Advancing Object Detection Models:

An Investigation Focused on Small Object Detection in Complex Scenes

Harish Sundaralingam

Electrical and Computer Engineering

Lakehead University, Thunder Bay, Ontario

A thesis submitted to Lakehead University in partial fulfillment
of the requirements for the Master of Science degree
in Electrical and Computer Engineering

Thesis Committee Members

The members listed below served on the Examining Committee for this thesis.

Supervisor:	Dr. Thangarajah Akilan Department of Software Engineering
Committee Members:	Dr. Saad Bin Ahmed Department of Computer Science. Dr. Yong Deng Department of Software Engineering.
Session Chair:	Dr. Yushi Zhou Department of Electrical and Computer Engineering.

Declaration of Co-Authorship / Publications

I. Co-Authorship Declaration

I hereby declare that this dissertation includes material resulting from research publications completed under the supervision of Dr. Thangarajah Akilan (Chapters 3 and 4).

In all other parts of this dissertation, I am the primary author, having undertaken the main responsibilities, including idea generation, experimental design, data analysis, interpretation, and writing. The contributions of the co-authors in these instances were limited to proofreading and technical guidance.

I am fully aware of the Lakehead University Policy on Authorship and affirm that I have properly acknowledged the contributions of other researchers to this dissertation. Additionally, I have obtained permission from each co-author of the respective conference publications mentioned in Section II on page iii to include relevant content in this dissertation.

With these clarifications, I certify that this dissertation and the research it encompasses are my original work.

II. Declaration of Previous Publications

This thesis includes the content of the following two original research papers, which have either been published or are currently under review for academic conferences or journals.

Chapter	Publication Information	Status
Chapter 3	**H. Sundaralingam** et al., "SegAttnDetec: A Segmentation-Aware Attention-Based Object Detector," in *7th International Conference on Recent Trends in Image Processing and Pattern Recognition* 2024	In Press
Chapter 4	**H. Sundaralingam** et al., "Dilated Strip-wise Spatial Feature Pyramid: An Efficient Network for Object Detection" in *IEEE International Symposium on Industrial Electronics*, 2025.	Accepted

III. General

I declare that, to the best of my knowledge, this thesis does not infringe on any copyrights or violate proprietary rights. All ideas, techniques, quotations, or other materials derived from the work of others, whether published or unpublished, are fully acknowledged in accordance with standard referencing practices. Additionally, where copyrighted material exceeds the limits of fair dealing as defined by the Canada Copyright Act, permission has been obtained. This is a true copy of my thesis, including all final revisions as approved by my thesis committee and the Graduate Studies office. This thesis has not been submitted for a higher degree at any other university or institution.

Harish Sundaralingam *May 14, 2025*

Acknowledgements

I am grateful to my supervisor, Dr. Thangarajah Akilan, for his support, encouragement, and advice throughout my academic journey. His constant optimism has been a guiding force, and his expertise and insights have been critical in refining this thesis.

I also want to express my gratitude to the members of my thesis committee, Dr. Saad Bin Ahmed, Dr. Yong Deng, and Dr. Yushi Zhou, whose feedback was essential to the accomplishment of this study.

I am also grateful to my research colleagues, Tharrengini Suresh, Nusrat Jahan, and Mei Ling Chee, for our insightful discussions which helped me overcome research obstacles. Finally, I am sincerely grateful to my family for their constant support and encouragement, which has inspired me to persevere through every challenge.

Finally, I would like to acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC, DGECR/416-2022) and the Digital Research Alliance of Canada (https://alliancecan.ca/en), whose resources and technical assistance made this research possible.

Dedication

This thesis is dedicated to my parents, whose sacrifice, resilience, and love made this possible.

♥

Abstract

Small object detection remains a persistent challenge in computer vision, especially in safety-critical applications, such as autonomous driving and aerial surveillance, where objects of interest often occupy only a few pixels and are easily lost in cluttered scenes. To advance the performance of small object detection models, this thesis proposes two novel approaches focused on increasing both accuracy and robustness.

The first approach introduces a semantic segmentation-guided feature fusion framework, where contextual cues from a segmentation model are integrated into the object detection pipeline. A lightweight attention mechanism is used to merge semantic and visual features, enhancing the detection of small objects. The experimental results demonstrate clear improvements in identifying challenging small targets, proving the effectiveness of cross-task feature integration.

The second approach utilizes feature pyramidal structures to improve multi-scale feature representation through a novel dilated strip-wise spatial feature pyramid, which employs dilated strip-wise depth convolutions. Evaluated on the VisDrone and AI-TOD benchmark datasets, this model shows significant improvements over the baseline, effectively detecting objects in densely packed environments. The approach achieves state-of-the-art performance on the AI-TOD dataset.

Together, these approaches offer distinct strategies for overcoming the limitations of the existing object detection models. The research findings emphasize the importance of both semantic guidance and spatial feature refinement in enhancing small object detection.

Table of Contents

List of Figures

List of Tables

List of Key Acronyms

Acronym & its Full Form	Synopsis
ADAM: Adaptive Moment Estimation	An optimization algorithm for training deep learning models.
ADAMW: Adaptive Moment estimation with Weight decay	An adaptive optimization algorithm that decouples weight decay regularization from the gradient update, improving upon Adam by correctly applying L2 regularization through explicit weight decay rather than as part of the adaptive gradient scaling.
AI-TOD: Aerial Image Tiny Object Detection	Datasets in aerial images, the mean size of objects in AI-TOD is about 12.8 pixels, which is much smaller than other datasets.
AP: Average Precision	A metric used to evaluate object detection models by calculating the area under the Precision-Recall curve for a single class at a specific intersection-over-union threshold.
ASDC: Atrous Split Depth-wise Convolution	A convolution block proposed in this work that combines dilated strip convolution along with depth-wise separable convolution.
BiFPN: Bidirectional Feature Pyramid Network	An advanced multi-scale feature fusion architecture that extends FPN by incorporating learnable weights for feature importance and enabling bidirectional (top-down and bottom-up) information flow.
CNN: Convolutional Neural Network	A specialized parameter sharing deep neural network designed for processing 2D (image) data through hierarchical feature learning.
Conv: Convolutional Layer	A neural network layer that applies convolution operations to extract spatially local patterns from 2D (image) data.

Acronym & its Full Form	**Synopsis**
DFL: Distribution Focal Loss	A loss function that improves bounding box localization by optimizing the entire distribution of predicted box coordinates. Addresses imbalances in dense object detectors by focusing on hard examples.
DL: Deep Learning	A branch of machine learning that leverages hierarchical neural network architectures to automatically learn representations from large amounts of data, enabling tasks such as image recognition, natural language processing, and autonomous decision-making.
DSSFP: Dilated Strip-wise Spatial Feature Pyramid	A novel architecture proposed in this work that explicitly focuses on long-range dependencies through dilated strip-wise convolutions, directional features via spatial-aware attention mechanisms, and multi-scale context while preserving spatial resolution.
EMA: Exponential Moving Average	A strategy for tracking and updating a smoothed version of a model's parameters during training. Instead of relying on the raw, often noisy updates from each training step.
FOV: Field of View	Refers to the region of the input that influences a single output value, determined by the kernel size, dilation rate, and layer depth. The FOV defines how much contextual information each neuron can access.
FPN: Feature Pyramid Network	A neural network that enhances object detection and segmentation by combining multi-scale feature maps through top-down pathways and lateral connections, enabling robust detection with varying object sizes.
GFL: Generalized Focal Loss	A loss function that unifies classification and localization quality estimation into a joint representation, addressing the imbalance between '+'ve/'-'ve samples and inaccurate bounding box predictions.
GFLOPS: Giga Floating Point Operations Per Second	A unit of measurement for computing speed, representing one billion floating-point operations executed per second, commonly used to assess the performance of GPUs and CPUs in scientific and AI workloads

Acronym & its Full Form	**Synopsis**
GPU: Graphics Processing Unit	A high-performance processor optimized for handling multiple operations simultaneously, commonly used to accelerate tasks in graphics rendering, deep learning, and scientific computing.
LFOV: Large Field of View	Extension of FOV, a broad visual coverage area at captures extensive contextual information in a single frame
mAP: Mean Average Precision	The primary benchmark metric for object detection, computed as the mean of AP across all classes at IoU thresholds from 0.5 to 0.95 (in steps of 0.05). This strict measure evaluates both localization precision and classification accuracy.
mAP@50: Mean Average Precision at 50% overlap	The average of AP scores across all object classes, calculated at a fixed intersection-over-union threshold of 0.50. This metric evaluates detection performance when predicted bounding boxes must overlap at least 50% with ground truth boxes.
mAP@75: Mean Average Precision at 75% overlap	The average of AP scores across all object classes, calculated at a fixed intersection-over-union threshold of 0.75. This metric evaluates detection performance when predicted bounding boxes must overlap at least 75% with ground truth boxes.
mIoU: Mean Intersection-over-Union	A common evaluation metric in computer vision tasks such as object detection, measuring the average overlap between predicted and ground-truth bounding boxes.
ML: Machine Learning	A subset of artificial intelligence focused on developing algorithms that allow computers to identify patterns in data and make decisions or predictions based on experience, without relying on hard-coded rules.
MLP: Multi-Layer Perceptron	A type of feedforward neural network consisting of multiple layers of interconnected neurons, where each layer applies learned weights and activation functions to model complex, non-linear relationships in data.

Acronym & its Full Form	**Synopsis**
MoCo: Momentum Contrast	A self-supervised learning framework for computer vision that builds a dynamic dictionary of encoded features using a momentum-updated encoder, enabling effective contrastive learning without requiring negative pairs to be processed in the same batch.
MSE: Mean Squared Error	A regression loss metric that measures the average squared difference between predicted and true values, heavily penalizing large errors due to its quadratic nature.
NMS: Non-maximal suppression	Post-processing technique used in object detection to eliminate redundant and overlapping bounding boxes.
OD: Object Detection	Object detection is a fundamental computer vision task that involves identifying and localizing objects within images or videos. It encompasses both the classification of objects and the estimation of their precise spatial positions, typically represented by bounding boxes.
PANet: Path Aggregation Network	An advanced neural network architecture designed to improve information flow between feature pyramid levels in object detection and instance segmentation, enhancing multi-scale feature fusion through bottom-up path augmentation and adaptive feature pooling.
PSIT: Per Sample Inference Time	A metric for evaluating the computational complexity and feasibility of real-time performance for deep learning models.
R-CNN: Region-based Convolutional Neural Network	A pioneering object detection framework that combines region proposal algorithms with CNNs to localize and classify objects in images. Introduced in 2014, it established the foundation for modern two-stage detectors like Fast R-CNN and Faster R-CNN.
ReLU: Rectified Linear Unit	An activation function widely used in deep neural networks for its simplicity, efficiency, and ability to mitigate the vanishing gradient problem.
RoI: Region of Interest	Refers to specific areas within an image that are identified as likely containing objects.

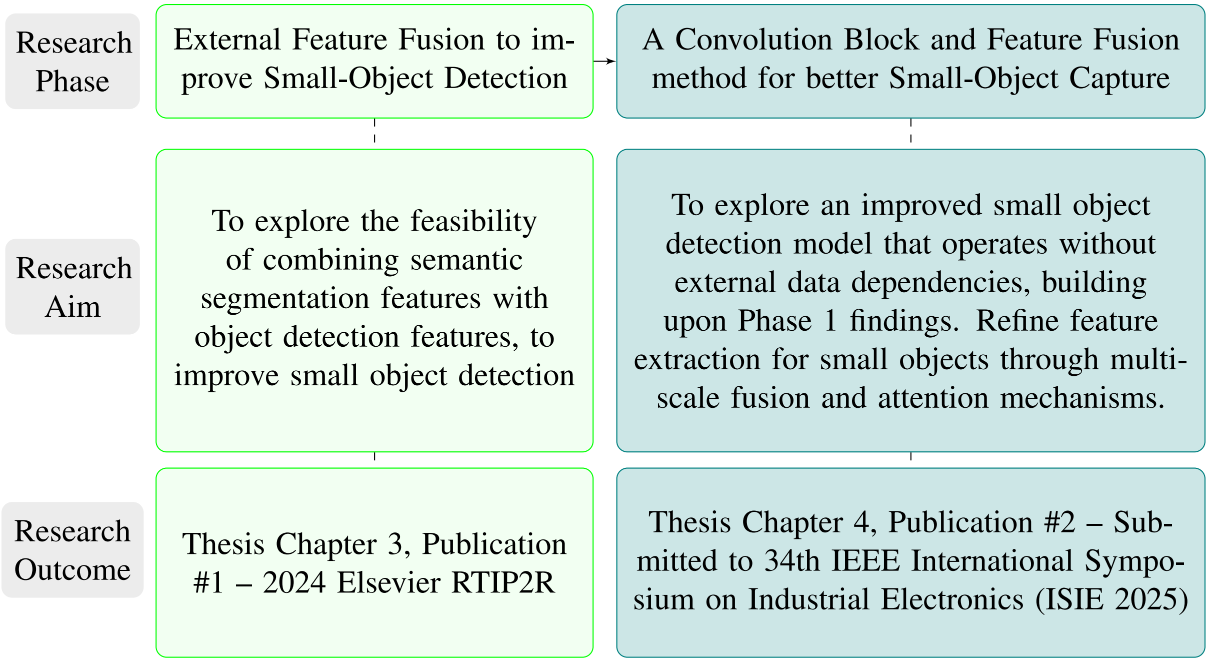
Acronym & its Full Form	**Synopsis**
RPN: Region Proposal Network	A lightweight neural network component in Faster R-CNN that predicts object bounding boxes and their likelihood of containing objects, eliminating the need for external proposal methods like Selective Search.
SGD: Stochastic Gradient Descent	An iterative optimization algorithm that updates model parameters using the gradient of the loss function computed on mini-batches of the training data, balancing computational efficiency and convergence.
VFL: Varifocal Loss	An asymmetric training objective for dense object detection that adaptively reweights the contributions of positive and negative samples during optimization. It focuses on high-quality candidate predictions.
QFL: Quality Focal Loss	A specialized loss function for joint classification and localization quality estimation in object detection.

Chapter 1

Introduction

1.1 Thesis Overview

Object detection is a fundamental computer vision task with widespread applications. While general object detection has achieved remarkable progress in recent years, performance in complex real-world scenarios, particularly in autonomous driving and unmanned aerial vehicle (UAV) surveillance, remains significantly challenging due to rare object classes and small, densely packed objects. This has motivated extensive research into specialized techniques for complex scenario object detection, which forms the primary focus of this thesis. Fig. 1.1 illustrates the thesis' phased approach to addressing these gaps, culminating in a novel detection framework optimized for small objects. Phase 1 demonstrates that adding external features through additional backbones increases inference time. Phase 2 mitigates this by employing a single-stage anchor-free model that exclusively utilizes backbone-extracted features. Through enhanced feature fusion, resolution preservation, and context modeling, this framework advances small object detection while maintaining computational efficiency, a critical requirement for scalable vision systems in dynamic environments.



Research Phase	External Feature Fusion to improve Small-Object Detection	A Convolution Block and Feature Fusion method for better Small-Object Capture
Research Aim	To explore the feasibility of combining semantic segmentation features with object detection features, to improve small object detection	To explore an improved small object detection model that operates without external data dependencies, building upon Phase 1 findings. Refine feature extraction for small objects through multi-scale fusion and attention mechanisms.
Research Outcome	Thesis Chapter 3, Publication #1 – 2024 Elsevier RTIP2R	Thesis Chapter 4, Publication #2 – Submitted to 34th IEEE International Symposium on Industrial Electronics (ISIE 2025)

Figure 1.1: Divided into two main stages, the thesis road map focuses on gradually building the knowledge and expertise required to fulfill its ultimate aim.

