset values for the sampling grid, the model will capture the geometric features of rotated objects more effectively than the fixed-grid convolutions used in the baseline model, directly addressing the issue of geometric rigidity.

2. **Implement Deformable Attention for Robust Feature Fusion:** Enhancing the feature fusion process by incorporating Deformable Attention mechanisms. Unlike standard attention modules which often treat spatial locations uniformly, this mechanism is designed to focus computational resources sparsely on the most informative key points of small objects. This objective seeks to demonstrate that dynamic, sparse attention can effectively suppress the background clutter and noise inherent in aerial imagery, therefore preserving the weak feature signals of small targets.

3. **Design and Validate a Hybrid Architecture While Balancing Accuracy and Speed:** Design and validate a cohesive hybrid architecture that synergizes Deformable Convolution and Deformable Attention to optimize Oriented Bounding Box (OBB) task. This objective involves benchmarking the proposed framework against state-of-the-art methods on large-scale aerial datasets (DOTA and SODA) to demonstrate a superior balance between detection accuracy (mAP) and computational efficiency (FLOPs), ensuring the model remains practical for real-world deployment.

1.4 Research Method

This research proposes a novel single-stage object detection framework that enhances the YOLO11 architecture by integrating two advanced adaptive feature extraction mechanisms: Deformable Convolutional Networks (DCN) and Deformable

Attention. These methods are selected to specifically address the two challenges of geometric variations in oriented objects and feature scarcity in small objects.

1.4.1 Deformable Convolutional Networks (DCN)

Standard Convolutional Neural Networks (CNNs) are inherently limited in modeling geometric transformations due to the fixed geometric structures of their building modules. A standard convolution unit samples the input feature map at fixed locations (e.g., a regular 3×3 grid) and pools features with a static receptive field. This rigidity is suboptimal for oriented object detection, where targets may appear with arbitrary rotation, scaling, or deformation.

To overcome this limitation, this research employs Deformable Convolution, as introduced by Dai et al. [14]. The core idea is to augment the spatial sampling locations in the convolution modules with learnable offsets.

In a standard 2D convolution, the output feature map y at a specific location p_0 is computed as:

$$y(p_0) = \sum_{p_n \in \mathcal{R}} w(p_n) \cdot x(p_0 + p_n) \tag{1.1}$$

where $\mathcal{R}$ defines the regular sampling grid (e.g., $\mathcal{R} = \{(-1,-1),(-1,0),\ldots,(1,1)\}$ for a 3×3 kernel) and $w(p_n)$ represents the weights.

In Deformable Convolution, the regular grid $\mathcal{R}$ is augmented with offsets $\{\Delta p_n | n = 1,\ldots,N\}$, where $N = |\mathcal{R}|$. The equation is reformulated as:

$$y(p_0) = \sum_{p_n \in \mathcal{R}} w(p_n) \cdot x(p_0 + p_n + \Delta p_n) \tag{1.2}$$

Here, the sampling is performed on the irregular and offset locations $p_n + \Delta p_n$. The offsets Δp_n are obtained by applying a separate convolutional layer over the same input feature map, allowing the deformation to be conditioned on the input features in a local, dense, and adaptive manner.

Since the learned offset Δp_n is typically fractional, the pixel value $x(p)$ at an arbitrary location $p = p_0 + p_n + \Delta p_n$ is computed via bilinear interpolation:

$$x(p) = \sum_q G(q,p) \cdot x(q) \tag{1.3}$$

where q enumerates all integral spatial locations in the feature map x, and $G(\cdot,\cdot)$ is the bilinear interpolation kernel. This differentiability allows the offsets to be

learned end-to-end via standard back-propagation.

1.4.2 Deformable Attention

While DCN improves geometric adaptability, detecting small objects in cluttered aerial images requires a mechanism to focus on sparse, informative key elements. Standard Transformer attention modules suffer from slow convergence and high computational complexity because they look over all possible spatial locations in the image feature maps.

To mitigate these issues, this research integrates the Deformable Attention module proposed by Zhu et al. [19]. This mechanism combines the sparse spatial sampling of deformable convolution with the relation modeling capability of Transformers.

Unlike standard Multi-Head Self-Attention (MHSA) which has quadratic complexity relative to pixel numbers, the Deformable Attention module only attends to a small set of key sampling points around a reference point.

Given an input feature map $x \in \mathbb{R}^{C \times H \times W}$, a query element q with content feature z_q, and a 2-D reference point p_q, the Deformable Attention feature is calculated as:

$$\text{DeformAttn}(z_q, p_q, x) = \sum_{m=1}^{M} W_m \left[\sum_{k=1}^{K} A_{mqk} \cdot W'_m x(p_q + \Delta p_{mqk}) \right] \tag{1.4}$$

where:

- m indexes the attention head (M heads total).
- k indexes the sampled keys (K keys total), where $K \ll HW$.
- Δp_{mqk} and A_{mqk} denote the sampling offset and attention weight of the k^{th} sampling point, respectively.
- A_{mqk} is normalized such that $\sum_{k=1}^{K} A_{mqk} = 1$.

Similar to DCN, the term $x(p_q + \Delta p_{mqk})$ is computed using bilinear interpolation. This design allows the model to focus on a small, fixed number of keys for each query, significantly reducing computational complexity to $O(2N_qC^2 + \min(HWC^2, N_qKC^2))$, which is linear with respect to the spatial size.

1.5 Hypothesis

Based on the structural limitations of standard CNNs and the proposed architectural enhancements, this research posits the following hypotheses regarding the

performance of the modified YOLO11 framework:

1. **Improved Accuracy for Oriented Objects:** Integrating Deformable Convolution modules into the YOLO11 backbone will significantly improve the detection accuracy of oriented objects compared to the baseline model. This prediction relies on the ability of Deformable Convolution to learn adaptive sampling offsets, allowing the receptive field to align dynamically with the rotation and geometric variations of aerial targets [14].

2. **Enhanced Small Object Detection:** Implementing Deformable Attention mechanisms will measurably increase the detection performance for small objects in cluttered environments. By focusing computational resources on a sparse set of key sampling points rather than the entire feature map, this mechanism is expected to suppress background noise and preserve the weak feature signals of small targets more effectively than the baseline architecture [19].

3. **Synergistic Performance of the Hybrid Architecture:** The hybrid combination of Deformable Convolution and Deformable Attention is expected to yield a synergistic effect, achieving a superior balance between accuracy (mAP) and computational efficiency. The proposed architecture is expected to outperform both the baseline YOLO11 and single-modification variants on the DOTA and SODA datasets by simultaneously addressing geometric misalignment and feature scarcity without incurring the quadratic complexity of standard Transformers.

1.6 Research Methodology

The research methodology employed in this thesis follows a systematic experimental approach, structured around distinct Work Packages (WP). This structure ensures a logical progression from theoretical understanding to practical implementation and final evaluation. The overall flow of the research is visualized in Figure 1.1.

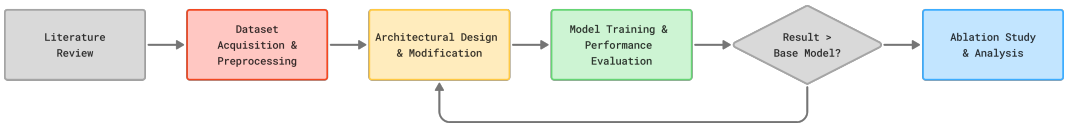


Fig. 1.1 Research Methodology Flowchart

The specific Work Packages (WP) for this research are outlined as follows:

- **WP 1: Literature Review**
  This work package involves an in-depth analysis of existing literature to establish a strong theoretical foundation. The review focuses on the evolution of the YOLO architecture up to YOLO11, the mathematical principles of Deformable Convolutional Networks (DCN), and the mechanics of Deformable Attention.