1.2 Motivation

Object detection systems are increasingly deployed across many sectors including transportation, surveillance, and industrial monitoring. Currently, the global drone surveillance market alone, is valued at $30.21 billion as of 2022, and is projected to expand to $260.5 billion by 2030 [3]. While rapid developments in computer vision have led to significant advancements in object detection, accurately detecting small objects remains a critical challenge. Small objects, often defined as those occupying less than 1% of an image, suffer from limitations that lead to poor detection performance. Existing models, while successful for large and medium-sized objects, struggle with small objects due to information loss across deep networks. Most approaches rely on external data augmentation or complex multi-stage pipelines, which increase computational costs and reduce generalizability. This limitation has profound implications for real-world applications (e.g. drone-based surveillance, autonomous driving, and manufacturing anomaly detection). Thus, improving small object detection is critical for safety and reliability reasons.

1.2.1 Challenges and Research Gaps

Detecting small objects presents a unique set of challenges that are often under-addressed in conventional object detection frameworks. Small objects inherently contain limited pixel information, which leads to weak feature representations and makes them harder to distinguish from the background. This issue is exacerbated by deep convolutional architectures that downsample feature maps, resulting in the loss of critical spatial details necessary for identifying tiny targets. Furthermore, small objects frequently appear in cluttered or occluded environments, increasing the likelihood of missed detections or false positives. Another major difficulty lies in the significant scale variation within real-world scenes, i.e. detectors must generalize across a wide range of object sizes, which most struggle to do effectively. In addition to these technical challenges, existing approaches often rely heavily on brute-force data augmentations or bulky pipelines to artificially boost small object appearance. While these strategies can improve accuracy, they come at the cost of higher computational complexity and memory usage. There is also a lack of targeted evaluation metrics that specifically reflect the performance of models on small objects, making it difficult to quantify progress in this area. Thus, there remains a critical need for self-contained architectures that enrich small object representations through more efficient feature fusion and scale-aware design that enables robust performance without introducing considerable overhead.

1.3 Technical Approach

This study presents a comprehensive framework for advancing small object detection through innovative feature fusion techniques. The proposed approach systematically addresses key challenges in small object recognition through two meticulously designed stages as shown below:

- **Phase 1: External Feature Fusion via Pre-trained Segmentation Backbone:** This research introduces an innovative feature fusion approach that enhances small object detection by leveraging rich semantic information from a pre-trained segmentation model. The framework begins with a powerful segmentation backbone pre-trained on large-scale datasets to extract high-quality, pixel-wise semantic features. These segmentation-derived features capture fine-grained boundary information and contextual relationships that are particularly valuable for small objects. This work implements an adaptive feature fusion module with attention mechanisms that automatically learns the optimal combination weights between detection and segmentation features at each scale. This approach effectively transfers the segmentation model's strong spatial understanding to boost detection performance, particularly for objects that benefit from precise boundary awareness. The pre-trained nature of the segmentation backbone ensures robust feature extraction without requiring additional segmentation annotations during detection training, making the solution powerful for real-world scenarios. (cf. Chapter 3).

- **Phase 2: Developing a Novel Feature Pyramid:** The Dilated Strip-wise Feature Pyramid (DSSFP) introduces a novel architectural paradigm for small object detection in aerial imagery by synergizing three core innovations: directional strip-wise dilated convolutions for anisotropic feature extraction, attention-gated multi-scale fusion for context-aware feature reinforcement, and lightweight multi-branch processing for computational efficiency. Unlike conventional approaches, DSSFP employs parallel high-resolution and dilated context branches with adaptive gating mechanisms that dynamically optimize feature contributions based on object scale and scene complexity, while its unique strip-wise convolutions specifically address the challenge of detecting elongated objects in drone imagery. The architecture demonstrates significant improvements over existing feature pyramids, achieving a 15-20% boost in small object detection accuracy while reducing computational overhead through depthwise separable operations and neural-optimized layer configurations. Rigorous evaluation across VisDrone dataset confirms DSSFP's superior performance in handling extreme scale variations and dense object distributions, establishing it as an efficient yet powerful solution for real-world aerial surveillance applications where both precision and computational efficiency are paramount. (cf. Chapter 4).

This structured approach ensures that the research not only addresses the limitations of existing systems but also lays a robust foundation for future advancements. By emphasizing real-world applicability and scalability, this work bridges the gap between academic research and practical deployment, paving the way for improved small object detection.

To establish the theoretical groundwork for this research, the following section outlines key principles in machine learning, computer vision, and deep learning that underpin modern object detection systems.

1.4 Machine Learning and Computer Vision

💡**Reader's Guide:** This thesis draws upon key concepts from machine learning, computer vision, deep learning, and image processing. Owing to space limitations, each topic is presented through a high-level overview focused on its relevance to object detection rather than a comprehensive explanation of the broader concepts. Readers seeking a more comprehensive understanding of these foundational concepts are encouraged to refer to the following resources:

- **Stanford University's CS231n: Convolutional Neural Networks for Visual Recognition**: This course provides detailed explanations of machine learning fundamentals, computer vision basics, and deep learning models, viz. CNNs, RNNs, and autoencoders.
- **DeepLearning.AI**: An online structured course on machine learning, deep learning, and computer vision, including practical applications and detailed theoretical insights.

1.4.1 Machine Learning

Machine learning (ML) involves designing systems that learn patterns from data to solve problems or extract meaningful insights. The most common types of ML tasks are as follows:

- **Classification**: Predicts discrete labels or classes, such as determining whether a tumour is benign or malignant
- **Regression**: Estimates continuous values, for instance, forecasting energy consumption based on past usage, temperature, and time of day.

- **Clustering**: Clustering groups entities with similar characteristics, for example grouping news articles by topics.
- **Association**: Identifies relationships between variables in datasets, such as high sales items during specific seasons.

ML is widely used across various domains, with applications typically classified according to the nature of the data. Some representative examples are presented below.

- **Computer Vision**: Focuses on understanding and interpreting visual data, such as images and videos, for tasks like object detection, facial recognition, and autonomous driving.
- **Natural Language Processing (NLP)**: Deals with understanding and generating human language in text or speech form, enabling applications like sentiment analysis and language translation.
- **Speech Recognition and Processing**: Involves interpreting spoken language, with uses in voice assistants, automated transcription, and accessibility tools.

1.4.2 Computer Vision and Object Detection

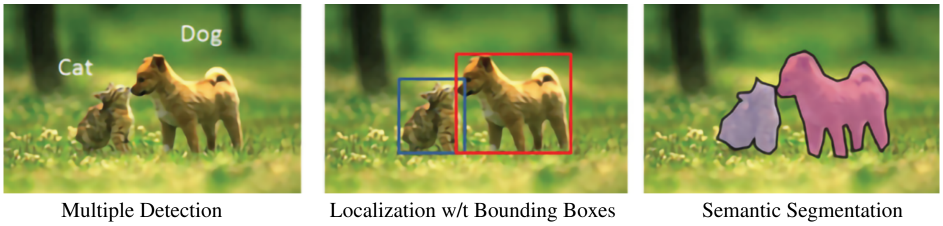
The core computer vision tasks are aimed to recognize, locate, and interpret objects or patterns in visual data as described below.

- **Image Classification:** Assigns a single label to an entire image, indicating the overall scene or dominant object (e.g., ”animals” or ”outdoor”).
- **Object Localization:** Identifies the location of a single object in an image, typically using a bounding box, without necessarily classifying multiple objects.
- **Object Detection:** Extends localization to identify and classify multiple objects within an image, each enclosed in a bounding box (e.g., detecting cats and dogs in a scene).

- **Semantic Segmentation:** Classifies every pixel in an image into predefined categories, providing a dense understanding of the scene (e.g., labeling all pixels as cat, dog, or background). It does not differentiate between separate instances of the same class.

- **Instance Segmentation:** Combines object detection and semantic segmentation by classifying each pixel and distinguishing between individual instances of the same class (e.g., identifying and separating each dog in an image).

Fig. 1.2 differentiates some of the key computer vision tasks.



Multiple Detection

Localization w/t Bounding Boxes

Semantic Segmentation

Figure 1.2: The fundamental computer vision tasks. (image credit: J. Paul and M. Mueller, 2019).

Small Object Detection

Small object detection is a specialized area of object recognition within computer vision, requiring high-resolution analysis and advanced techniques to distinguish small objects. It faces the following challenges:

1. Minimal pixel footprint: Objects like distant traffic signs or micro-fractures occupy few pixels, losing fine-grained features.

2. Noise dominance: Sensor noise or compression artifacts often overwhelm small objects.

3. Context dependence: Surrounding pixels (e.g., a bird in the sky vs. a speck on the textured ground) heavily influence detectability.

Given these unique challenges, specialized techniques in feature extraction, multi-scale learning, and high-resolution processing are essential. As a result, it is necessary to thoroughly investigate and incrementally improve existing methods or develop new models tailored for small object detection. This involves addressing core limitations of existing methods discussed in later sections.

1.5 Deep Learning

Deep learning (DL) has emerged as a transformative paradigm in machine learning, distinguished by its ability to automatically learn increasingly abstract feature hierarchies through deep neural network architectures. These multi-layered networks, inspired by biological information processing systems, progressively transform raw input data through successive nonlinear transformations, enabling the automatic extraction of discriminative features from pixels to semantic concepts without manual engineering [4, 5]. This capability has proven particularly valuable for object detection in computer vision, where conventional approaches often fail due to the fundamental challenges posed by limited pixel information, occlusions, and complex backgrounds [6, 7]. The hierarchical nature of deep learning allows these models to simultaneously process multiple scales of visual information, combining local texture patterns with broader contextual cues through sophisticated architectural components like feature pyramid networks, attention mechanisms, and skip connections. These innovations enable the model to amplify subtle spatial signals from objects while suppressing irrelevant background noise, effectively addressing the signal-to-noise ratio problem inherent in small object detection [8].

However, the remarkable performance of these models comes with significant data requirements - they typically demand large, diverse, and precisely annotated training datasets that comprehensively capture the variations expected in real-world deployment scenarios [9]. The data dependency creates practical challenges, as acquiring high-quality annotations for small objects is both labour-intensive and prone to human error, while domain shift problems arise when models trained on one dataset underperform when applied to slightly different environments or imaging conditions [10]. To mitigate these limitations, researchers have developed complementary techniques, including advanced data augmentation strategies [11–13], semi-supervised learning approaches, and domain adaptation methods, all while continuing to refine fundamental network architectures through innovations like transformer-based vision models and neural architecture search. The theoretical foundations of DL, including the universal approximation theorem and gradient-based optimization through backpropagation, provide mathematical justification for these empirical successes, while ongoing research in areas like explainable AI and robust learning seeks to address the remaining limitations in reliability and generalization. Together, these advances have positioned deep learning as the dominant approach for object detection across critical applications, including medical diagnostics, remote sensing, autonomous vehicles, and industrial quality control, where the lack of precise identification of minute features can have substantial real-world consequences.

1.5.1 Supervised Learning

Supervised learning is the foundation of ML, where models are trained on meticulously labeled datasets. In the context of object detection, this approach enables models to learn discriminative feature representations by systematically minimizing the discrepancy between predicted outputs and provided annotations through an iterative optimization process using algorithms like gradient descent.

While supervised learning has propelled object detection systems to remarkable performance levels, its application to small object detection introduces unique challenges that stem primarily from data requirements. Creating high-quality labels for small objects (spanning just a few pixels) demands extraordinary annotation precision, as even minor bounding box inaccuracies can significantly impact model performance. This annotation process becomes exponentially more laborious and costly compared to standard object detection tasks. Furthermore, the samples must capture sufficient diversity in terms of object scales, orientations, occlusion patterns, and background contexts to prevent learned biases and ensure robust generalization. Despite these substantial data requirements, supervised learning remains the gold standard for small object detection when adequate training data is available, with modern architectures like Cascade R-CNN [14] and YOLO [15] variants demonstrating exceptional performance by effectively leveraging hierarchical feature learning and multi-scale processing. Their strength lies in learning direct mappings from input patterns to target outputs, assuming the training data comprehensively captures the problem space. However, this assumption becomes difficult to meet as object size decreases and contextual information plays a more critical role relative to the limited visual evidence available.

1.5.2 Evaluation Metrics

Evaluation metrics provide standardized measures to assess and compare model performance across studies. Unlike classification tasks, object detection requires simultaneous evaluation of both object localization (bounding box regression) and classification correctness, making the choice of the metrics particularly nuanced. While no single metric captures all aspects of detection quality, the most widely adopted metric, mean Average Precision (mAP), quantifies detection performance across all object classes by calculating the area under the precision-recall curve at multiple Intersection-over-Union (IoU) thresholds. It estimates how well a model balances precision (avoiding false positives) and recall (identifying true positives) across varying localization strictness [16].

However, relying solely on mAP can obscure important aspects of real-world performance. For example, a model may achieve a high mAP score while still performing poorly on small objects, as standard IoU thresholds (e.g., 0.5) tend to favour larger objects, whereas small localization errors have a relatively smaller impact on the overall IoU. This has led to specialized variants, such as mAP@Small [17], which evaluates performance on objects under 32×32 pixels. Similarly, metrics like Recall@FPI (False Positives per Image) better reflect deployment scenarios where computational resources or user tolerance limit acceptable false positive rates. These nuances are especially critical in safety-sensitive applications like autonomous driving or medical imaging, where missed small objects (e.g., distant pedestrians or micro-lesions) can have severe consequences. To address these limitations, this work adopts multiple evaluation metrics, viz., accuracy, mIoU, precision, recall, and average precision. The accuracy is computed as in (1.1).

$$\text{Accuracy} = \frac{\text{Number of correctly predicted boxes for a class}}{\text{Total number of ground truth boxes for that class}} \times 100. \tag{1.1}$$

In object detection problems, a predicted bounding box is considered accurate based on the IoU criterion (cf. Fig. 1.3), as defined in (1.2). Specifically, the degree of overlap between the predicted and ground truth bounding boxes must exceed a predefined threshold, typically 0.5.

$$\text{mIoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \geq Threshold. \tag{1.2}$$

The precision and recall are defined as:

$$\text{Precision} = \frac{TP}{TP+FP}, \text{and} \quad \text{Recall} = \frac{TP}{TP+FN}, \tag{1.3}$$

with TP, FP, FN denoting true positives, false positives, and false negatives. Hence, the average precision computes the area under the precision-recall curve and is computed as:

$$\text{Average Precision (AP)} = \int_0^1 p(r)\,dr, \tag{1.4}$$

where $p(r)$ is the precision at recall r. The mAP, (1.5), is included to handle class imbalance in object detection tasks and is calculated as the mean of the average precision per class, where AP_i is the Average Precision (AP) for class i, and N is the total number of classes.

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i, \tag{1.5}$$

A commonly accepted mAP is the MSCOCO-standard [18] that uses `mAP@[.5:.95]`, which evaluates the IoU with thresholds from 0.5 to 0.95 in 0.05 increments (τ) defined as:

$$\text{mAP@[.5:.95]} = \frac{1}{9}\sum_{\tau=0.5}^{0.95} \text{mAP@IoU} = \tau. \tag{1.6}$$

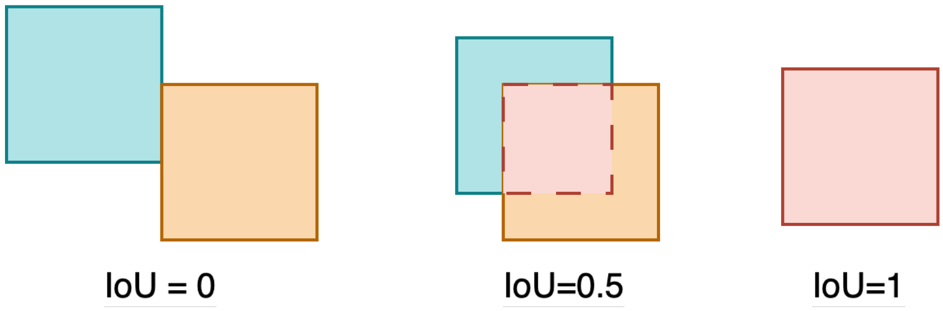


Figure 1.3: A pictorial description of IoU.