- **WP 2: Dataset Acquisition and Preprocessing**
  This work package focuses on preparing the data required for training and evaluation. Acquiring standard aerial imagery datasets such as DOTA (Dataset for Object deTection in Aerial images) and SODA (Small Object Detection dAtaset). Then preprocess data annotation labels from polygon or complex formats into the YOLO OBB format (center-point, width, height, angle).

- **WP 3: Architectural Design and Modification**
  This is the core development phase where the YOLO11 baseline is structurally enhanced. The modifications include:

  1. **DCN Integration:** Replacing standard convolutional layers in the Backbone and Neck with Deformable Convolution modules to enable adaptive receptive field learning.
  2. **Attention Integration:** Designing the replacement of the standard C2PSA block with the Deformable Attention mechanism to improve feature focusing on small targets.
  3. **Hybrid Design:** Validating the two methods compatibility and ensuring that the combined architecture maintains computational efficiency while enhancing detection capabilities.

- **WP 4: Model Training and Performance Evaluation**
  This work package include the implementation, training, and quantitative evaluation of the proposed architecture. The model is implemented in PyTorch within Ultralytics YOLO framework. Then the model is then trained and evaluated on a unseen test set using standard metrics, including Mean Average Precision (mAP) for Oriented Bounding Boxes (mAP50 and mAP50-95). Furthermore, computational efficiency is benchmarked via Number of Parameters (Params) and Floating Point Operations (FLOPs) to ensure practical viability against the baseline YOLO11.

- **WP 5: Ablation Study and Analysis**
  To validate the individual contributions of the proposed enhancements, this WP involves conducting ablation studies. Separate models will be trained with single modifications (e.g., YOLO11+DCN only, YOLO11+Deformable Attention only) to isolate the performance gains attributed to Deformable Convolution versus Deformable Attention. The analysis will also include qualitative visualization of detection results to identify improvements in handling background clutter and rotation.

1.7 Timeline

The research activities are scheduled to be completed within a period of six months. The timeline is structured to ensure that each phase of the methodology is given sufficient time for thorough execution and analysis. Table 1.1 details the schedule of activities.

Table 1.1 Research Timeline

No.	Activity	Month					
		1	**2**	**3**	**4**	**5**	**6**
1	Literature Review						
2	Dataset Acquisition and Preprocessing						
3	Architectural Design and Modification						
4	Model Training and Performance Evaluation						
5	Ablation Study and Analysis						
6	Conclusion and Thesis Writing						

- **Month 1:** Focus on understanding the theoretical background, reviewing related works, and finalizing the research proposal. Initial dataset acquisition begins.
- **Month 2:** Completion of dataset preprocessing. Start of the architectural design phase, identifying where to integrate DCN and Attention modules.
- **Month 3:** Finalizing the system design. Beginning the implementation of the modified YOLO11 model in code and starting initial training runs.
- **Month 4:** Intensive model training and tuning. Conducting the first round of evaluations on the validation set.

- **Month 5:** Performing comprehensive testing on the test set. Conducting ablation studies to isolate the effects of specific modules. Beginning the analysis of results.
- **Month 6:** Finalizing the analysis. Writing the complete thesis report, preparing for the defense, and drafting a paper for publication.

CHAPTER 2
BASIC CONCEPT

This chapter discusses the theoretical and architectural foundations necessary for comprehending the proposed enhancements to the YOLO11 framework. It provides an overview of the evolution of object detection paradigms, highlighting the unique challenges associated with detecting small, arbitrarily oriented objects in aerial imagery. Additionally, this chapter presents a detailed explanation of Convolutional Neural Networks (CNNs), the architectural specifics of YOLO11, and the adaptive mechanisms of Deformable Convolution Networks (DCN) and Deformable Attention, which form the core technical contributions of this thesis.

2.1 Object Detection

Object detection stands as one of the most fundamental and challenging problems in the field of computer vision. Unlike image classification, which assigns a single label to an entire image, or semantic segmentation, which classifies pixels without differentiating object instances, object detection requires the simultaneous localization and classification of multiple objects within a scene [20, 21]. The primary objective is to predict a set of bounding boxes, a rectangular delineations of object boundaries and associate a class probability score within each box. The complexity of this task is compounded in remote sensing and aerial surveillance domains, where varying altitudes, sensor angles, and environmental conditions introduce significant geometric variability [9, 20].

2.1.1 Small Object Detection

The detection of small objects represents a specialized sub-domain within computer vision. While general object detection on datasets like MS COCO achieves high performance, accuracy degrades precipitously as object size decreases. This is particularly critical in aerial imagery, where the combination of high altitude and wide field-of-view results in targets of interest, such as vehicles, pedestrians, or small maritime vessels occupying a very small fraction of the pixel space [11, 21]. Figure 2.1 illustrates typical scenarios where small object detection is required, highlighting the challenges posed by limited pixel representation and contextual ambiguity.

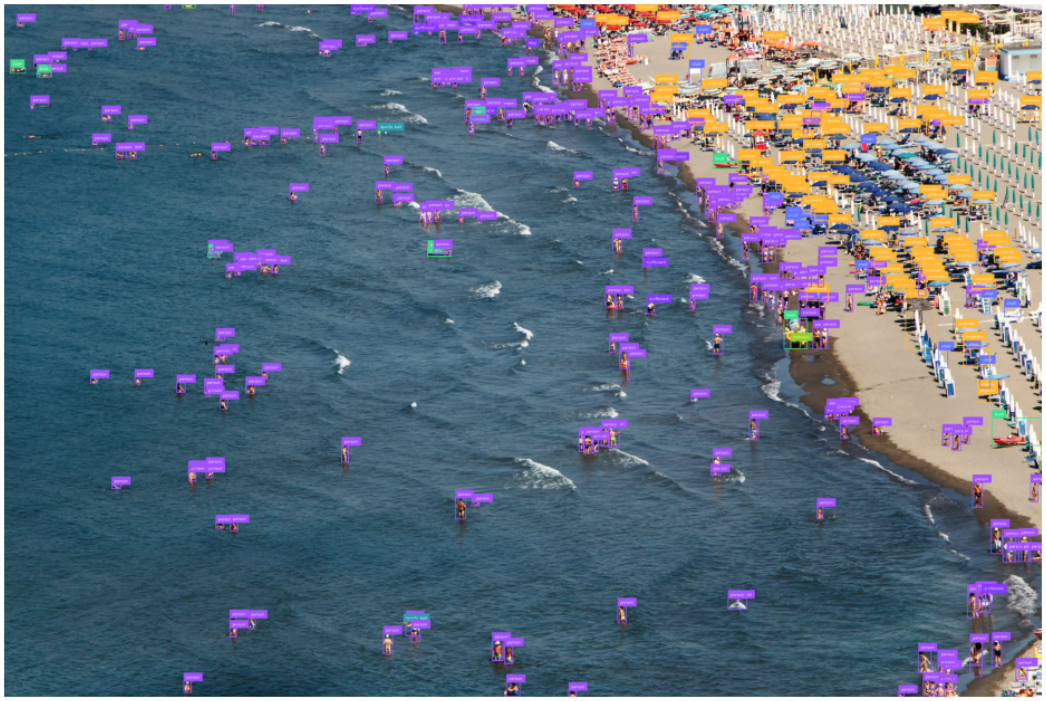


Fig. 2.1 Examples of small object detection challenges in aerial imagery

The definition of a "small object" is broadly dependent upon the application context and the dataset standards:

1. **Absolute Scale (MS COCO Definition):** The most widely accepted definition comes from the MS COCO evaluation protocol, which categorizes objects with a spatial area of less than 32×32 pixels as small ($area < 1024$ pixels) [11, 22].