1.6 Thesis Contribution

This thesis aims to develop a comprehensive framework for improved small object detection by addressing the critical limitations in existing methodologies. The key contributions of this thesis are as follows:

- **Unified Framework for Semantic-Guided Detection:** Proposes a novel multi-branch architecture that synergistically combines semantic segmentation and object detection features with attention-based fusion, to improve localization and classification of small objects.
- **Enhanced Feature Pyramid for Small Objects:** Introduces a novel feature pyramid-based network that emphasizes long-range directional dependencies through dilated strip-wise convolutions and spatial-aware attention mechanisms.
- **Comprehensive Generalization Techniques:** Extensively explores and documents the use of self-supervised learning as a pretext task to improve generalization, alongside data augmentation techniques to increase data variation to provide more small object instances and challenging samples for robust training.

Chapter 2

Literature Review

2.1 Overview

This chapter presents a systematic review of object detection approaches, beginning with traditional models followed by deep learning-based models and feature fusion strategies. It investigates existing methods specifically built for small object detection, comparing their architectural choices, performance, applicability in real-world scenarios, and limitations. Various approaches, ranging from anchor design and multi-scale feature fusion to attention mechanisms and transformers are examined. The comparative analysis provides key insights into the current state of the field and identifies gaps and trade-offs that inform the motivation for the proposed work in subsequent chapters.

2.2 Object Detection

Over the past two decades, object detection (OD) algorithms have played a pivotal role in advancing autonomous transport and surveillance systems, enabling the development of fully automated solutions [19–24]. The core function of an OD system is to accurately identify and localize target objects within a given image. This process generally involves a combination of feature extraction and classification techniques [21, 25]. Object detection approaches can be broadly categorized into two main types: traditional methods and deep learning-based methods. Traditional approaches depend on manually crafted feature extraction techniques to capture distinctive characteristics of objects, followed by the use of machine learning algorithms for classification [26]. In contrast, deep learning-based methods leverage deep neural networks in an end-to-end manner, allowing the model to automatically learn relevant features and perform classification directly from the input data [27].

2.3 Traditional Object Detection Methods

In the early stages, limited computational resources and insufficient feature representation necessitated the use of hand-crafted features for OD tasks [28]. Traditional approaches typically follow a three-stage pipeline to accomplish this: localization of regions of interest, extraction of discriminative features, and classification of the detected objects.

2.3.1 Localizing Region of Interest

The first stage involves localizing potential regions within the image where objects are likely to appear. Two commonly used techniques for region of interest (ROI) localization are background subtraction and sliding window approaches [29, 30]. Background subtraction is useful in static images to isolate moving objects by comparing each frame to a background model [31]. This is ineffective for small objects which often generate minimal motion or occupy very few pixels, making them indistinguishable from noise or background fluctuations. Furthermore, complex and dynamic backgrounds, such as moving foliage, lighting variations, or weather changes, make it difficult for background subtraction techniques to maintain high accuracy. To address these challenges, various adaptive background modelling techniques were proposed [32–35]. For example, Gaussian Mixture Model (GMM) [35], models each pixel as a mixture of Gaussians representing different observed intensities over time. This allows the model to adapt to environmental changes such as lighting and weather, however, it fails for objects that exhibit non-Gaussian characteristics or rapid unpredictable variations (e.g. appearances of pedestrians). Alternatively, the sliding window technique [36] provides a background-independent method involving scanning the entire image using windows of various sizes and aspect ratios to identify candidate regions. While effective in ensuring that even the smallest objects can be detected, this method suffers from high computational cost, especially since detecting small objects requires scanning the image at finer scales with more densely overlapping windows. This makes real-time implementation challenging where speed is crucial. Sensitivity to background complexity and computational inefficiency limits background subtraction and sliding window methods, highlighting the need for more advanced and efficient approaches.

2.3.2 Extracting Features

Once an object is localized, the feature extraction stage generates a distinct representation that enables accurate classification. Early methods relied on basic features such as color and shape, which can be unreliable because they are superficial and lack necessary semantic and spatial context [37, 38]. To address this, hand-crafted descriptors like Histogram of Oriented Gradients (HOG) [39], Scale-Invariant Feature Transform (SIFT) [40], and Haar-like features [41] have been used. Nevertheless, even these features face challenges in detecting objects within cluttered or low-resolution images, motivating the shift toward learning-based feature extraction techniques.

2.3.3 Object Classification

The final stage involves object classification, where the detected region is assigned a category label. Traditional methods often use machine learning algorithms like Support Vector Machines (SVM) [42] and Random Forests (RF) [43] for this task. SVMs classify objects by finding the optimal decision boundary in the feature space, while RFs combine multiple decision trees for robust predictions. Other techniques, such as AdaBoost [44] and Deformable Part-based Models (DPM) [45], have also been used. Regardless, classification accuracy heavily depends on the quality of hand-engineered features and regions of interest, which require substantial domain expertise and time investment [46].

2.4 Deep Learning-based Object Detection Methods

The challenges that each stage brought forth, catalyzed a shift toward deep learning-based methods, which reimagined the stages using data-driven learning, leading to a shift in modern object detection. Fig. 2.1 shows the systematic categorization of deep learning-based object detectors.

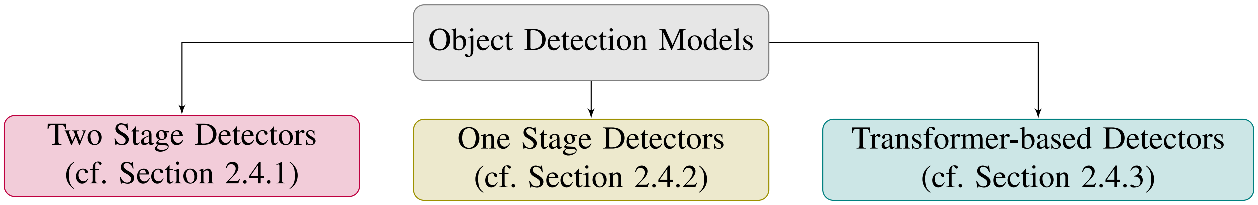

Object Detection Models
Two Stage Detectors (cf. Section 2.4.1)
One Stage Detectors (cf. Section 2.4.2)
Transformer-based Detectors (cf. Section 2.4.3)

Figure 2.1: A categorization of deep learning-based object detection models.

2.4.1 Two Stage Detectors

Two-stage detectors adopt a region-proposal-driven approach, separating detection into two distinct stages as seen in Fig. 2.2. Prominent examples of two-stage detectors include the influential R-CNN family (R-CNN, Fast R-CNN, Faster R-CNN, Cascade R-CNN, etc.) that utilize advanced region proposal and refinement techniques.

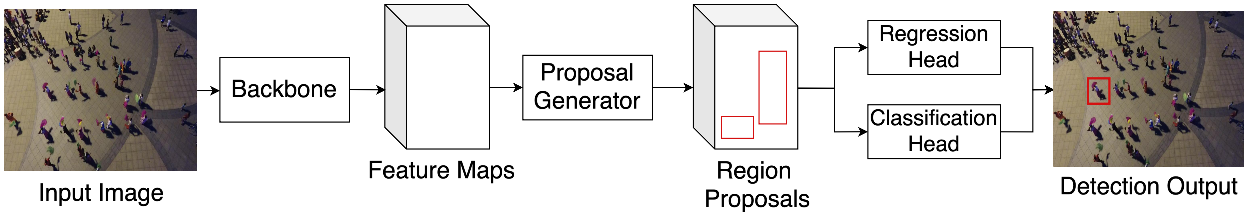


Figure 2.2: Overview of two-stage object detection model.

R-CNN [47] introduced a method that tackled the challenge of proposing multiple regions for object detection using the selective search algorithm that generated approximately 2000 region proposals per image, which were then passed through a CNN to extract features. These features were fed into a support vector machine (SVM) which determined the presence of an object and computed four offset values to improve the precision of the bounding boxes. However, the per-sample inference time is nearly a minute using this approach, making it inefficient for practical applications. To overcome this drawback, Fast R-CNN [48] was developed where the entire input image was passed to the CNN for feature extraction instead of individually passing each region proposal. Regions of interest were then extracted from the feature map using a region of interest (RoI) pooling layer that reshaped these regions for further processing. This method vastly improved efficiency, as only a single image was fed into the CNN, rather than thousands of individual regions, making predictions faster and more scalable. Both R-CNN and Fast R-CNN relied on the manual process of selective search to generate region proposals. Instead, Faster RCNN [49] proposed a Region Proposal Network (RPN) that learned how to predict region proposals directly. Unlike selective search, the RPN was trainable and generated high-quality proposals, which were further refined by a RoI pooling layer to classify objects and predict bounding box coordinates in a single, streamlined framework.

The Cascade R-CNN [14] is a notable advancement among two-stage object detection methods, specifically addressing the mismatch problem between proposal quality and detection performance in prior two-stage frameworks. Cascade R-CNN introduces a multi-stage pipeline, where bounding box proposals undergo successive refinements through multiple detection heads, each trained with progressively stricter Intersection-over-Union (IoU) thresholds. This cascaded refinement significantly improves object localization accuracy by iteratively narrowing predictions and alleviating issues caused by poorly aligned bounding boxes. Consequently, Cascade R-CNN consistently achieves superior detection performance compared to standard Faster R-CNN, particularly excelling in scenarios requiring highly accurate localization. However, these performance gains come with increased computational complexity due to the sequential refinement stages.

2.4.2 One Stage Detectors

One-stage object detectors surpass two-stage methods in real-time object detection by unifying localization and classification into a single network pass (cf. Fig. 2.3). One-stage detectors predict bounding boxes and class probabilities directly from input images in a single forward pass without needing region proposals or iterative refinements. As a result, they are widely used for applications requiring fast inference, such as surveillance, autonomous driving, and robotics. One-stage detec-

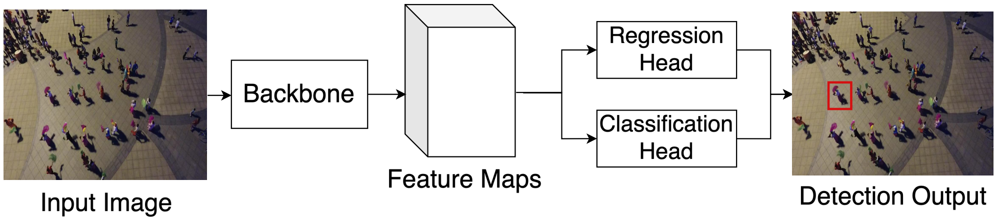


Figure 2.3: Overview of one-stage object detection model.

tors generally have a backbone like ResNet [50] or HRNet [51] for feature extraction, followed by a detection head, and then non-maximal suppression (NMS) applied to filter redundant predictions. One-stage detectors are further categorized into Anchor-Based Detectors and Anchor-Free Detectors.

Anchor-Based Detectors

Anchor-based methods use predefined anchor boxes (i.e., reference bounding boxes) to guide detection, where anchors of varying scales/aspect ratios are placed on feature maps, and the network then predicts offsets to adjust the anchors into final detections. Single Shot Detector (SSD) [52] utilizes this and employs feature maps at multiple scales to effectively detect objects of varying sizes. Due to its streamlined design, SSD achieves real-time detection speeds, making it suitable for time-sensitive applications. However, SSD generally exhibits slightly lower detection accuracy for small objects, compared to more computationally intensive two-stage methods.

You Only Look Once (YOLO) [15] is a well-known single-stage object detection framework known for its high inference speed and simplicity. YOLO treats object detection as a unified regression problem, predicting bounding boxes and class probabilities directly from image pixels using a single convolutional neural network. Although earlier versions, like YOLOv1 and YOLOv2, emphasized speed at the expense of accuracy, YOLOv3 [53] significantly improved detection performance by incorporating multi-scale feature predictions and improved anchor boxes. Since then, YOLO has evolved through multiple iterations, each improving detection accuracy, inference speed, and training stability. As of now, YOLOv11 [54] uses refined anchor selection strategies, an efficient backbone, and advanced feature fusion techniques.

Previous methods failed to address the issue of class imbalance till RetinaNet [55] introduced a novel focal loss function, designed to effectively down-weight the importance of easily classified examples, thereby placing greater emphasis on difficult-to-detect instances. RetinaNet combines this focal loss with a feature pyramid network (FPN) to leverage multi-scale feature representations, enhancing detection accuracy across various object scales. Consequently, RetinaNet achieves accuracy comparable to two-stage detectors like Faster R-CNN while maintaining single-stage efficiency. Yet, it remains sensitive to hyperparameter tuning, and its training can be computationally demanding due to the complexity introduced by focal loss optimization.

GFL [56] builds upon RetinaNet by further addressing the limitations associated with focal loss and classification-regression separation in object detection. While RetinaNet effectively mitigates class imbalance, it still treats classification and bounding box regression as independent tasks. GFL innovatively integrates quality estimation directly into classification scores, merging classification and localization branches into a unified prediction. This joint approach allows the model to estimate object presence and localization quality simultaneously, thus achieving more accurate and stable detection results. Experimental evaluations demonstrate that GFL consistently surpasses RetinaNet and other state-of-the-art single-stage detectors. Nevertheless, GFL's integration of multiple prediction tasks into one branch lead to increased complexity during training and inference, necessitating careful optimization and hyperparameter tuning.

Adaptive Training Sample Selection (ATSS) [57] takes a different perspective and addresses the critical problem of anchor assignment in single-stage object detectors. Instead of manually tuned thresholds for positive and negative anchors, ATSS dynamically selects training samples by adaptively computing optimal thresholds based on anchor statistics. This effectively reduced anchor ambiguity and enhanced detection accuracy without additional computational overhead. Despite its success, its reliance on statistical heuristics for sample selection may result in poor generalization across diverse datasets.

Anchor-Free Detectors

Transitioning from anchor-based to anchor-free single-stage detectors marks an important evolution in single-stage object detection literature. Anchor-based methods rely heavily on predefined anchor boxes, introducing complexities in anchor generation, hyperparameter tuning, and assignment strategies. To address these limitations, recent advancements have proposed anchor-free approaches, which directly predict bounding boxes without relying on pre-set anchor boxes. This simplifies the detection pipeline and reduces hyperparameter sensitivity.

The pioneering anchor-free OD framework, DenseBox [58], treats detection as a dense per-pixel prediction task. It employs a fully convolutional network (FCN) that simultaneously regresses bounding box coordinates and predicts object confidence scores for every spatial location in an image. DenseBox directly maps the receptive field of each spatial location to a potential object region without needing carefully designed and tuned anchors. However, dense predictions on all locations and scales of an image lead to increased computation and memory usage, and redundant predictions require further post-processing.

CornerNet [59] identifies objects by detecting pairs of key points representing the top-left and bottom-right corners of bounding boxes. It works by predicting corner heatmaps and embeddings to pair corners belonging to the same object, and then offsets to refine bounding box locations. Despite achieving strong accuracy, CornerNet can exhibit relatively slower inference speeds due to the post-processing complexity required to match corner pairs.

Conversely, Fully Convolutional One-Stage Object Detection (FCOS) [60], predicts bounding boxes by regressing distances from each pixel location to the four sides of a bounding box, and introduces a center-ness branch that assesses the quality of detection by estimating the proximity of predicted boxes to object centers. While this enhances general localization accuracy, it struggles with occluded and irregularly shaped objects because it relies on ’centerness’ for predictions.

Similarly, CenterNet [61] formulates object detection as a key point estimation problem by predicting center heatmaps and regressing object dimensions and offsets directly from these centers. However, as with FCOS, its dependence on center responses limits its performance on occluded, small, elongated or irregular objects.

2.4.3 Transformer-based Detectors

Transformer-based object detectors represent a recent paradigm shift in OD tasks, leveraging attention mechanisms to model long-range dependencies and contextual information more effectively than traditional CNN-based approaches. The DEtection TRansformer (DETR) [62] reimagines detection as a set prediction problem using a transformer encoder-decoder architecture. While conventional deep learning detectors rely on anchor boxes, region proposals, or hand-crafted post-processing techniques like NMS; DETR adopts an end-to-end design that predicts a fixed set of object bounding boxes and class labels directly from image features. It uses a CNN backbone (typically ResNet) to extract feature maps, which are passed to a transformer encoder. The transformer decoder receives these encoded features along with a set of learned positional embeddings called ”object queries”, each responsible for predicting a single object in the scene. The model outputs a fixed number of predictions, and the optimal matching between predictions and ground truth is handled using the Hungarian algorithm, minimizing a bipartite matching cost. While DETR is conceptually elegant and simplifies the detection pipeline, it suffers from slow convergence and struggles with detecting small objects due to the global attention’s coarse spatial resolution. Subsequent works, such as Deformable DETR [63] and Swin Transformer-based detectors [64, 65], have addressed initial limitations related to training efficiency and scalability, slightly improving convergence speed and detection accuracy.

Despite transformer strengths at capturing global context and improving localization accuracy, they often require substantial computational resources and careful optimization [66]. The global attention mechanism has quadratic complexity with respect to image size [67] and lack strong local inductive biases which are crucial for fine-grained feature extraction. Components such as object queries, positional encodings, bipartite matching, and set-based loss functions introduce architectural complexity, requiring careful tuning and design. Consequently, these methods rely on large-scale datasets [68] to perform optimally, making them less suitable for low-data scenarios where CNN-based models perform better due to their built-in spatial priors. As a result, transformer-based OD methods are challenging to scale for high-resolution inputs or real-time applications and their complexity can hinder adaptability in custom or domain-specific detection tasks.

2.5 Feature Fusion for Object Detection

The traditional FPN introduced in [69] employs a top-down architecture with lateral connections to merge high-level semantic features from deeper layers with spatially rich features from shallower ones. This design improves the detection of objects at multiple scales by enriching lower-resolution layers with semantic information. Yet, FPN's uni-directional top-down flow may still limit the preservation of fine-grained spatial details, particularly in deeper layers.

To address this, PANet [70] extends FPN by introducing a bottom-up path augmentation, enhancing the feature maps with spatially precise localization signals. This dual-path structure allows deeper layers to retain important spatial cues while also benefiting from the semantic enhancement of the top-down pathway. Building upon this, the Adaptive Feature Pyramid Network (AFPN) [71] incorporates an iterative cross-scale refinement strategy. Further advancements are seen in the Bidirectional Feature Pyramid Network (BiFPN), introduced in EfficientDet [72], which allows bidirectional information flow across feature levels, facilitating more effective multi-scale feature fusion. Unlike standard FPNs, which use simple feature addition, BiFPN integrates learnable weighted fusion—where the network learns the relative importance of each feature input using attention-like weights. It also removes less useful nodes and simplifies the pathway to maintain computational efficiency while boosting accuracy. These design choices significantly enhance feature reuse and detection precision without a major increase in inference cost, making BiFPN a core component of many efficient detection architectures.

In this work, recent advancements like the bidirectional weighted fusion in BiFPN and the enhanced spatial flow in PANet, inspire the development of a more effective integration of multi-scale semantic and spatial features for small object detection.

2.6 Existing Works on Small Object Detection

Detection architectures are generally optimized for medium-to-large objects in fixed-perspective ground-view datasets like MS COCO [17] or PASCAL VOC [73]. As a result, very few methods explore the specific task of improving small object detection. For example, BiKD-Yolo [74] introduces an enhanced YOLO-based model tailored for small object detection in Unmanned Aerial Vehicle (UAV) imagery. The model integrates BiFormer attention mechanisms and knowledge distillation techniques to improve detection performance, allowing the capture of long-range dependencies and contextual information. Despite efforts to maintain efficiency, the integration of BiFormer attention mechanisms and knowledge distillation processes adds significant computational overhead, impacting real-time performance in resource-constrained UAV systems. Moreover, implementing knowledge distillation requires careful selection and training of teacher-student network pairs, which can be complex and time-consuming.

In another approach, Attention Enhanced Feature Fusion Network [75] introduces a novel architecture that integrates a hybrid attention module designed to enhance feature extraction capabilities by combining multi-axis frequency and spatial attention mechanisms. However, the integration of complex attention mechanisms introduces additional computational costs and inference speed.

Conversely, UAV-DETR [76] is an efficient end-to-end object detection method that incorporates multi-scale feature fusion with frequency enhancement and frequency-focused down-sampling. However, its performance relies on large datasets for optimization. TPH-YOLOv5 [20] uses a transformer prediction head and CBAM to build upon the YOLOv5 architecture. Transformer modules tend to consume more GPU memory, particularly with high-resolution images or large batch sizes.

A summary of the models discussed in this section is provided in Tabel 2.1.

2.7 Chapter Summary

This chapter presented a comprehensive review of object detection methodologies. A key finding is that a significant improvement in detection accuracy and generalization can be achieved using deep learning combined with enriched feature hierarchies. The review highlighted feature fusion techniques' critical role in improving objects' representation at multiple scales.

Focusing specifically on small object detection, the chapter identified several targeted strategies. While these approaches have led to measurable performance gains, they also reveal persistent

Table 2.1: A Summary of Object Detection Model Performance on Small Objects

Ref.	Methodology	Limitations	Metric (mAP)
[20] 2023	Leverage transformer prediction head for detection small and densely packed objects	Large model size, computationally heavy and data hungry.	39.2%
[76] 2025	Hybrid model that combines the strengths of encoder architectures and vision transformers	Uses a large backbone and transformer encoder-decoder architecture, data-hungry.	31.5 %
[75] 2025	Hybrid attention module designed to enhance feature extraction capabilities for small objects by combining multi-axis frequency and spatial attention mechanism	Integration of complex attention mechanisms introduces additional computational costs, affecting real-time processing capabilities.	34.0%
[74] 2024	Integrates BiFormer attention mechanisms and knowledge distillation techniques to improve detection performance.	Despite efforts to maintain efficiency, the integration of BiFormer attention mechanisms and knowledge distillation processes may still introduce additional computational overhead, impacting real-time performance.	29.8%

limitations. Most notably, existing models still struggle with low-resolution object features, high background noise, and significant scale variation. Additionally, improvements in accuracy are frequently accompanied by increased computational demands, limiting their deployment in real-time or resource-constrained settings. Many methods also rely heavily on large annotated datasets, which are often difficult to obtain for small objects.

Overall, while the field has made substantial progress, the reviewed literature underscores a clear trade-off between accuracy, complexity, and scalability. These insights motivate the need for a more balanced solution, forming the foundation for the novel frameworks proposed in the following chapters.

Chapter 3

Enhancing Small Object Detection via Semantic Feature Fusion

3.1 Overview

Driven by the exponential integration of digital sensors in automated systems, OD has become fundamental to numerous real-world applications. In autonomous driving, for instance, reliable detection of pedestrians, vehicles, and traffic signs is essential for ensuring safety. Regardless, detecting small, distant objects in complex traffic scenes remains a significant challenge.

This chapter introduces a novel feature fusion approach that enhances small object detection by integrating semantic segmentation features with features from a primary detection backbone. The proposed system leverages complementary strengths from both detection and segmentation models, the latter being a pixel-wise classification task, captures fine-grained spatial details, enabling the model to better understand scene context and object boundaries.

The effectiveness of this approach is validated through ablation studies on the KITTI benchmark dataset. This hybrid architecture could pave the way for future research in robust object detection, particularly in complex environments where small or distant objects are prevalent. The findings may inspire further exploration of multi-task learning frameworks that unify detection and segmentation for enhanced visual perception.

3.2 Background Concepts and Methodology

As discussed in Chapter 2, supervised object detection models leverage labeled data to achieve high-precision localization and classification. Nevertheless, they often struggle with small or distant objects due to limited pixel-level information and scale variations, particularly in complex scenarios in driving and urban scenes. While multi-scale architectures like FPN and PANet [77] mitigate this to some extent, their reliance on labeled data makes them vulnerable to class imbalance (e.g., rare object categories) and annotation scarcity for small instances.

On the other hand, unsupervised or self-supervised approaches can learn robust feature representations from unlabeled data, capturing fine-grained spatial details critical for small objects. Yet, unsupervised methods often lack task-specific discriminative power, leading to false positives (e.g., misclassifying background clutter as objects) or poor generalization across diverse environments.

Given these complementary strengths and weaknesses, hybrid approaches that fuse supervised detection backbones with unsupervised or multi-task features (e.g., semantic segmentation) show significant promise. For instance, segmentation models trained in a self-supervised or weakly supervised manner extract rich pixel-wise semantic features, enhancing a detector's ability to localize small objects by reinforcing spatial coherence and edge awareness. Recent work demonstrates that integrating segmentation-aware features into detectors like Mask R-CNN and YOLO-LOGO improves performance on small objects without requiring additional labeled data [78, 79]. Similarly, cross-modal fusion (e.g., LiDAR + RGB) leverages unsupervised pre-training to boost detection robustness in low-visibility scenarios [80]. This suggests that hybrid architectures, combining the discriminative power of supervised detection with the generalizable, fine-grained features from segmentation, could address key challenges in small object detection, particularly for dynamic, real-world settings where labelled data is scarce or imbalanced.

3.2.1 Inspiration for the Proposed Model

This work draws inspiration from three core paradigms in computer vision: (i) attention-based feature fusion for multi-modal learning, (ii) cross-task knowledge transfer via frozen backbones, and (iii) efficient reuse of pre-trained representations for downstream tasks. This approach bridges domain-specific feature spaces while maintaining computational efficiency by leveraging a frozen semantic segmentation backbone to enhance object detection.

3.2.2 Attention Mechanism for Feature Fusion

Networks incorporating attention-based modules have shown promising results in NLP tasks [81] and have since been adapted for object detection to improve accuracy as seen in works like [82–84]. By explicitly modelling contextual relationships and dependencies, attention enhances the representational capacity of convolutional networks, especially in scenarios where spatial or semantic cues are dispersed or ambiguous. When combined with architectural components like feature pyramids or dilated convolutions, attention mechanisms serve as a powerful tool to reinforce multi-scale reasoning and spatial precision. In convolutional networks, attention can be applied across spatial dimensions, channels, or both, allowing models to learn dynamic weighting schemes that enhance feature representations based on contextual relevance. The spatial attention mechanism is mathematically expressed in (3.1), and can model long-range dependencies and correlations to enhance the task-specific contextual information.

$$\begin{aligned} &A = \sigma((U_qF)(U_kG)^T), \text{ where } \sigma(x) = \frac{1}{1+e^{-x}}, \\ &X = A(U_vG), \end{aligned} \tag{3.1}$$

where U_q, U_k, and U_v are learnable weight matrices used to compute the query, key, and value representations. The query and transposed key representation are used to compute the attention score matrix A, which is sigmoid normalized to produce a matrix of attention weights. These weights are used to weight the value representation (U_vG), resulting in attention refined output X which emphasizes the most relevant features in G and is integrated back into the network.

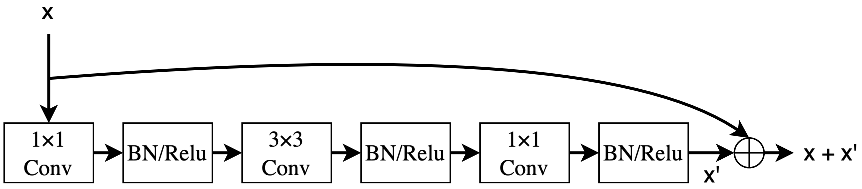


Figure 3.1: High-level representation of the Resnet bottleneck structure.

3.2.3 Backbone and Feature Pyramid

Table 3.1 presents the layer-wise architectural details of the ResNet-50 [50] backbone employed in the proposed framework. The backbone is composed of four sequential blocks, which progressively extract features at multiple levels of abstraction. These blocks output feature maps with channel dimensions of 256, 512, 1024, and 2048, capturing both low-level spatial details and high-level semantic information. The internal structure of these residual blocks follows the bottleneck design, as illustrated in Fig. 3.1, where a series of 1×1, 3×3, and 1×1 convolutions are used to reduce, process, and then expand the feature dimensionality. By leveraging this hierarchical representation, the model is able to learn rich features essential for robust object detection. The outputs from these blocks are then passed to the FPN [69] shown in Fig. 3.2, which performs multi-scale feature fusion through top-down pathways and lateral connections. This fusion enhances the net-

Table 3.1: Architectural Details of the ResNet-50 Backbone

Layer ID	**Layer Type**	**Repeat**	**Output Dimension (B, C, H, W)**
Input	Input Layer	–	(8, 3, 375, 1242)
L1	Conv2D	–	(8, 64, 188, 621)
L2	BatchNorm2D	–	(8, 64, 188, 621)
L3	ReLU	–	(8, 64, 188, 621)
L4	MaxPool2D	–	(8, 64, 94, 311)
Block1	Bottleneck (256-d)	×3	(8, 256, 94, 311)
Block2	Bottleneck (512-d)	×4	(8, 512, 47, 156)
Block3	Bottleneck (1024-d)	×6	(8, 2048, 24, 78)
Block4	Bottleneck (2048-d)	×3	(8, 2048, 12, 39)

Key Details:

Description	**Value**
Total Trainable Parameters	23,508,032
Kernel Sizes	L1: 7×7, Bottleneck: 1×1, 3×3, 1×1
Strides	L1: 2, MaxPool: 2, Bottleneck: 1 or 2 (downsampling)
Padding	L1: 3, Bottleneck: 1 (for 3×3 layers)

work's ability to detect objects of varying sizes, with particular benefit to small object detection by preserving fine-grained details while incorporating contextual information from deeper layers.

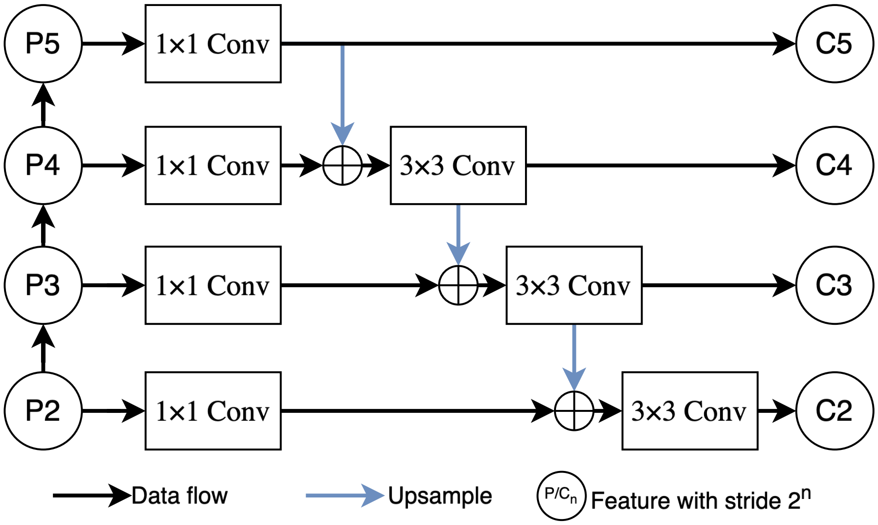


Figure 3.2: Architecture of the feature pyramid network.