2. **Relative Scale (SPIE Definition):** The Society of Photo-Optical Instrumentation Engineers (SPIE) defines small objects based on the image coverage ratio. An object is considered small if it occupies less than 0.12% of the total image area (e.g., roughly 9×9 pixels in a 256×256 image) [9].

3. **Aerial-Specific Scale (AI-TOD):** In specialized aerial datasets like AI-TOD, the definition is even more stringent, with the average object size often being around 12.8 pixels, significantly smaller than the COCO standard [23].

The difficulty in detecting small objects stems from fundamental limitations in the architecture of Convolutional Neural Networks (CNNs), specifically related to feature hierarchy and resolution.

- **Feature Vanishing and Dilution:** Deep CNNs rely on successive downsampling operations (strided convolutions or pooling) to increase the Receptive

Field (RF) and abstract high-level semantic features. A standard backbone (e.g., ResNet or CSPDarknet) typically has a total stride of 32 ($S = 32$). This means an input image of size 640×640 is reduced to a feature map of 20×20. Under this transformation, a small object of size 16×16 pixels in the input is theoretically mapped to an area of 0.5×0.5 pixels in the final feature map [9, 16]. In practice, this sub-pixel representation means the object's spatial information is effectively aggregated into a single feature vector mixed with surrounding background information, leading to severe feature dilution or complete vanishing of the signal [16, 17].

- **Low Signal-to-Noise Ratio (SNR):** Small objects possess very few pixels, limiting the visual information available for the network to learn discriminative features. Unlike large objects that exhibit rich internal textures and clear geometric structures (edges, corners), small objects often appear as amorphous blobs [9, 16]. This feature scarcity makes them highly susceptible to background clutter; a small rock or a patch of texture can easily be misclassified as a vehicle due to the lack of distinguishing details [10].

- **Occlusion and Dense Clustering:** In aerial imagery, small objects often appear in dense clusters (e.g., a parking lot full of cars or a dock full of boats). The overlap between valid objects and the interference from the background complicates the bounding box regression. Standard Intersection over Union (IoU) metrics are highly sensitive to small positional shifts for small objects, a misalignment of just a few pixels can result in a zero IoU score, destabilizing the training process [9, 13].

- **Context Dependence:** Because intrinsic features are weak, the detection of small objects relies heavily on contextual cues (e.g., a car is likely on a road, a ship is likely in water). However, standard CNN operations with fixed receptive fields may fail to capture this global context effectively if the object is isolated or the background is heterogeneous [16, 17].

Addressing these challenges requires architectural innovations that preserve high-resolution features, enhance feature extraction adaptively, and incorporate contextual reasoning. The subsequent sections will explore how Deformable Convolution Networks and Deformable Attention mechanisms can be integrated into the YOLO11 framework to specifically tackle the intricacies of small object detection in aerial imagery.

2.1.2 Oriented Object Detection

Traditional object detection systems rely on Horizontal Bounding Boxes (HBB), parameterized by $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ or (x_c, y_c, w, h). While sufficient for ground-level photography where gravity imposes a natural vertical orientation on most objects, HBBs are fundamentally inadequate for aerial and satellite imagery [21, 24].

In top-down aerial views, objects have an additional degree of freedom that is rotation. A ship or vehicle can point in any direction, so axis-aligned bounding boxes either include excessive background or misalign the target, which makes IoU scores unreliable even when the localization is correct [24, 25]. The predominance of arbitrary orientations in aerial data renders Horizontal Bounding Boxes inadequate. Instead, oriented rectangles parameterized by center coordinates, width, height, and rotation angle are preferred for tight localization. Figure 2.2 illustrates the difference between Oriented Bounding Boxes (OBB) and Horizontal Bounding Boxes (HBB) in aerial imagery.

Fig. 2.2 Comparison between (a) Oriented Bounding Box (OBB) and (b) Horizontal Bounding Box (HBB) in aerial imagery

2.2 YOLO11 Model

The YOLO11 model is a state-of-the-art single-stage object detection framework that builds upon the foundational principles of the YOLO (You Only Look Once) series. It is designed to achieve real-time detection speeds while maintaining high accuracy, making it suitable for applications requiring rapid inference, such as aerial surveillance and autonomous navigation [6]. YOLO11 introduces several architectural enhancements over its predecessors, including improved backbone networks for feature extraction, advanced neck designs for multi-scale feature fu-

sion, and refined head structures for precise bounding box regression and classification [8]. Figure 2.3 illustrates the overall architecture of the YOLO11 model.

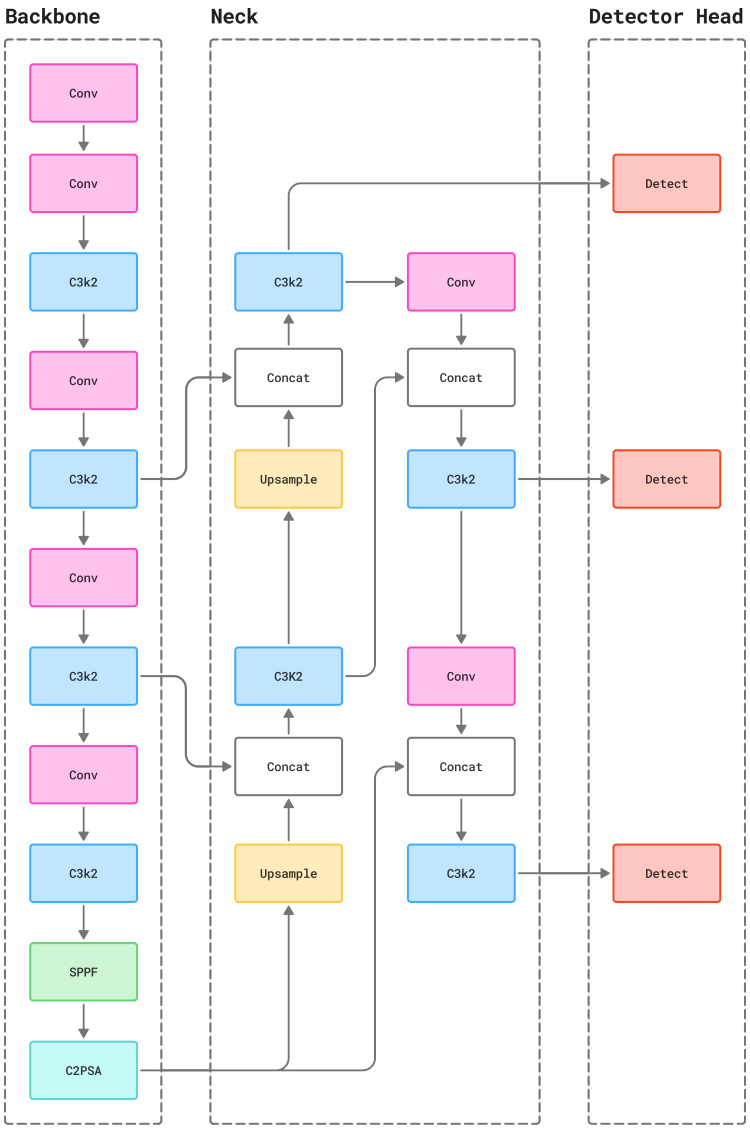


Fig. 2.3 YOLO11 Model Architecture.

At its core, the YOLO architecture has three main components. The backbone is the primary feature extractor, typically a deep convolutional neural network (CNN) that processes the input image to generate rich feature maps. The neck component aggregates and refines these features across multiple scales, often employing structures like Feature Pyramid Networks (FPN) or Path Aggregation Networks (PANet) to enhance the model's ability to detect objects of varying sizes. Finally, the head is responsible for predicting bounding boxes, objectness scores, and class probabilities based on the processed features.