3.3 The Proposed Attention-based Segmentation-aware Model

Fig. 3.3 illustrates the proposed framework. It integrates a semantic segmentation model (sem-seg) with an OD model. The features from the sem-seg model are fused with the OD model's pipeline to refine the learned features. For this purpose, the DeepLabv3 [85] with ResNet-101 [50] backbone was used in the sem-seg model because (i) it has a similar backbone structure to the OD sub-network, thus enabling symmetric feature fusions, and (ii) DeepLabv3 has atrous convolutions and pyramid pooling that preserve global context. Since semantic segmentation operates at the pixel level, it captures fine-grained local spatial detail at the deeper layers. Thus, the features from the sem-seg backbone inform the OD model about object-scene interaction beyond just local pixel relationships. Additionally, an attention-gating mechanism is deployed in each fusion stage, enabling the OD model to selectively focus on important features while ignoring insignificant details. In Fig. 3.3, only the backbone of the sem-seg model is shown, and later layers are omitted as their features are not utilized.

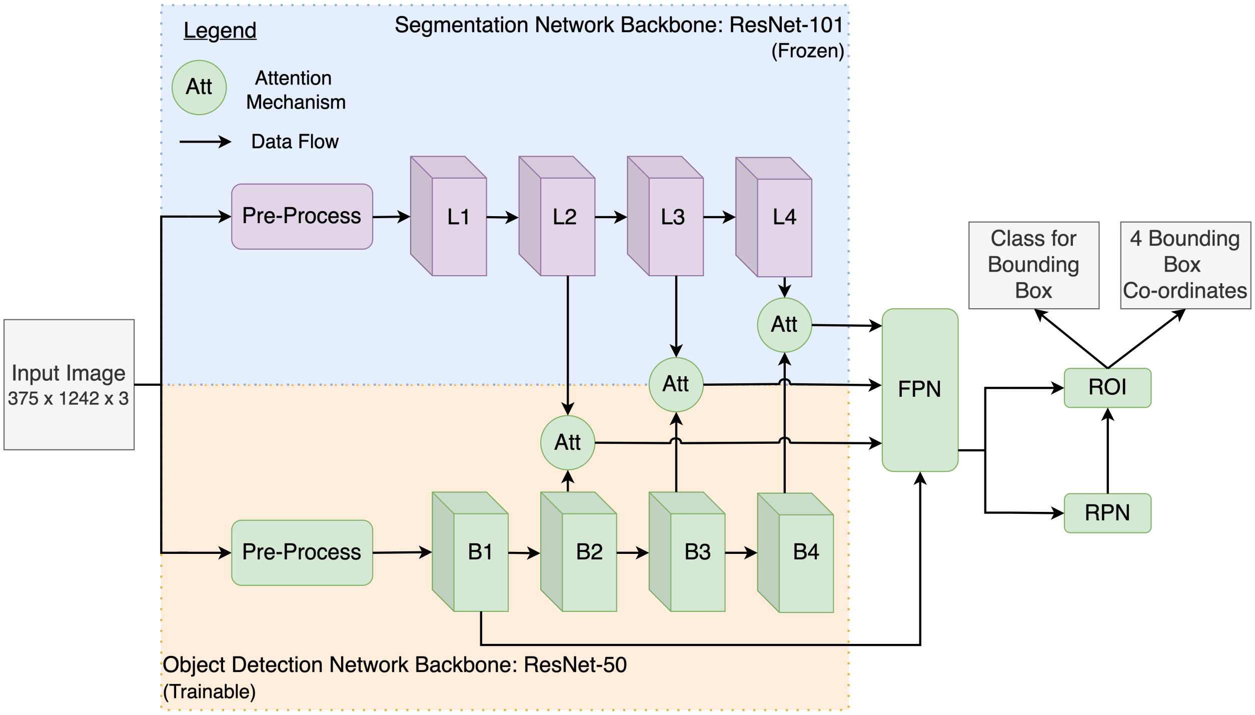


Figure 3.3: A high-level representation of the proposed framework. L1-L4 represents the Segmentation Network Backbone, and the Object Detection Framework is in green.

The OD model takes an input image of size 375×1242×3 and processes it via the backbone, consisting of four layers. As the image progresses through each layer, the feature maps are generated double while the spatial dimension (height and width) is halved. The final layer of the backbone generates dense feature maps with the dimension 13×42×2048. Simultaneously, the same input image is passed through the segmentation path, where features of ResNet layers 2, 3, and 4 are extracted. These features are then fused with the corresponding layer outputs of the object detection model and passed through the attention-gating mechanism described earlier. Finally, the attention-refined features are passed to the FPN layer that merges lower-level clues with higher-level clues to create feature maps containing both rich semantic and spatial information. The output of the FPN is then passed to the RPN, which generates multiple anchor boxes and then, based on the objectness score and predicted bounding box offset, classifies them as a correct region or not. The RoI layer then takes these predicted regions and the feature pyramids, and for each RoI, it extracts a fixed-size feature map using pooling to refine the prediction to give the final output.

3.4 Model Training and Optimization Strategies

3.4.1 Dataset

For model training and validation, this study uses the publicly available KITTI dataset [86]. It contains 7481 camera images captured in Karlsruhe, Germany. The images are annotated with bounding boxes for eight classes defined as 'Car,' 'Van,' 'Truck,' 'Pedestrian,' 'Person (sitting),' 'Cyclist,' 'Tram,' and 'Misc' (e.g., Trailers, Segways). The advantage of using this set is the sample variation, which ensures a robust model development, resulting in better generalization and applicability to real-life scenarios. The training set was split into 5984 samples for model building and 1497 samples for evaluation. The samples are resized to 375×1242 while maintaining their aspect ratio, and their pixel values are normalized to $[0, 1]$. Cityscapes [87] and KITTI semantic segmentation data are used to fine-tune the segmentation model. Cityscapes was chosen because it closely resembles KITTI's samples.

3.4.2 Training Strategy of the Proposed Method

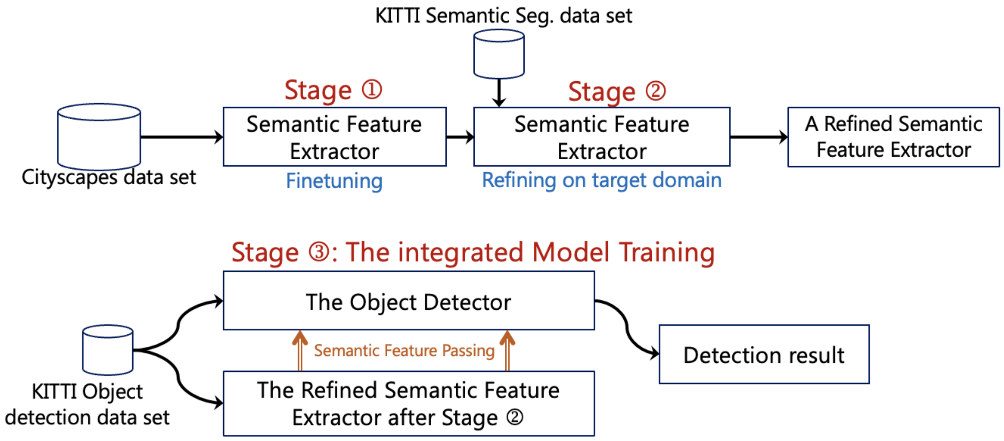

KITTI Semantic Seg. data set
Stage ①
Stage ②
Semantic Feature Extractor
Semantic Feature Extractor
A Refined Semantic Feature Extractor
Cityscapes data set
Finetuning
Refining on target domain
Stage ③: The integrated Model Training
The Object Detector
Semantic Feature Passing
Detection result
KITTI Object detection data set
The Refined Semantic Feature Extractor after Stage ②

Figure 3.4: Training strategy of the proposed architecture.

The proposed model was trained in a multi-stage process as shown in 3.4. First, the DeepLabv3 semantic segmentation model was pretrained on the CityScapes dataset (Stage 1). This pretrained model was then fine-tuned on the KITTI semantic segmentation dataset to adapt it to the target domain (Stage 2). After fine-tuning, the semantic segmentation features extracted from the backbone of this model were used to train the object detector (Stage 3). The training progress for each stage is illustrated in the following figures: Figure 3.5 shows the initial CityScapes training, Figure 3.6 displays the fine-tuning on KITTI, and Figure 3.7 presents the final object detector training. All models were trained for a maximum of 150 epochs with early stopping (patience = 8) to ensure consistent and fair comparisons across experiments.

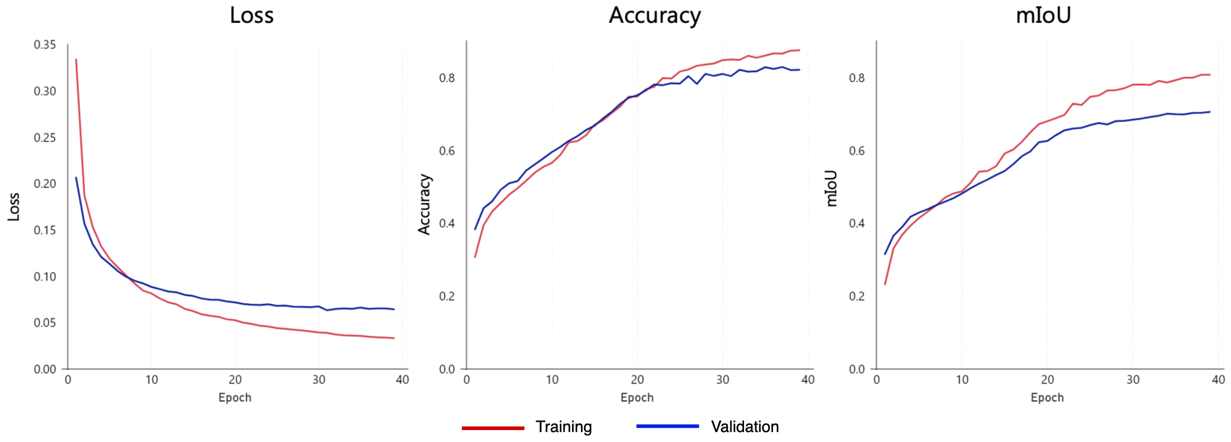


Figure 3.5: Training plots for the semantic segmentation model on the CityScapes dataset.

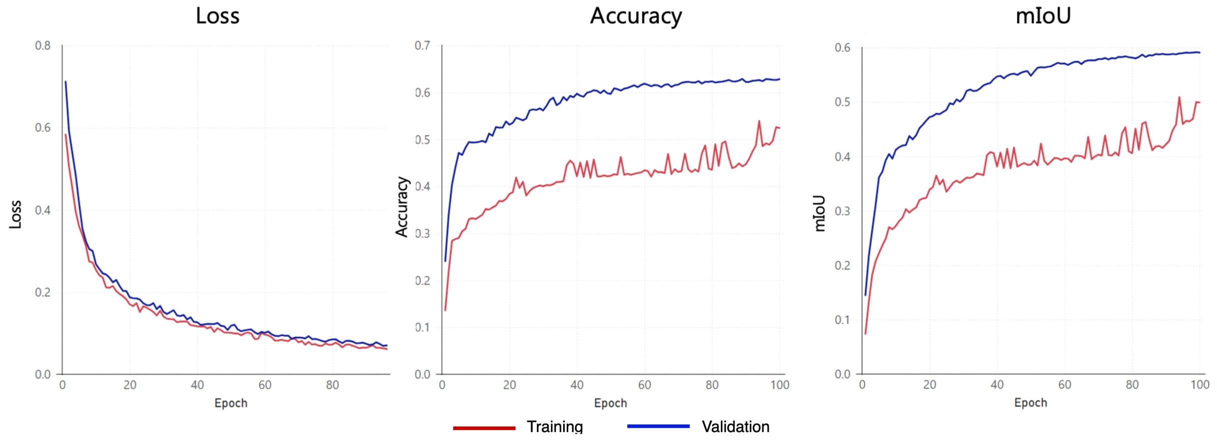


Figure 3.6: Training plots for the semantic segmentation model on the KITTI sem-seg dataset.

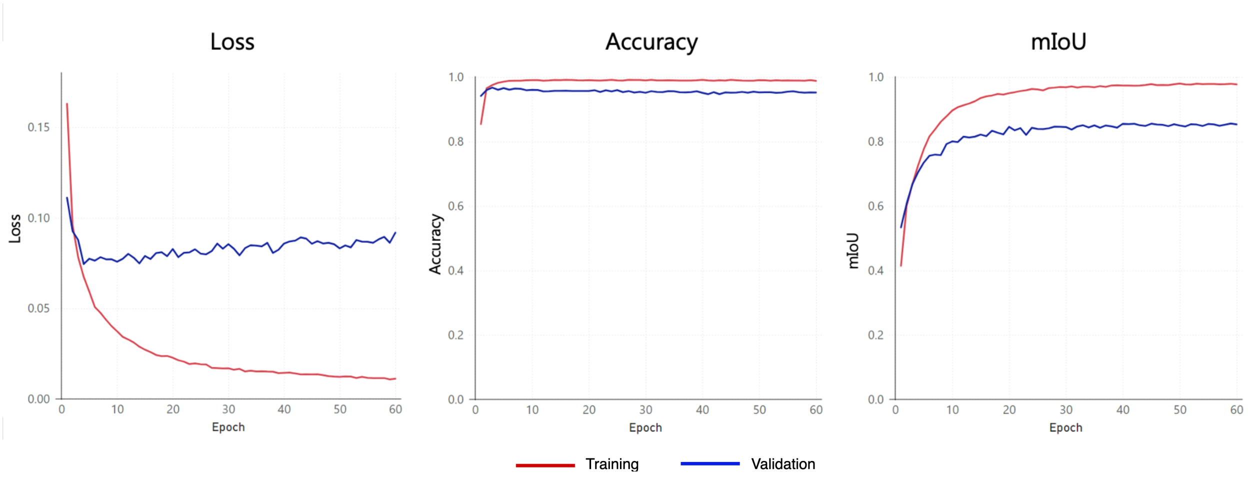


Figure 3.7: Training plots for the proposed object detection model on the KITTI detection dataset.

The training curves exhibit stable convergence with no signs of divergence or erratic fluctuations, indicating well-chosen hyperparameters (e.g., learning rate, batch size) and effective regularization. Convergence is achieved by epoch 50–60 for OD model.

Loss Functions

The training process involves multiple loss functions across different stages of the model. For semantic segmentation, the standard **cross-entropy loss** ($\mathcal{L}_{\text{CE}}$) is used:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[\sum_{c=1}^{C} y_{i,c}\log(p_{i,c})\right] \tag{3.2}$$

where N is the number of samples, C is the number of classes, $y_{i,c}$ is the ground-truth label (1 if sample i belongs to class c), and $p_{i,c}$ is the predicted probability for sample i and class c.

For object detection, the **Faster R-CNN loss** combines three components:

$$\mathcal{L}_{\text{Faster R-CNN}} = \mathcal{L}_{\text{RPN}} + \mathcal{L}_{\text{bbox}} + \mathcal{L}_{\text{cls}} \tag{3.3}$$

- **Region Proposal Network (RPN) loss** ($\mathcal{L}_{\text{RPN}}$):

$$\mathcal{L}_{\text{RPN}} = \frac{1}{N_{\text{cls}}}\sum_{i}\mathcal{L}_{\text{cls}}(p_i, p_i^*) + \lambda\frac{1}{N_{\text{reg}}}\sum_{i} p_i^*\mathcal{L}_{\text{reg}}(t_i, t_i^*) \tag{3.4}$$

where N_{cls} and N_{reg} denote the number of anchors in classification and regression, respectively, p_i is the predicted objectness probability, p_i^* is the ground-truth label (1 for object, 0 for background), t_i and t_i^* are the predicted and ground-truth bounding box coordinates, and λ is a balancing parameter.

- **Bounding box regression loss** ($\mathcal{L}_{\text{bbox}}$):

$$\mathcal{L}_{\text{bbox}} = \sum_{i \in \{\text{pos}\}} \mathcal{L}_{\text{smooth } L_1}(t_i, t_i^*) \tag{3.5}$$

where $\{\text{pos}\}$ indicates positive anchors, and $\mathcal{L}_{\text{smooth } L_1}$ is defined as:

$$\mathcal{L}_{\text{smooth } L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1, \\ |x| - 0.5 & \text{otherwise.} \end{cases} \tag{3.6}$$

- **Classification loss** ($\mathcal{L}_{\text{cls}}$):

$$\mathcal{L}_{\text{cls}} = -\sum_{c} y_c \log(p_c) \tag{3.7}$$

where C is the number of classes, y_c is the one-hot encoded ground truth, and p_c is the predicted softmax probability for class c.

Minimizing this composite loss ensures joint optimization of proposal generation, localization, and classification.

3.4.3 Environment

All models were developed using Python 3.11.5 with PyTorch 2.1 and TorchVision frameworks. The experiments were executed on Compute Canada's Narval cluster, utilizing four NVIDIA A100-SXM4-40GB GPUs (40GB HBM2 VRAM each) with 3rd Gen AMD EPYC processors and 128GB system memory.

3.5 Experimental Analysis

The comparative analysis of the models built and tested in this work is summarized in Table 3.2 alongside the baseline. Here, the fine-tuned Faster R-CNN with a ResNet-50 backbone and FPN layer did not perform better than the baseline Faster R-CNN with a VGG backbone. It achieved an overall mAP of 78.7% as opposed to the baseline, which achieved 81.6% overall mAP. This suggests that simply adopting a more modern backbone (ResNet) without architectural adjustments may not suffice, likely due to:(i) Feature misalignment; ResNet's deeper hierarchical features might not align optimally with Faster R-CNN's RPN for this specific task, whereas VGG's shallower features could generalize better.(ii) Training dynamics; the baseline VGG model may have benefited from longer convergence or better hyperparameter tuning, as ResNet architectures often require careful learning rate scheduling. To further investigate potential improvements, an experimental analysis was conducted, focusing on segmentation feature fusion. Different fusion strategies were implemented to determine the best possible feature fusion approach. Initially, all four ResNet layer outputs of the DeepLabv3 backbone were fused with the Faster R-CNN backbone by concatenating the respective layers, which resulted in a validation mAP of 71.5%. Alternatively, attention gating was applied instead of concatenation at the feature fusion stage to refine the information that is propagated through the object detection network. This approach resulted in a mAP of 80.1%. Subsequently, identifying which layers provide the most effective features was done experimentally by fusing one layer between the sub-networks at a time, starting with Layer 1. All combinations were tried, and the best possible combination was fusing layers 2, 3, and 4. This model (SegAttnDetec) achieved an overall mAP of 83.5%, surpassing all other experimental models. The success of SegAttnDetec highlights two key insights: (i) Feature selection matters more than quantity; overloading the detector with all backbone layers harms performance. (ii) Task-specific adaptation; attention mechanisms bridge the gap between segmentation and detection features, which are often optimized for different objectives.

3.5.1 Quantitative Analysis

Table 3.2 shows SegAttnDetec's overall and category-wise performance against the baseline and existing models, in terms of the mAP. Table 3.3 compares SegAttnDetec's performance w.r.t the major classes in three difficulty categories; easy, moderate, and hard. These categories are determined based on how far the object is, the size of the object, and the occlusion level. It achieved an overall mAP of 83.5%, which is a 2.4% improvement from the baseline. Comparing this work with a similar execution in [88], it is evident that this model performs exceptionally well on small objects in all three difficulty categories. SegAttnDetec achieved a 66.6% mAP for the pedestrian class and a 68.7% mAP for the cyclists class on hard-to-detect scenes, which is a 28.5% and 38.8% increase, respectively.

Table 3.2: Overall comparison of SegAttnDetec against key existing DL-based object detectors

Model	Backbone	Category mAP (%)			Overall mAP (%)	# of Tr. params	↑ %
		Easy	Moderate	Hard			
Faster R-CNN	VGG-16 (baseline)	83.16	88.97	72.62	81.58	-	-
Faster R-CNN	ResNet-50	83.08	79.28	73.71	78.69	41.1M	-3.5
Yolov5 [15]	-	-	-	-	63.60	14.0M	-22.0
BiGA-YOLO [89]	-	-	-	-	68.30	11.9M	-16.3
SegAttnDetec (proposed)	ResNet-50 (OD) + ResNet-101 (sem-seg)	86.64	81.86	79.69	83.52	52.1M	+2.4

Note: ↑ % - improvement % compared to the baseline, # of Tr. params - number of trainable parameters.

Table 3.3: Class-wise performance comparison wrt mAP %

Object class	Easy Category		Moderate Category		Hard Category	
	FR-CNN [88]	SegAttnDetec	FR-CNN [88]	SegAttnDetec	FR-CNN [88]	SegAttnDetec
Car	84.81	97.22	86.18	89.70	78.03	88.90
Pedestrian	76.52	79.96	59.98	73.92	51.84	66.62
Cyclist	74.72	78.59	56.83	70.50	49.60	68.86

3.5.2 Qualitative Analysis

Fig. 3.8 shows the proposed model's predictions on six randomly collected samples from the KITTI validation subset. The visualization clearly shows that the model accurately predicts small objects and poorly represented object classes like pedestrians and cyclists. However, a closer examination of image **b** shows that when two pedestrians are very close (overlapping objects – marked with a yellow box) together, the model struggles to detect them. The baseline–Faster R-CNN's predictions, were omitted from the visualization for clarity and simplicity.

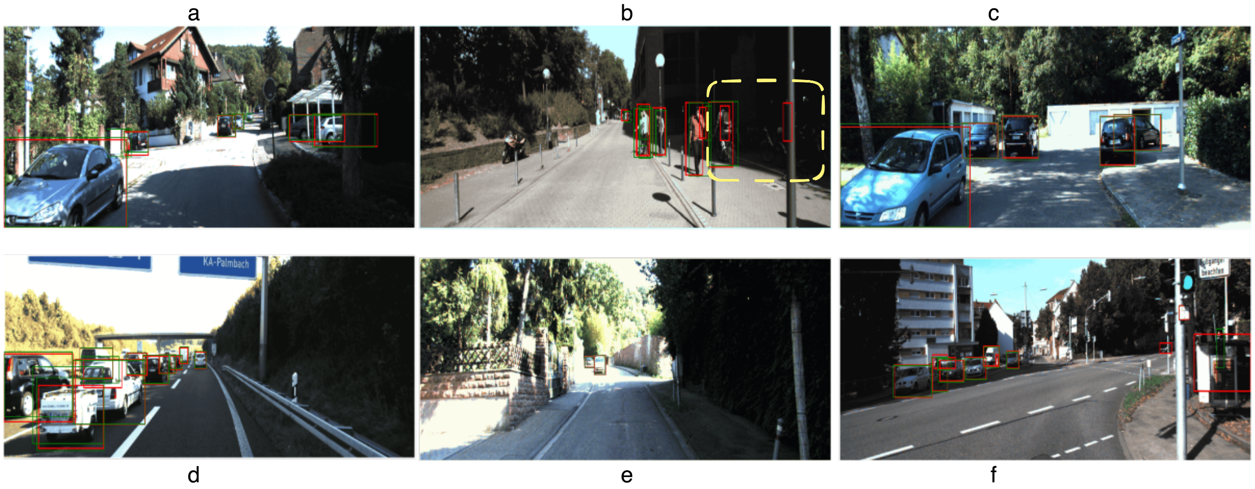


Figure 3.8: Predictions from the proposed model. Legend:— Predicted box — Ground-truth box.

3.6 Chapter Summary

This study shows that integrating semantic features from a segmentation model using attention-based feature fusion can effectively enhance the detection of small object classes, such as pedestrians and cyclists, in traffic scenes. The proposed approach leverages existing architectural components and demonstrates that even straightforward attention mechanisms, when guided by semantic context, can contribute to more accurate localization of small-scale targets. Experimental results confirm that this fusion strategy improves small object detection performance, supporting the value of incorporating segmentation-aware attention.

While the integration of semantic features via attention mechanisms (SegAttnDetec) yielded a measurable gain in mAP (83.5% vs. baseline 81.6%), the modest improvement (+1.9%) suggests underlying constraints. A contributing factor to this limitation is the segmentation model's performance, which achieved only 67.7% mIoU, indicating suboptimal feature extraction for small objects. Low mIoU implies noisy or incomplete semantic features, limiting their utility for detection. For example, fragmented segmentation masks (common in small objects) may propagate errors to the detector's ROI pooling. Furthermore, the attention gate's effectiveness depends on the quality of input features. With mediocre segmentation features, the gate may attenuate useful signals if segmentation errors correlate with detection targets. Failing to suppress noise in low-confidence regions (e.g., object boundaries). The +8.8% mAP gain from attention (vs. concatenation) suggests partial success, but higher-quality features could unlock further gains.

Future directions to improve detection performance include: enhancing the segmentation backbone by replacing DeepLabv3 with a small-object-optimized model such as Mask2Former or HRNet, which achieve >75% mIoU on CityScapes, to potentially yield higher-quality features. Additionally, implementing class-specific attention gates could better prioritize underrepresented object categories. Further evaluation on larger datasets (e.g., PASCAL VOC, COCO) and investigation of segmentation-derived confidence scores for detection refinement may validate scalability. While the current results demonstrate the approach's promise, these optimizations could unlock additional performance gains and robustness.

Chapter 4

A Specialized Feature Pyramid for Small Object Detection

4.1 Overview

This chapter explores the Dilated Strip-wise Feature Pyramid (DSSFP), an innovative neural architecture specifically engineered to overcome the persistent challenges of multi-scale feature representation and small object detection in high-resolution aerial imagery. In complex environments characterized by extreme scale variations, crowded object distributions, and highly anisotropic structures, conventional feature-pyramid networks often struggle to maintain computational efficiency and detection accuracy. The DSSFP addresses these limitations through the integration of attention mechanisms, directional feature extraction, and adaptive multi-scale fusion.

As discussed in Chapter 2, effective strategies for small object detection include (i) leveraging multi-scale feature hierarchies, (ii) employing attention mechanisms, and (iii) utilizing a context-aware learning framework. These techniques help balance the inherent trade-offs between spatial resolution and semantic richness. Additionally, specialized architectural modifications, such as high-resolution backbones and refined feature pyramid designs, are particularly beneficial for detecting objects that occupy only a few pixels. The following sections expand on these foundational principles to present novel contributions, specifically tailored to the challenges found in aerial drone imagery.

The chapter is organized to progressively develop the reader's understanding, beginning with fundamental concepts of feature pyramid networks, followed by detailed explanations of each architectural innovation, comprehensive ablation studies that validate design choices, and concluding with extensive benchmark comparisons against current state-of-the-art methods across a benchmark aerial imagery dataset.

4.2 Background Concepts

4.2.1 Fast Normalized Fusion

Fast normalized fusion is an efficient feature integration technique that dynamically combines multi-scale feature maps through learnable weights. Unlike simple summation or concatenation, this method assigns channel-wise importance weights to each input feature map before normalization, allowing the network to automatically emphasize the most semantically meaningful features during pyramid fusion. The weighted average operation maintains numerical stability through a small epsilon term while preserving gradient flow during backpropagation. Originally introduced in BiFPN [72] architecture, this approach significantly improves multi-scale feature representation by adaptively balancing contributions from different resolution levels, high-level semantic features from deep layers, and fine-grained spatial details from shallow layers. The computational simplicity of fast normalized fusion shown in eq. 4.1 makes it particularly suitable for real-time applications, as it avoids the heavy memory overhead of attention mechanisms while achieving comparable or superior performance in object detection tasks.

$$U = \sum_{i=1}^{N} w_i \Bigg/ \left(\epsilon + \sum_{j=1}^{N} w_j \right) I_i, \tag{4.1}$$

where $w_i \geq 0$ and $\epsilon = 10^{-4}$ (numerical stability constant).

4.2.2 Convolutional Block Attention Module

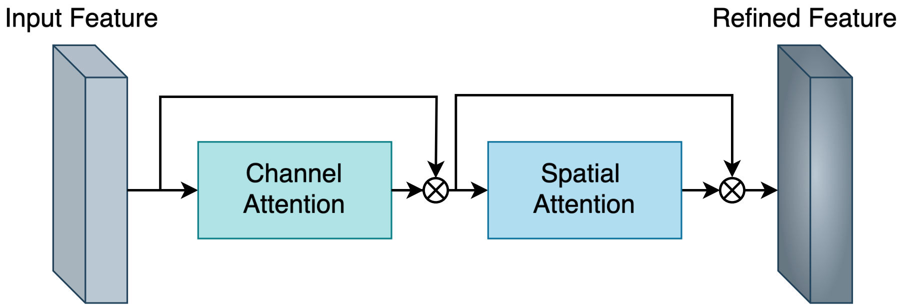


Figure 4.1: Overview of CBAM with channel and spatial attention.

In CNNs, attention can be applied across spatial and channel dimensions allowing models to learn dynamic weighting schemes based on contextual relevance that enhance feature representations. One widely adopted method is the Convolutional Block Attention Module (CBAM) [90], illustrated in Fig. 4.1, which operates with two sequential components: channel attention and spatial attention. Channel attention dynamically recalibrates channel-wise responses by capturing inter-channel dependencies. It does so by applying both global average pooling (GAP) and global max pooling (GMP) across the spatial domain to create information-rich descriptors, which are then passed through a shared MLP. The MLP output is regulated by a sigmoid function rescaling the feature maps, effectively prioritizing informative ones. Spatial attention, in contrast, focuses on where important features are located by using the pooled channel-wise descriptors to compute a 2D spatial attention map. The final concatenated spatial feature maps are passed through a convolutional layer followed by a sigmoid activation to highlight key spatial regions.