2.3 Deformable Convolution Networks (DCN)

Deformable Convolution Networks (DCN) were introduced to enhance the spatial sampling flexibility of standard convolutional operations in Convolutional Neural Networks (CNNs). Traditional convolutions sample input features at fixed, regular grid locations, which can limit their ability to capture geometric transformations and object deformations present in real-world images [14]. DCN addresses this limitation by learning additional offsets for the sampling locations, allowing the convolutional kernel to adaptively focus on relevant regions of the input feature map. This adaptability is particularly beneficial for detecting small, oriented objects in aerial imagery, where objects may appear at various scales and orientations [26].

2.3.1 Deformable Convolution (DCNv1)

The output feature $y(p_0)$ at location p_0 of a regular convolution is the weighted sum of features sampled at a fixed grid:

$$y(p_0) = \sum_{p_n \in \mathcal{R}} w(p_n) \cdot x(p_0 + p_n) \tag{2.1}$$

where $w(p_n)$ are the kernel weights and x is the input map [14, 27].

Deformable convolution augments the regular grid $\mathcal{R}$ with 2D offsets $\{\Delta p_n | n = 1, \ldots, N\}$, where $N = |\mathcal{R}|$. The equation transforms to:

$$y(p_0) = \sum_{p_n \in \mathcal{R}} w(p_n) \cdot x(p_0 + p_n + \Delta p_n) \tag{2.2}$$

Here, the sampling is performed at the irregular locations $p_n + \Delta p_n$. These offsets allow the sampling points to shift to cover the semantic parts of an object—for instance, spreading along the wings of a rotated airplane rather than sampling the empty tarmac [14].

Since the learned offsets Δp_n are typically fractional (continuous values), the pixel coordinates $p = p_0 + p_n + \Delta p_n$ do not align with the integer grid of the feature map. To compute the pixel value $x(p)$, bilinear interpolation is employed:

$$x(p) = \sum_q G(q, p) \cdot x(q) \tag{2.3}$$

where q enumerates the integral spatial locations in the feature map (e.g., the 4

nearest neighbors), and $G(q,p)$ is the bilinear interpolation kernel:

$$G(q,p) = \max(0, 1 - |q_x - p_x|) \cdot \max(0, 1 - |q_y - p_y|)$$

This formulation ensures that the operation is fully differentiable, allowing the offsets Δp_n to be learned via standard backpropagation [14].

2.3.2 Modulated Deformable Convolution (DCNv2)

While DCNv1 allows the sampling points to move, DCNv2 introduces a modulation mechanism to further enhance feature extraction capability. It adds a learnable scalar weight Δm_k to each sampling point, bounded between 0 and 1 via a sigmoid function. The formulation becomes:

$$y(p_0) = \sum_{p_n \in \mathcal{R}} w(p_n) \cdot x(p_0 + p_n + \Delta p_n) \cdot \Delta m_n \tag{2.4}$$

This modulation scalar Δm_n allows the network to adjust the "amplitude" or importance of each sampling point. Crucially, it enables the network to "turn off" sampling points that fall on irrelevant background regions or noise, acting as a local attention mechanism within the convolution kernel.

2.3.3 Integration and Offset Learning

The offsets Δp_n and modulation scalars Δm_n are not fixed parameters, they are dynamic outputs of the network itself. They are generated by a separate, lightweight convolutional layer (typically a 3×3 conv) applied to the same input feature map x [14, 26].

- **Input:** A feature map x of size $H \times W \times C$ serves as the shared source for both the detection head and the offset generator.
- **Offset Generator:** A convolution layer produces a tensor of size $H \times W \times 3N$ (with N denoting the kernel size, e.g., 9 for 3×3). The $3N$ channels encode the x-offset, y-offset, and modulation scalar m for each kernel element, enabling the kernel to adapt to local geometry [14, 26].
- **Conditioning:** Because offsets are predicted from the same visual evidence they transform, the network can "look" at the object's morphology (e.g., a ship's elongated hull) and emit offsets that align the receptive field with that structure, achieving rotation and scale adaptability.

2.4 Attention Mechanisms

Attention mechanisms have revolutionized the field of deep learning, particularly in natural language processing and computer vision. They enable models to dynamically focus on the most relevant parts of the input data, enhancing feature representation and improving performance on various tasks [28, 29]. In the context of object detection, attention mechanisms help models to better capture contextual relationships and spatial dependencies, which is especially beneficial for detecting small and oriented objects in complex scenes [17].

2.4.1 Fundamentals of Visual Attention

The concept of "attention" in computer vision draws inspiration from human cognitive systems, where the visual cortex selectively focuses on specific parts of a scene to process relevant information while ignoring the rest [28, 30]. In the context of Deep Learning, an attention mechanism is mathematically defined as a dynamic weight adjustment function. It computes a set of "importance scores" or weights based on the input features and applies these weights to emphasize informative regions (e.g., objects) and suppress irrelevant ones (e.g., background clutter) [28].

Standard attention mechanisms, such as the Self-Attention found in Transformers, compute the relationship between every pair of pixels in the feature map to capture global context.

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.5}$$

While powerful, this global computation has a quadratic complexity of $O(H^2W^2)$ with respect to the feature map size. For small object detection, which requires high-resolution feature maps to prevent feature vanishing, this computational cost is often prohibitive [19, 28].

2.4.2 Deformable Attention

Deformable Attention, introduced in Deformable DETR, addresses the computational bottleneck of standard attention by combining the sparse sampling principles of DCN with the relation modeling of Transformers. Instead of attending to all pixels in the image, each query element attends only to a small, fixed set of key sampling points learned from the data.

Given an input feature map x, a query element q with content feature z_q, and a 2D reference point p_q, the Deformable Attention output is computed as:

$$\text{DeformAttn}(z_q, p_q, x) = \sum_{m=1}^{M} W_m \sum_{k=1}^{K} A_{mqk} \cdot W'_m x(p_q + \Delta p_{mqk}) \tag{2.6}$$

- **Sparse Sampling (K):** k indexes a small set of sampling points (e.g., $K = 4$). This reduces the complexity from quadratic $O(H^2W^2)$ to linear $O(HW \cdot K)$, making it feasible for high-resolution maps needed for small objects [19].

- **Reference Point (p_q):** Unlike standard attention which is location-agnostic, Deformable Attention is grounded at a reference point p_q. In the encoder, this is the grid location; in the decoder, it is predicted from object queries [19].

- **Sampling Offsets (Δp_{mqk}):** Similar to DCN, the network predicts offsets Δp_{mqk} from the query feature z_q. This allows the attention head to dynamically "look" for information in flexible locations relative to the reference point, adapting to the object's scale and shape [19].

- **Attention Weights (A_{mqk}):** These are scalar weights predicted from z_q, normalized via softmax such that $\sum A_{mqk} = 1$. They determine the contribution of each sampled point [19].

The Deformable Attention mechanism acts as a **spatial filter**. By restricting attention to K points, it inherently suppresses the vast majority of background noise pixels that would otherwise contribute to the weighted sum in global attention [18, 19]. For small, oriented objects, the learnable offsets Δp_{mqk} allow the model to concentrate its limited computational resources ("glances") precisely on the object's key features (e.g., the bow and stern of a ship), regardless of its orientation, while ignoring the surrounding water or land clutter. This significantly enhances the feature representation of small targets that are otherwise prone to being washed out in standard aggregation operations.

CHAPTER 3
SYSTEM DESIGN AND MODEL

This chapter details the research methodology employed in this study. It includes descriptions of the dataset used, the overall system architecture, modifications made to the YOLO11 model, and the performance evaluation metrics applied to assess the effectiveness of the proposed approach.