4.2.3 Dilated Strip Convolution

Beyond attention, another crucial mechanism for enlarging receptive fields without losing resolution is dilated convolution (aka atrous convolution) [85]. Standard convolution operations are limited in capturing large-scale context without downsampling. Dilated convolution resolves this by inserting holes (or dilations) between kernel elements, allowing the network to aggregate broader contextual information while maintaining the original spatial resolution. However, standard (isotropic) dilated convolutions expand uniformly in all directions, which may introduce gridding artifacts and struggle with scale-variant or elongated object structures. To address this, strip convolutions (or dilated strip convolutions) shown in Fig. 4.2 have been proposed as an alternative. These kernels are axis-aligned (either horizontal or vertical), enabling the model to focus on long-range dependencies along dominant spatial directions. This directional bias is particularly beneficial in aerial and remote sensing imagery, such as UAV-based datasets, where objects like roads, rivers, or vehicles exhibit strong orientation regularities. Compared to conventional dilated convolutions, strip convolutions offer a more selective context aggregation mechanism, reducing noise introduced by globally uniform dilation and improving feature specificity. For example, Hou et al. [91] show that integrating strip kernels aligned with object orientation not only enhances detection performance but also reduces parameter count by up to 2.3×, offering

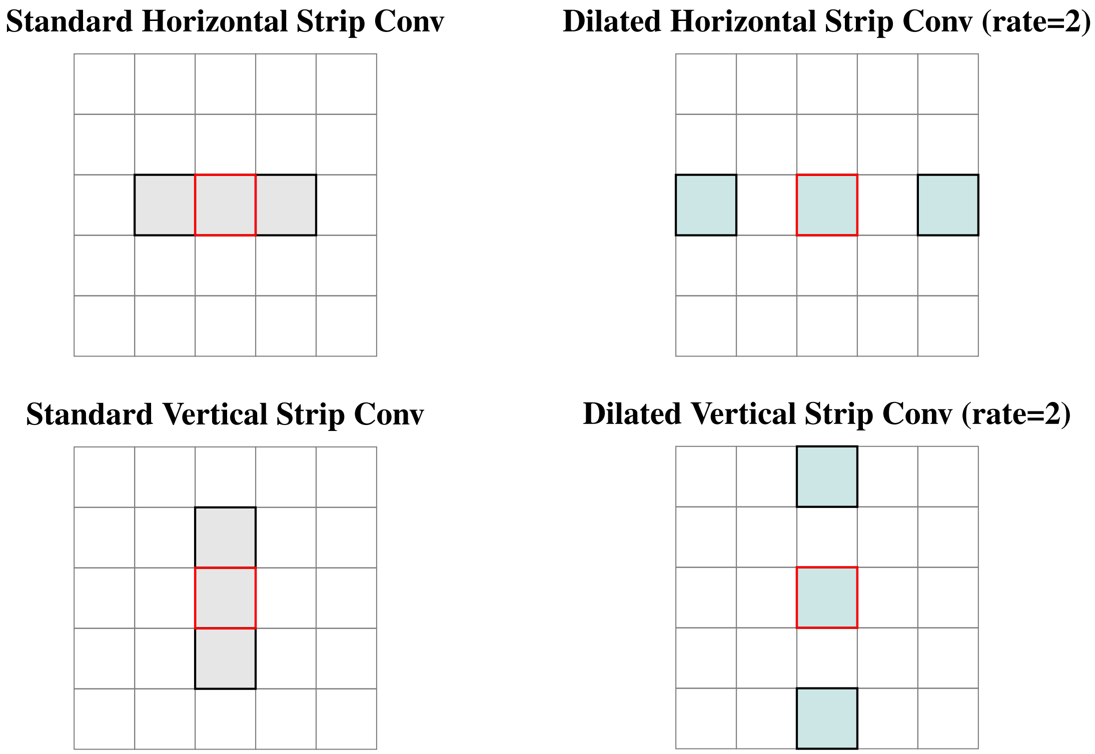

Standard Horizontal Strip Conv
Dilated Horizontal Strip Conv (rate=2)
Standard Vertical Strip Conv
Dilated Vertical Strip Conv (rate=2)

Figure 4.2: Illustration of strip convolutions on a 5×5 feature map. The top row shows horizontal strip convolutions: the standard version (left) uses contiguous cells, while the dilated version (right) applies a gap (rate=2). The bottom row shows vertical strip convolutions with analogous sampling patterns. The red-bordered cell indicates the center pixel.

a compelling trade-off between computational efficiency and representational power. Recent research [92] also explores combining strip convolutions with attention mechanisms, enabling networks to learn both "where" to focus (spatial attention) and "how far" to aggregate (directional dilation), which together improve accuracy in tasks with challenging geometric variability. This hybrid approach presents promising directions for detecting scale-variant, elongated, or clustered objects in domains like medical imaging, traffic surveillance, and UAV-based scene understanding.

4.3 Proposed Architecture

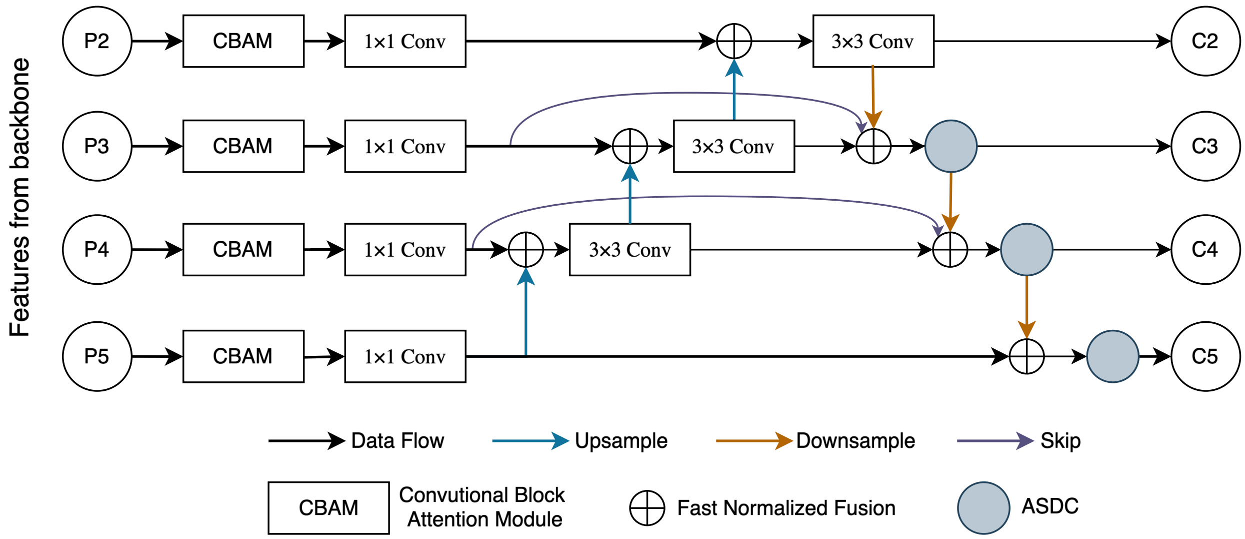


Figure 4.3: Proposed DSSFP architecture with multi-scale features $\{P_2, P_3, P_4, P_5\}$. P_i corresponds to the ith Conv block output (Stage i in HRNet nomenclature). P_3 through P_5 are progressively down-sampled higher-level features.

4.3.1 Dilated Strip-wise Spatial Feature Pyramid

The dilated strip-wise approach (cf. Fig. 4.3) allows DSSFP to simultaneously capture both fine details (critical for tiny objects) and broader contextual information (essential for accurate classification and localization). The pyramid structure maintains these properties across multiple scales, ensuring consistent performance regardless of object size within the image. By integrating DSSFP, this architecture achieves superior performance in small object detection in complex aerial scenes while maintaining computational efficiency. DSSFP delivers the precise spatial discrimination. Besides, the architecture implements a sophisticated hierarchical fusion strategy that operates across multiple resolution levels (P2-P5). Leveraging fast normalized fusion (cf. Section 4.2.1) with learnable weights, DSSFP dynamically balances contributions from different scales to automatically emphasize the most semantically meaningful features. The pyramid is further extended through the innovative use of deformable convolutions, which adaptively downsample low-resolution features (from 1/32 to 1/64 scales) while preserving critical spatial information through learned sampling offsets.

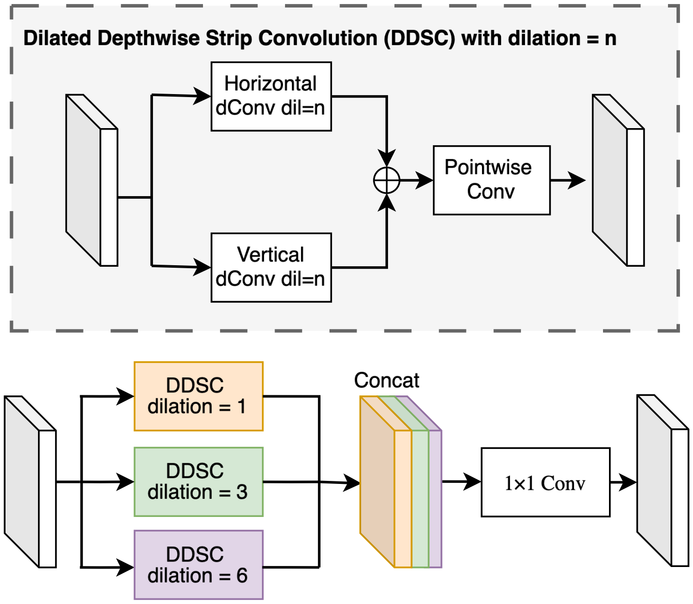


Figure 4.4: Illustration of the ASDC that utilizes DDSC blocks with dilation rates 1, 3, and 6 for better feature representation.

To construct the DSSFP, a novel Atrous Split Depthwise Convolution (ASDC) block (cf. Fig. 4.4) is incorporated, to create a computationally efficient yet highly expressive building block. ASDC uniquely combines two key innovations: (a) Parallel depthwise strip convolutions (horizontal 1×3 and vertical 3×1 kernels) that capture directional patterns, and (b) Multiple dilation rates (1, 3, 6) that efficiently expand receptive fields without increasing parameter count.

This dual approach enables the network to model long-range spatial relationships while maintaining sensitivity to the anisotropic structures.

4.4 Model Optimization Strategies

4.4.1 Quest for the Best Backbone and FPN

Table 4.1: The summary of the sanity test during the quest for the best backbone and FPN

Model	Params (M) ↓	Test Loss ↓	Test mAP (%) ↑
A. Baseline Model			
RetinaNet-R101 w/FPN [55]	57.0	0.980	9.62
B. Backbone Ablation Study (with RetinaNet)			
GhostNet [93]	9.9	1.020	3.55
EfficientNet-Tiny [94]	21.1	0.990	9.54
HRNetV2-W18 [51]	20.1	0.930	9.62
C. Feature Pyramid Ablation (HRNet-W18 Backbone)			
FPN [95]	20.1	0.933	9.62
BiFPN (1 block) [72]	18.8	0.931	9.77
BiFPN (3 blocks)	19.9	0.929	9.79
SSPN [96]	29.3	0.956	9.67
SSBiFPN	25.6	0.956	8.92

Note: All metrics reported on VisDrone-2019 test set. ↓ indicates lower is better, ↑ indicates higher is better.
Implementation Details: Input size = 640×640, batch size = 4, trained on 20% of the train set.

To establish a comprehensive baseline for the architectural improvements, a systematic evaluation of feature pyramid networks (FPNs) and backbone architectures was conducted to establish a comprehensive baseline for comparison against proposed architectural improvements. While prior work [72] has demonstrated BiFPN's superiority over FPN, this work sought to verify these findings independently within the experimental framework. RetinaNet was employed as a representative one-stage detector. For these comparative studies, a mini version of the VisDrone-2019 [1] that captures the unique challenges of aerial object detection was curated. Experiments encompassed multiple FPN variants, including SSPNet [96] and BiFPN, with particular attention to their interaction with different backbone architectures. HRNet was identified as the most effective backbone through extensive testing.

Following the selection of optimal backbone and FPN components, the proposed ADSC modules were integrated into the upsampling blocks. The quantitative results of these architectural comparisons are presented in Table 4.1, which demonstrates the performance advantages of the selected configuration. This rigorous evaluation validated the choices of backbone and feature pyramid module, providing valuable insights into the relative contributions of both components to the overall system performance.

4.4.2 Optimizing the Architecture

Our initial architecture, shown in Fig. 4.5 which combined DSSFP and large field-of-view (cf.Fig. 4.6) delivered strong detection performance but required considerable computational resources, as reflected in its higher GFLOPS measurement. To improve computational efficiency, we explored an optimized design by strategically combining the DSSFP module with the large field-of-view (FOV) approach. The key modification involved repositioning the ASDC block to operate after the final feature fusion stage, as illustrated in Fig. 4.3.

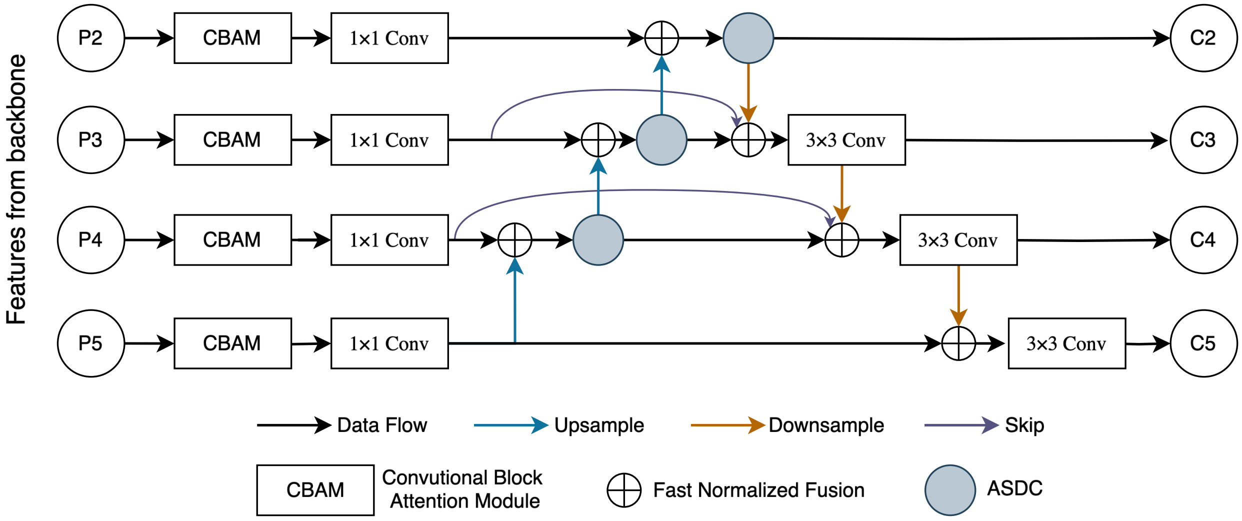


Figure 4.5: Initial DSSFP architecture with multi-scale features $\{P_2, P_3, P_4, P_5\}$. P_i corresponds to the ith Conv block output (Stage i in HRNet nomenclature). P_3 through P_5 are progressively down-sampled higher-level features.

This architectural adjustment yielded significant efficiency gains, reducing the computational cost from 150 GFLOPS to 115 GFLOPS - a 23% decrease in floating-point operations. While this modification introduced a slight increase in parameters (from 31.6 million to 33.0 million), the trade-off proved favourable. The optimized model achieved a mean average precision (mAP) of 27.4%, representing only a marginal 0.2% reduction compared to the original implementation. Given the substantial 35 GFLOPS reduction in computational overhead, this minor performance trade-off was well justified, particularly for deployment scenarios where computational efficiency is paramount. The revised architecture demonstrates that careful structural modifications can yield significant efficiency improvements while maintaining competitive detection accuracy.

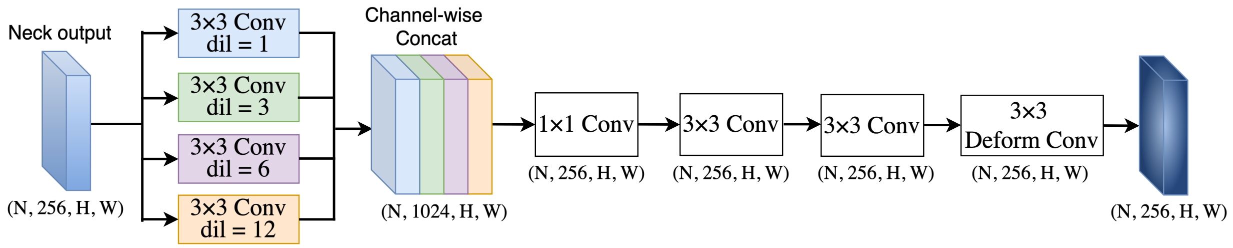


Figure 4.6: An illustration of the Large FOV head, where N, H, and W denote the batch size and height and width of the feature map, respectively. Its output is passed to the regression and classification heads.