3.1 Dataset

This research utilizes two primary datasets for training and evaluating the object detection models: the Dataset for Object Detection in Aerial Images (DOTA) and the Small Object Detection dAtaset (SODA). Both datasets are specifically designed for aerial imagery and present unique challenges for object detection tasks.

3.1.1 DOTA Dataset

The DOTA (Dataset for Object deTection in Aerial images) is a large-scale geospatial dataset widely used as a baseline for oriented object detection [25]. It is built from 2,806 high-resolution aerial images collected from different sensors and platforms, including Google Earth, to ensure a wide diversity of acquisition conditions. The images span from approximately 800×800 to $4,000 \times 4,000$ pixels, and together provide 188,282 annotated instances belonging to 15 common object categories such as plane, ship, storage tank, stadiums, harbor, bridge, large and small vehicles, helicopter, roundabout, and swimming pool. DOTA is split into training, validation, and test subsets with proportions of $1/2$, $1/6$, and $1/3$ respectively, which facilitates standard benchmarking protocols. Each instance is annotated with an Oriented Bounding Box (OBB) defined by four vertices (x_i, y_i) arranged clockwise, which makes the dataset particularly suitable for aerial scenarios where objects appear at arbitrary rotations. The combination of extreme scale variation, dense object populations, and varying aspect ratios makes DOTA a challenging benchmark for evaluating object detection algorithms in aerial imagery. Figure 3.1 illustrates several instances from the DOTA dataset.

Fig. 3.1 Instances from the DOTA dataset.

3.1.2 SODA Dataset

The second dataset used in this research is the Small Object Detection dAtaset (SODA) benchmark [11], which contains two subsets tailored for driving (SODA-D) and aerial (SODA-A) scenarios. SODA-D spans 24,828 high-resolution street-level images with 278,433 instances labeled with Horizontal Bounding Boxes, while SODA-A focuses on aerial imagery and is the subset used in this research.

SODA-A comprises 2,513 aerial images harvested from Google Earth, each at an average resolution of 4761×2777 pixels to preserve the fine structure of tiny targets. The dataset contains 872,069 objects distributed over 9 classes, including Airplane, Helicopter, Small-vehicle, Large-vehicle, Ship, Container, Storage-tank, Swimming-pool, and Windmill. Data splits follow a 40% / 25% / 35% ratio for training, validation, and testing, respectively. Similar to DOTA, annotations use Oriented Bounding Boxes (OBB) to capture rotated objects, but SODA-A explicitly targets the small-object challenge: objects with area under 1024 pixels are considered small, and the average instance size is only 14.75 pixels, with many instances qualifying as ”Extremely Small” (area ¡ 144 pixels). The dataset also exhibits extreme density, with some images containing over 11,000 instances, which compounds occlusion and background interference issues. These characteristics make

SODA-A a demanding benchmark for evaluating detection performance on small, densely packed aerial targets. Figure 3.2 shows several instances from SODA-A.

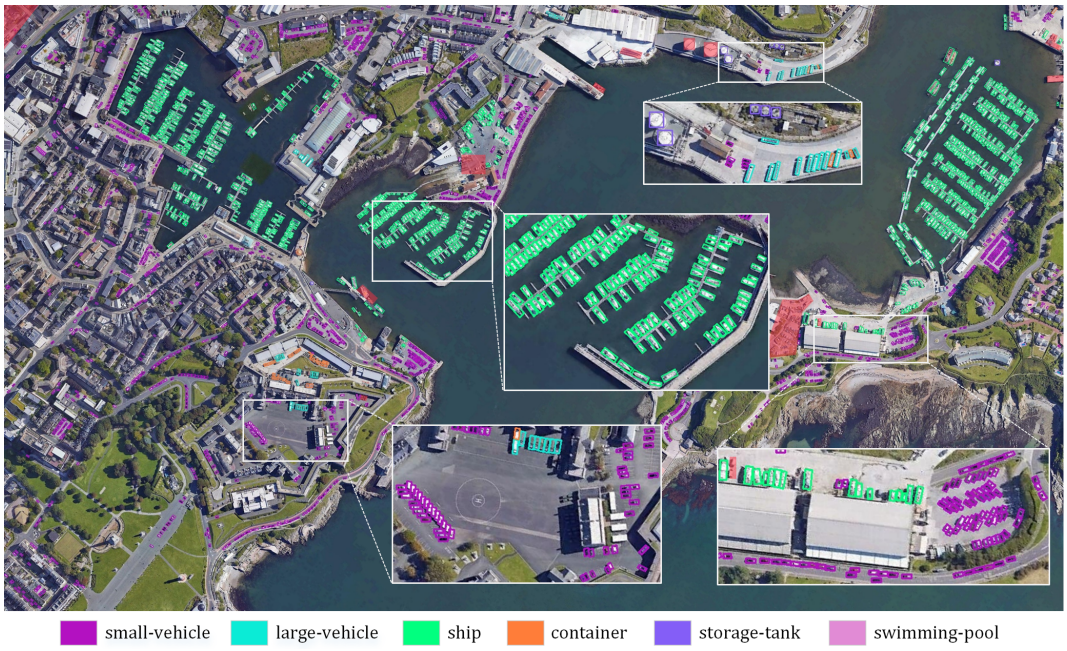


Fig. 3.2 Instances from the SODA-A dataset.

The following Table 3.1 summarizes the key features of both datasets used in this research.

Table 3.1 Summary of DOTA and SODA-A Datasets.

Feature	DOTA	SODA-A
Domain	General Aerial Detection	Small Object / Dense Aerial Detection
Images	2,806	2,513
Instances	188,282	872,069
Categories	15	9
Annotation Type	Oriented Bounding Box (OBB)	Oriented Bounding Box (OBB)
Key Challenge	Scale Variation, Orientation	Extremely Small Objects, High Density

3.2 System Process Architecture

Figure 3.3 illustrates the overall Experiment and Testing flow used in this research. The process begins with dataset preparation, where images and annotations from DOTA and SODA-A are standardized and converted into the annotation format required by the Ultralytics YOLO framework. The prepared dataset is then partitioned into training, validation, and testing subsets according to each dataset's predefined allocation ratios.

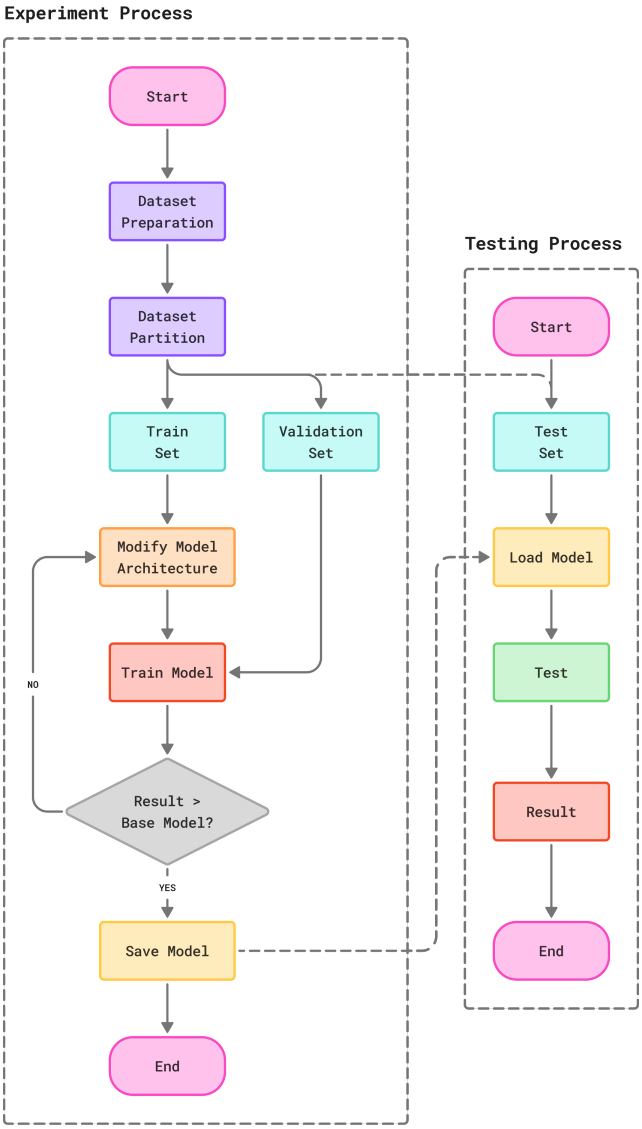


Fig. 3.3 System Process Architecture.

Next, the modified YOLO11 model is trained on the training set while monitoring validation performance to guide early stopping or further architecture tuning. When a trained checkpoint surpasses the established baseline, it is saved for later testing; otherwise, additional architectural modifications or hyperparameter adjustments are applied and the training loop repeats. The saved model is eventually loaded during the testing phase, where inference is performed on the unseen test set and the predicted boxes are compared to the ground-truth annotations to compute the evaluation metrics in this research. This system-wide process ensures that data collection, model development, and evaluation are tightly coordinated, which is essential for improving accuracy and efficiency in detecting aerial objects.

3.3 Modified YOLO11 Architecture

The baseline YOLO11 is highly optimized for general object detection but exhibits specific limitations when applied to aerial imagery, particularly regarding oriented and small objects. These limitations stem from the fixed geometric structures in standard convolutional layers, which struggle to capture the arbitrary orientations and scale variations typical in aerial views. Furthermore, the local receptive fields of standard CNNs restrict the modeling of long-range dependencies, making it difficult to identify small objects within complex backgrounds. To address these challenges, this research proposes YOLO11-DCN-DA, a hybrid architecture that integrates Deformable Convolutional Networks (DCN) and Deformable Attention mechanisms. Specifically, C3k2_DCN modules replace standard C3k2 units in the backbone and neck to enhance geometric adaptability, while Deformable Attention blocks are added for context-aware feature fusion. As shown in Figure 3.4, these modifications are strategically placed where feature extraction is most critical rather than being applied uniformly across the network.

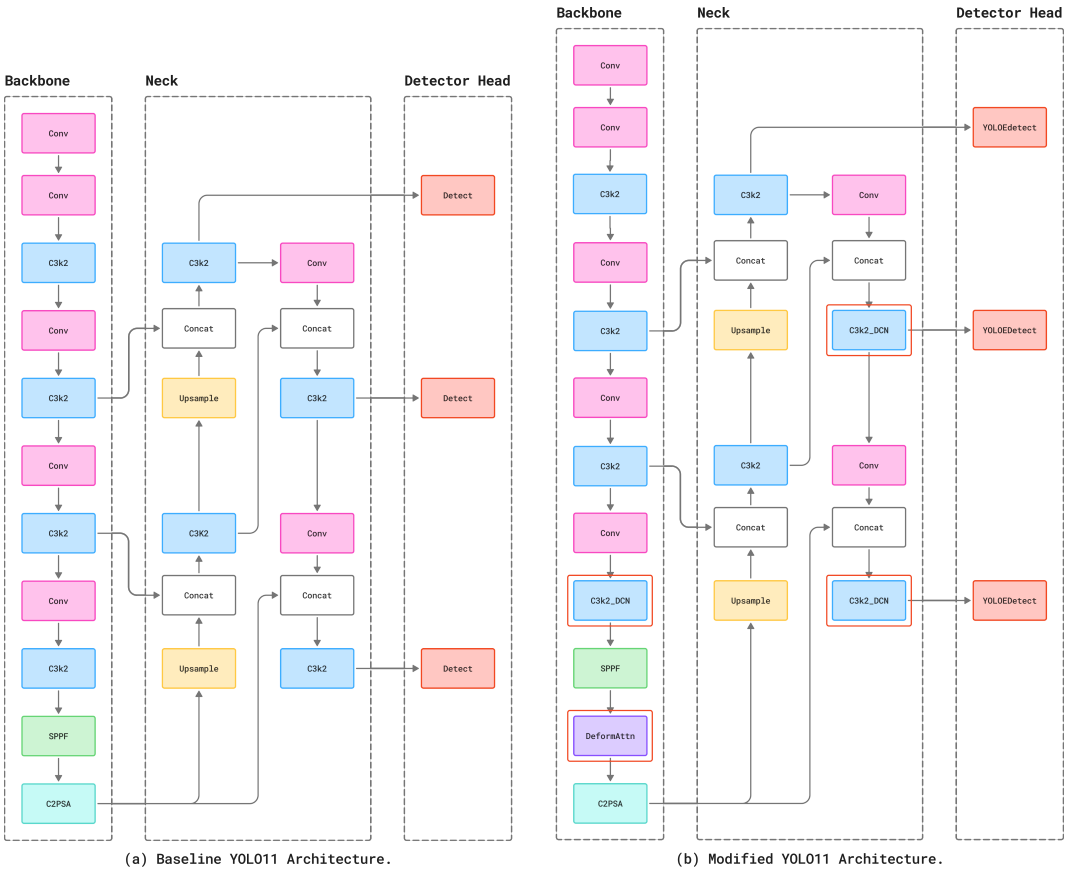


Fig. 3.4 Comparison of the (a) Baseline YOLO11 Architecture and (b) Proposed YOLO11-DCN-DA Architecture.

3.3.1 Integration of C3k2_DCN Module

The fundamental building block of the YOLO11 backbone is the C3k2 module, an evolution of the Cross Stage Partial (CSP) bottleneck designed to optimize gradient flow. In its standard form, the internal bottleneck relies on 3×3 convolutions with a fixed sampling grid. This rigidity prevents the network from effectively aligning its receptive field with rotated objects, such as ships or vehicles in aerial views.

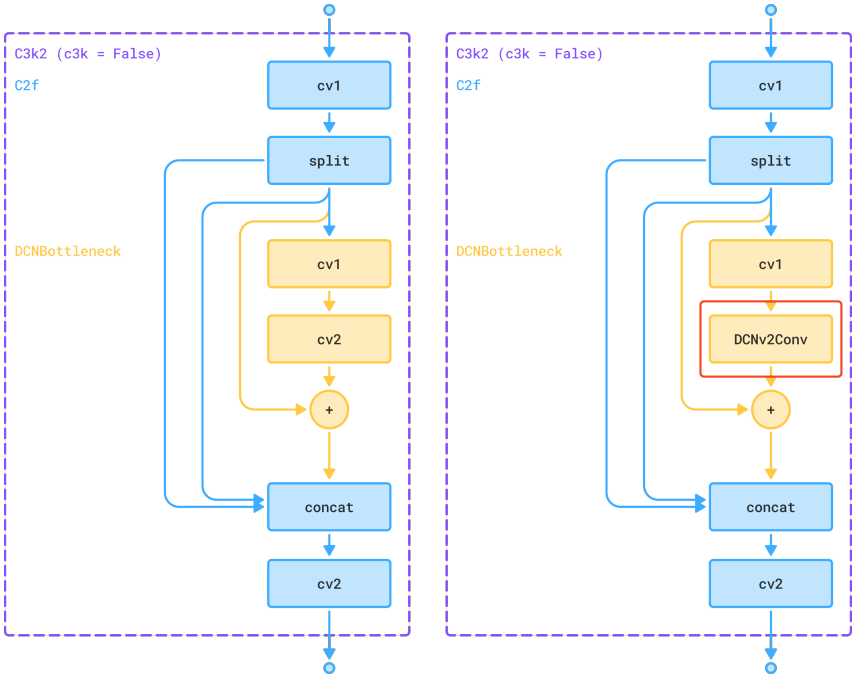
To address this, we introduce the C3k2_DCN module. This custom block modifies the internal bottleneck structure by replacing the standard spatial convolution with a Modulated Deformable Convolution (DCNv2) layer. The modification process involves three key steps:

1. Split and Branching
   Consistent with the CSP design principle, the input tensor is first processed by a 1×1 convolution and then split into two branches. This preserves the gradient flow benefits of the original architecture.

2. Offset Regression Branch
   In the processing branch, a parallel lightweight convolutional layer is introduced. This layer takes the input feature map and predicts a dense offset field. For a kernel size of 3×3, this layer outputs a tensor with 27 channels at each spatial location: 18 channels for the x and y coordinate offsets (Δp) and 9 channels for the modulation scalars (Δm).

3. Adaptive Sampling
   The DCNv2 layer utilizes these learned offsets to deform the sampling grid. This effectively "stretching" or "rotates" the kernel's receptive field to align with the geometric structure of the target object, capturing semantic features that would otherwise be missed by a fixed grid.

Figure 3.5 details the internal structure of the proposed C3k2_DCN module. The module were strategically deployed in the medium and large stages of the Backbone and Neck, where high-level semantic information is richest and geometric transformation modeling is most critical.



(a) Standard C3k2 block. (b) Modified C3k2_DCN block.

Fig. 3.5 Comparison between (a) Standard C3k2 module and (b) Proposed C3k2_DCN module with deformable convolutions on bottleneck.

3.3.2 Deformable Attention Block Implementation

While DCN enhances local geometric adaptability, detecting small objects in dense clusters requires a mechanism to capture global context and suppress background noise. Standard global attention mechanisms (like in Vision Transformers) compute relationships between all pixel pairs, resulting in quadratic computational complexity ($O(H^2W^2)$), which is prohibitive for high-resolution feature maps.

To solve this, we implement a custom DeformableAtten block that wraps the Multi-Scale Deformable Attention (MSDeformAttn) mechanism into a CNN-compatible module. This block is designed to operate on 4D tensors (B,C,H,W) typically found in YOLO architectures, acting as a bridge between the CNN backbone and the feature fusion neck.

The implementation of this block involves a specific sequence of operations to translate between the spatial domain of CNNs and the sequence domain of attention mechanisms:

1. CNN-to-Sequence Transformation

   The core MSDeformAttn operator expects input in a sequence format. Our wrapper first flattens the spatial dimensions of the input feature map ($H \times W$) into a sequence of query elements of length $N = H \times W$. This transformation

prepares the data for the attention mechanism while preserving the feature dimensionality C.

2. Reference Point Generation
   Unlike standard attention which is location-agnostic, Deformable Attention requires a set of reference points for each query element. Our implementation generates a normalized 2D grid $p_q \in [0,1] \times [0,1]$ corresponding to the spatial locations of each pixel in the feature map. These reference points serve as the "anchor" locations from which the attention mechanism learns to sample.

$$p_{q(i,j)} = \left(\frac{j+0.5}{W}, \frac{i+0.5}{H} \right) \tag{3.1}$$

   where (i, j) are the spatial coordinates. This grid is generated dynamically based on the input feature map size, ensuring the block can handle varying resolutions.

3. Sparse Deformable Attention
   The flattened features and generated reference points are passed to the MSDeformAttn module. For each query element, the module predicts a small set of sampling offsets relative to its reference point. It then samples features from these offset locations using bilinear interpolation and aggregates them. This sparse sampling reduces the complexity to linear time $O(HW)$, allowing the model to capture long-range dependencies without processing every pixel pair.

4. Feed-Forward Network (FFN) and Residuals
   Following the standard Transformer design, the output of the attention mechanism is processed through a Feed-Forward Network (FFN). Our implementation includes normalization layers (LayerNorm) and residual connections around both the attention module and the FFN to facilitate gradient flow and training stability.

$$x = x + \text{Dropout}(\text{MSDeformAttn}(x, p_q)) \tag{3.2}$$

$$x = \text{LayerNorm}(x) \tag{3.3}$$

5. Sequence-to-CNN Reconstruction
   Finally, the processed sequence is reshaped back into the original 4D tensor format (B, C, H, W). This allows the feature map to be seamlessly passed to subsequent convolutional layers in the Neck.

Figure 3.6 illustrates the data flow within the Deformable Attention block. This block is placed at the end of the Backbone after the SPPF module, to refine the highest-level features with global context before multi-scale fusion occurs.

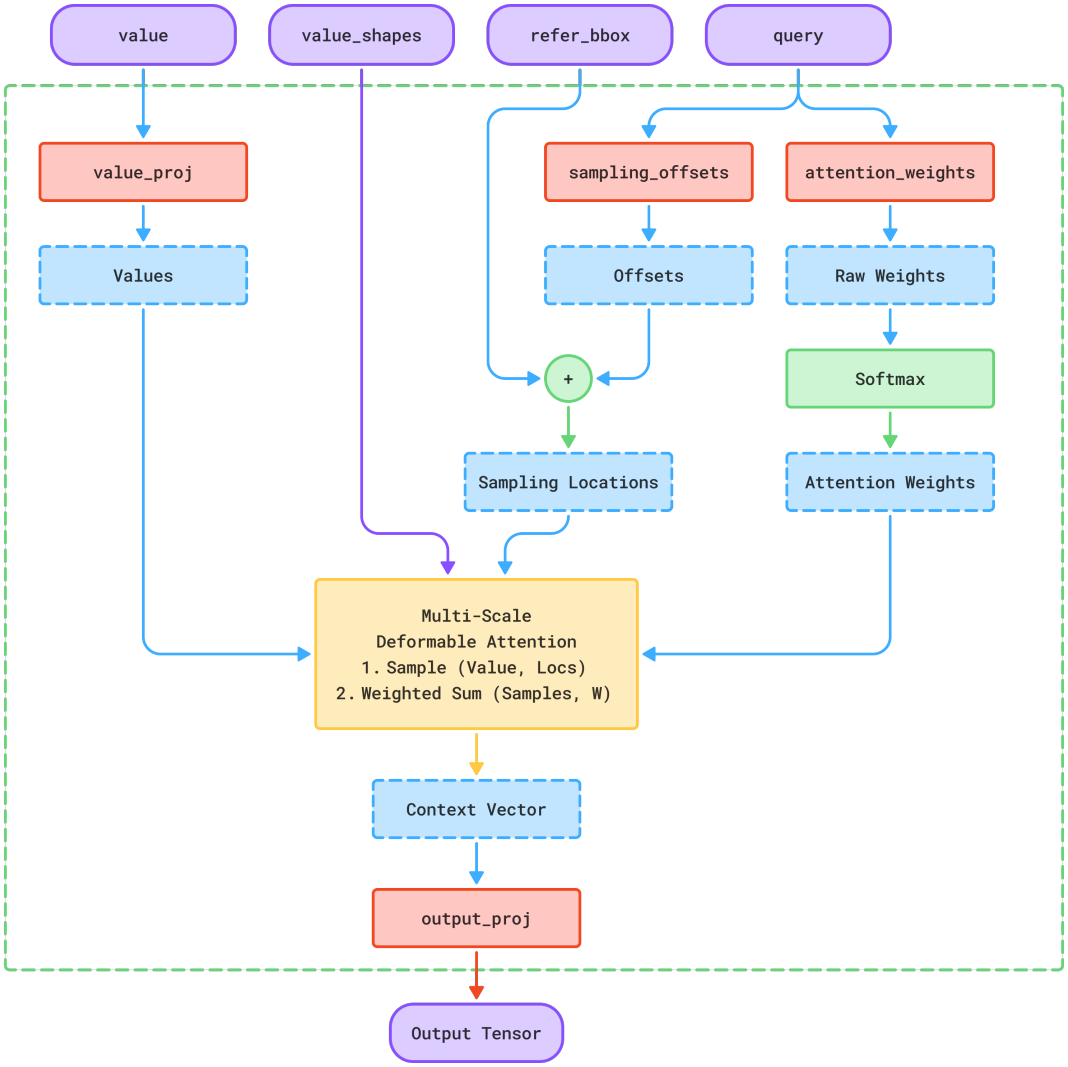


Fig. 3.6 Deformable Attention implementation block.

3.4 Performance Evaluation Metrics

In this research, the performance of the modified YOLO11 model is evaluated using standard object detection metrics, primarily focusing on Mean Average Precision (mAP) at different Intersection over Union (IoU) thresholds. Others metrics such as Precision, Recall, and F1-Score are also computed to provide a comprehensive assessment of the model's detection capabilities, especially in handling oriented and small objects in aerial imagery. To quantify computational cost, we also track the number of model parameters (Params) and Floating Point Operations(FLOPs),

ensuring that accuracy improvements remain practical for real-world deployment.

3.4.1 Precision

Precision measures the proportion of correctly predicted positive instances (true positives) out of all instances predicted as positive (true positives + false positives). It reflects the model's ability to avoid false alarms.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.4}$$

where TP is the number of true positives and FP is the number of false positives.

3.4.2 Recall

Recall measures the proportion of correctly predicted positive instances (true positives) out of all actual positive instances (true positives + false negatives). It indicates the model's ability to detect all relevant objects.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.5}$$

where TP is the number of true positives and FN is the number of false negatives.

3.4.3 F1-Score

The F1-Score is the harmonic mean of Precision and Recall, providing a single metric that balances both aspects of model performance. It is particularly useful when the class distribution is imbalanced.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.6}$$

3.4.4 Mean Average Precision (mAP)

Mean Average Precision (mAP) is the primary metric used to evaluate the object detection performance of the model. It is calculated by averaging the Average Precision (AP) across all object classes. AP is derived from the Precision-Recall curve for each class, which plots precision against recall at various confidence thresholds. The mAP is computed at different IoU thresholds (e.g., mAP@0.5, mAP@0.50-95) to assess how well the model detects objects with varying degrees of overlap with ground truth boxes.

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{3.7}$$

where N is the number of classes and AP_i is the Average Precision for class i.

3.4.5 Number of Parameters (Params)

The number of parameters (Params) in the model quantifies its complexity and size. It is calculated by summing all learnable weights and biases across all layers of the neural network. A lower number of parameters generally indicates a more efficient model, which is beneficial for deployment in resource-constrained environments.

$$\text{Params} = \sum_{i=1}^{L} (W_i + B_i) \tag{3.8}$$

where L is the number of layers, W_i is the number of weights in layer i, and B_i is the number of biases in layer i.

3.4.6 Floating Point Operations (FLOPs)

FLOPs measures the computational efficiency of the model by counting the number of floating-point operations required to process a single input image. It provides insight into the model's speed and resource requirements, which are critical for real-time applications.

$$\text{FLOPs} = \sum_{i=1}^{L} F_i \tag{3.9}$$

where L is the number of layers and F_i is the number of floating-point operations in layer i.

REFERENCES

[1] J. Sobek, J. R. M. Inojosa, B. J. M. Inojosa, S. M. Rassoulinejad-Mousavi, G. M. Conte, F. Lopez-Jimenez, and B. J. Erickson, "Medyolo: A medical image object detection framework," *Journal of Imaging Informatics in Medicine*, vol. 37, pp. 3208–3216, 12 2024.

[2] N. M. Alahdal, F. Abukhodair, L. H. Meftah, and A. Cherif, "Real-time object detection in autonomous vehicles with yolo," in *Procedia Computer Science*, vol. 246. Elsevier B.V., 2024, pp. 2792–2801.

[3] S. Abba, A. M. Bizi, J. A. Lee, S. Bakouri, and M. L. Crespo, "Real-time object detection, tracking, and monitoring framework for security surveillance systems," *Heliyon*, vol. 10, 8 2024.

[4] N. Saini, A. Dubey, D. Das, and C. Chattopadhyay, "Advancing open-set object detection in remote sensing using multimodal large language model," Tech. Rep., 2025.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016. [Online]. Available: https://arxiv.org/abs/1506.02640

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, *SSD: Single Shot MultiBox Detector*. Springer International Publishing, 2016, pp. 21–37. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46448-0_2

[8] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," 2024. [Online]. Available: https://arxiv.org/abs/2410.17725

[9] M. Nikouei, B. Baroutian, S. Nabavi, F. Taraghi, A. Aghaei, A. Sajedi, and M. E. Moghaddam, "Small object detection: A comprehensive survey on challenges, techniques and real-world applications," *Intelligent Systems with Applications*, vol. 27, p. 200561, Sep. 2025. [Online]. Available: http://dx.doi.org/10.1016/j.iswa.2025.200561

[10] J. Qiu, F. Cai, N. Fu, and Y. Yao, "Yolo-air: An efficient deep learning network for small object detection in drone-based imagery," *IEEE Access*, vol. 13, pp. 79 718–79 735, 2025.

[11] G. Cheng, X. Yuan, X. Yao, K. Yan, Q. Zeng, X. Xie, and J. Han, "Towards large-scale small object detection: Survey and benchmarks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2023. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2023.3290594

[12] Z. Qu, T. Han, and T. Yi, "Mffamm: A small object detection with multi-scale feature fusion and attention mechanism module," *Applied Sciences*, vol. 12, no. 18, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/18/8940

[13] X. Yang, J. Yan, Z. Feng, and T. He, "R3det: Refined single-stage detector with feature refinement for rotating object," 2020. [Online]. Available: https://arxiv.org/abs/1908.05612

[14] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," 2017. [Online]. Available: https://arxiv.org/abs/1703.06211

[15] X. Yuan, A. Chakravarty, L. Gu, Z. Wei, E. Lichtenberg, and T. Chen, "An empirical study of methods for small object detection from satellite imagery," 2025. [Online]. Available: https://arxiv.org/abs/2502.03674

[16] Y.-L. Chen, C.-L. Lin, Y.-C. Lin, and T.-C. Chen, "Transformer-cnn for small image object detection," *Signal Processing: Image Communication*, vol. 129, p. 117194, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092359652400095X

[17] A. M. Rekavandi, S. Rashidi, F. Boussaid, S. Hoefs, E. Akbas, and M. bennamoun, "Transformers in small object detection: A benchmark and survey of state-of-the-art," 2023. [Online]. Available: https://arxiv.org/abs/2309.04902

[18] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, X. Wang, and Y. Qiao, "Internimage: Exploring large-scale vision foundation models with deformable convolutions," 2023. [Online]. Available: https://arxiv.org/abs/2211.05778

[19] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," 2021. [Online]. Available: https://arxiv.org/abs/2010.04159

[20] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, 2023.

[21] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International journal of computer vision*, vol. 128, pp. 261–318, 2020.

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[23] J. Wang, W. Yang, H. Guo, R. Zhang, and G.-S. Xia, "Aitod: A challenge scheme for tiny object detection in aerial images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 504–513.

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

[25] 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," 2019. [Online]. Available: https://arxiv.org/abs/1711.10398

[26] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9308–9316.

[27] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, vol. 30, 2017.

[29] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[30] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, "Vision transformer with deformable attention," 2022. [Online]. Available: https://arxiv.org/abs/2201.00520