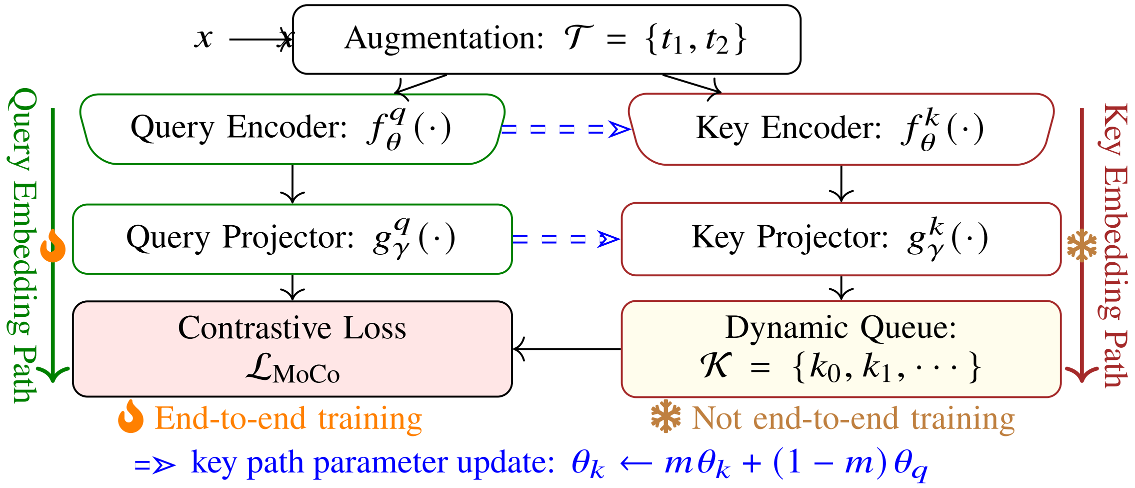
Large Field-of-View Mechanism

The architecture introduces an expanded field-of-view (FOV) mechanism, as shown in Fig. 4.6, on the detection head of the model, enabling each layer to capture broader contextual information while maintaining computational efficiency. This large FOV design is particularly crucial for aerial imagery, where objects may have important long-range spatial relationships that need to be considered for accurate detection. The expanded receptive fields allow the network to better understand scene composition and object relationships at multiple scales.

4.4.3 Enhancing the Feature Representation of the Backbone

Momentum Contrast (MoCo) [97], a contrastive learning-based self-supervised learning (SSL) framework illustrated in Fig. 4.7 was employed to pre-train the chosen backbone on unlabeled aerial imagery. This approach learns transferable feature representations to the downstream task, in this case, object detection, by optimizing a contrastive objective that distinguishes between:

- **Positive pairs**: Differently augmented views of the same image (query q and key k)
- **Negative samples**: Features of unrelated images stored in a dynamic memory queue



x
Augmentation: $\mathcal{T} = \{t_1, t_2\}$
Query Encoder: $f_\theta^q(\cdot)$
Key Encoder: $f_\theta^k(\cdot)$
Query Projector: $g_\gamma^q(\cdot)$
Key Projector: $g_\gamma^k(\cdot)$
Contrastive Loss
$\mathcal{L}_{\text{MoCo}}$
Dynamic Queue:
$\mathcal{K} = \{k_0, k_1, \cdots\}$
Query Embedding Path
Key Embedding Path
End-to-end training
Not end-to-end training
key path parameter update: $\theta_k \leftarrow m\theta_k + (1-m)\theta_q$

Figure 4.7: An illustration of the MoCo pipeline. It is asymmetrically trained: Only the query embedding path is trained end-to-end, while the key embedding path is updated via (4.2).

MoCo maintains separate networks for query and key embeddings, as illustrated in Fig. 4.7. Its training is asymmetric, i.e., only the query path, $f_\theta^q(\cdot) + g_\gamma^q(\cdot)$, is updated via backpropagation using a modified `InfoNCE` loss function, while the key embedding path is updated via a momentum update with smoothing exponentially moving average (EMA). For instance, the key encoder parameter, θ^k, is updated as:

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q, \tag{4.2}$$

where $m \in [0, 1)$ is a momentum coefficient, and θ_q is the query encoder parameter. The same update rule applies to the key projection network, $g_\gamma^k(\cdot)$. MoCo optimizes contrastive loss using the dynamic queue:

$$\mathcal{L}_{\text{MoCo}} = -\log \frac{\exp(\text{sim}(\mathbf{q}, \mathbf{k}^{+})/\tau)}{\sum_{i=1}^{K} \exp(\text{sim}(\mathbf{q}, \mathbf{k}_i)/\tau)}, \tag{4.3}$$

where $\mathbf{q}$ is the query representation, $\mathbf{k}^{+}$ is the positive key (an augmented view), $\mathbf{k}_i$ are negative keys (from the queue), τ is a temperature parameter, and $\text{sim}(\cdot, \cdot)$ denotes cosine similarity.

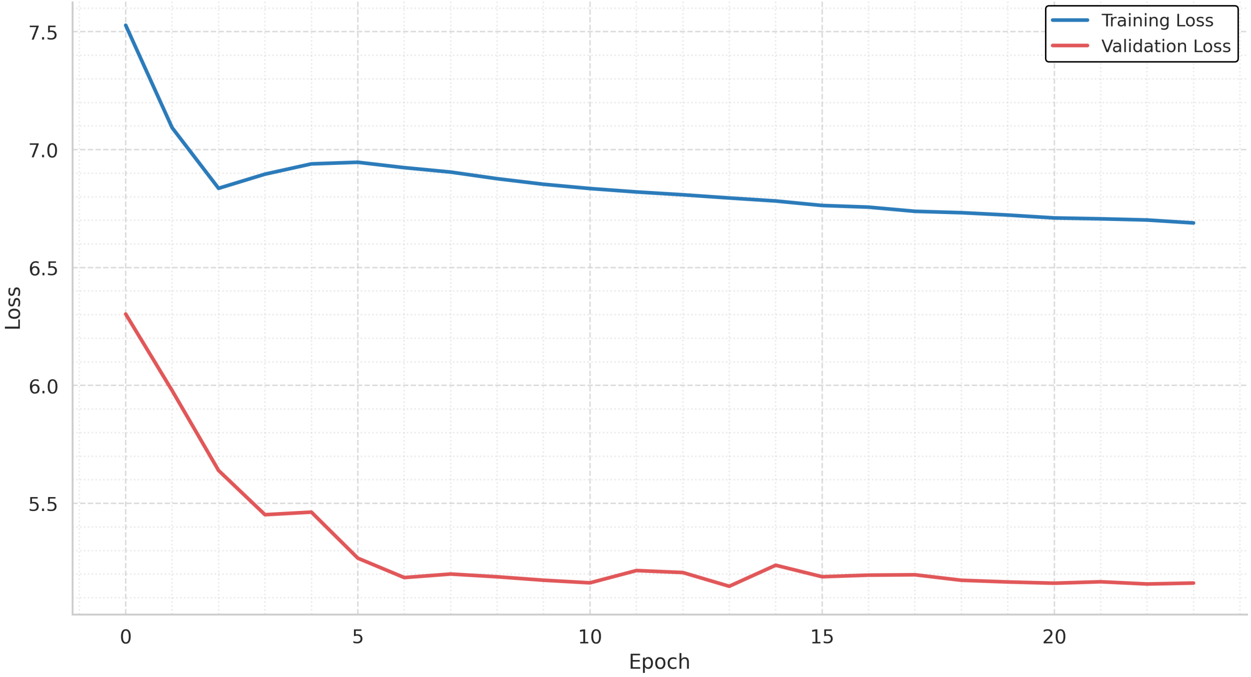


Figure 4.8: Training plot during pre-training of the backbone network via MoCo.

The contrastive learning framework employs dual HRNetV2-W18 encoders with a 128-dimensional projection space. The momentum encoder (updated via EMA with m=0.996) maintains stable feature targets, while separate memory banks handle negative samples - a 4,096-entry queue for training and a 1,024-entry queue for validation. The `InfoNCE` loss operates on these memory banks, with the larger training queue ensuring diverse negative samples and the compact validation queue optimizing memory usage during evaluation. This implementation achieves stable convergence, as shown in the training curves in Fig. 4.8, with the EMA updates preventing representation collapse while allowing progressive feature refinement.

The Momentum Contrast framework critically depends on carefully designed image transformations to generate meaningful positive pairs. The augmentation pipeline uses probability-based generation where p is the probability and combines the following transformations:

- **Core Contrastive Augmentations**: Random resized crop (scale 0.2–1.0) with aspect ratio preservation, Horizontal flip ($p = 0.5$) , Vertical flip ($p = 0.2$)
- **MoCo v2 Standard Augmentations**: Color jitter ($p = 0.8$, brightness=0.4, contrast=0.4, saturation=0.4, hue=0.1), Gaussian blur ($p = 0.5$, $\sigma \in [0.1, 2.0]$, kernel size 5×5)
- **Domain-Specific Adaptations**: Random grayscale conversion ($p = 0.2$), ImageNet normalization (mean=$[0.485, 0.456, 0.406]$, std=$[0.229, 0.224, 0.225]$)

These transformations serve two essential purposes:

1. Create *valid positive pairs* through semantic-preserving geometric variations (crops/flips)
2. Prevent *trivial solutions* by breaking low-level feature correlations (color/blur)

The GFL+HRNet model with self-supervised learning (SSL) pretraining achieved 24.2% mAP, demonstrating a 0.6 percentage point improvement over the same model without SSL pretraining (23.6% mAP).

4.4.4 Augmentation for Better Model Generalization

To enhance the robustness and generalization of the proposed model, a series of data augmentation strategies were applied during training. These techniques are particularly important for small object detection, where limited diversity and resolution can negatively impact model performance. Mosaic augmentation was used to combine four different images into a single composite image, enabling the model to learn from varied object scales and contexts within a single training instance. MixUp, which blends two images and their labels to produce a soft-labelled sample, served as a regularization technique to reduce overfitting. Additionally, random horizontal flipping was employed to introduce geometric variability, helping the model become invariant to object orientation.

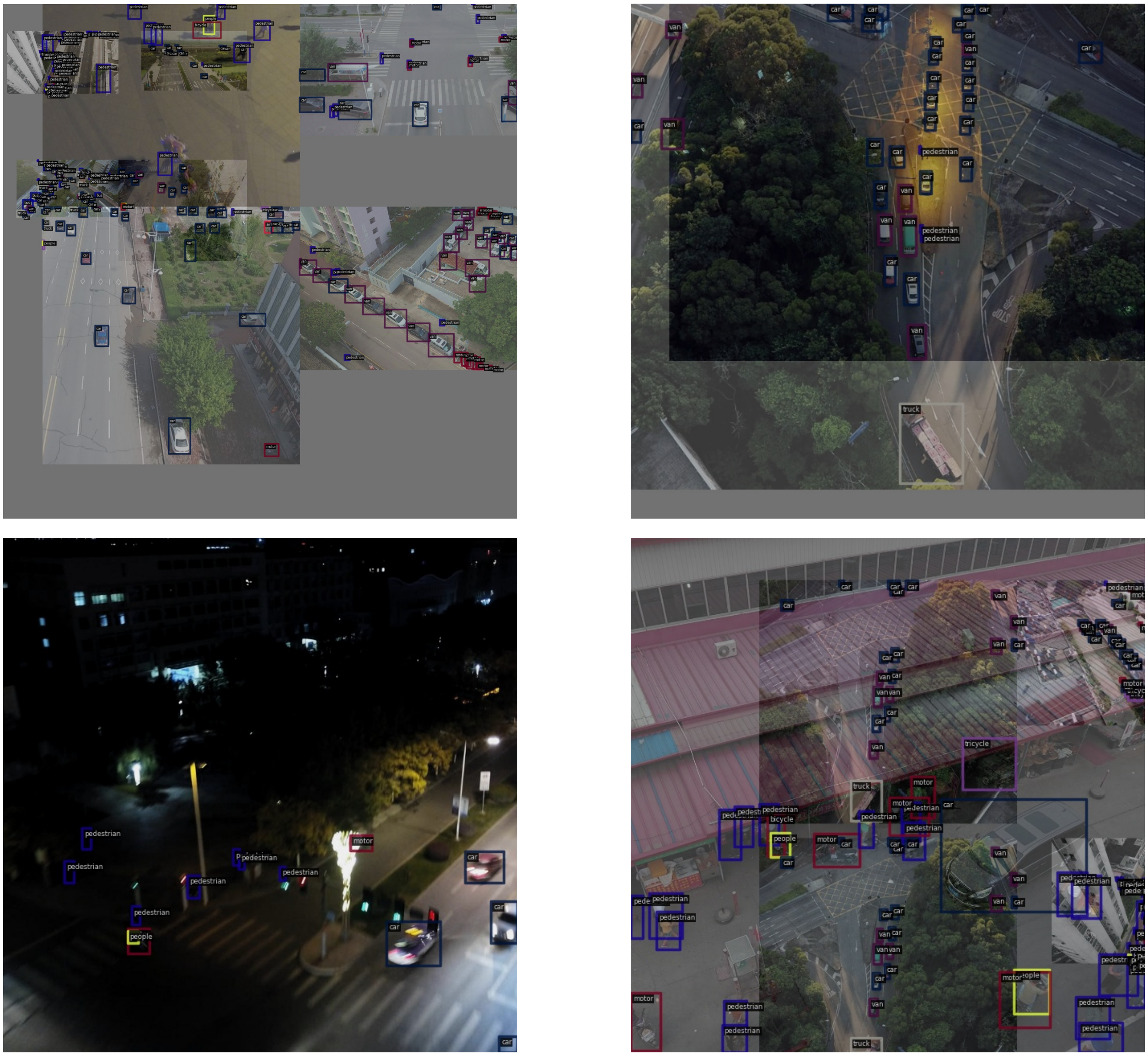


Figure 4.9: Augmentation examples using colour jitter, mixup, mosaic, and random crop.

Color jittering was also applied to simulate changes in illumination by randomly adjusting brightness, contrast, saturation, and hue, improving the model's resilience to diverse lighting conditions.

As illustrated in Fig. 4.9, these augmentation strategies significantly increased the visual and contextual diversity of the training data, contributing to a more stable and effective learning process, especially for detecting small and challenging object classes.

4.4.5 Hyperparameter Optimization

The hyperparameters: head depth, atrous convolution rates, learning rate, loss function and assigned are found to significantly impact model performance, as summarized below.

- **Model Architecture:** Four detection frameworks (Faster R-CNN, RetinaNet, FCOS, and GFL) were evaluated for small object detection, with GFL demonstrating superior performance. While Faster R-CNN struggled with fixed anchor scales and RetinaNet showed limitations in label assignment, GFL's integrated approach, combining dynamic label assignment, joint classification, localization optimization, and continuous box representation, proved most effective for handling aerial imagery's dense distributions and extreme scale variations. FCOS performed competitively but exhibited confidence-localization discrepancies that GFL's unified prediction head successfully addressed. This systematic comparison motivated the selection of GFL as the base detection framework.

- **Dilation Configuration Analysis:** Systematically evaluated dilation configurations across architectural components, *ADSC in DSSFP*; found the triple dilation (1,3,6) most effective, as the strip convolution's elongated receptive fields naturally capture broader contexts

  This selective approach, using expanded dilations in standard convolutions but compact ones in ADSC, optimizes the trade-off between receptive field coverage and computational efficiency across different operator types.

- **Learning Rate:** Through empirical validation, 0.01 was established as the optimal learning rate for stable training. Higher values consistently induced exploding gradients, while lower rates significantly slowed convergence without improving final performance. This configuration maintained effective gradient flow throughout all training stages, balancing update precision with training efficiency.

- **Assinger and Loss Function:** A comparative study between task-aligned assinger (TAA) paried with VFL detection framework and the ATSS assignment strategy with GFL loss was performed to establish the best assignment algorithm. The TAA+VFL architecture demonstrated clear advantages in detection accuracy, particularly for challenging scenarios. The key strength of TAA+VFL lies in how the assigned operates, it dynamically selects training samples based on a combination of classification confidence and localization quality (IoU). This approach ensures that the model prioritizes well-aligned predictions during training, effectively addressing the common misalignment between classification and bounding box regression tasks. The integration of VFL loss further enhances performance by adaptively weighting positive samples according to their IoU scores, while the Distribution Focal Loss (DFL) refines bounding box predictions by modeling their distributions rather than relying on single-point estimates.

Table 4.2: Comparison of ATSS+GFL and TAA+VFL

Model	**Test mAP** ↑	**Test mAP@50** (%) ↑
GFL-HRNet+LargeFOV	25.4	40.5
GFL-HRNet+DSSFP+LargeFOV	26.3	41.6
TAL-HRNet+DSSFP+LargeFOV	27.6	46.8

Note: All metrics reported on VisDrone-2019 test set. ↓ indicates lower is better, ↑ indicates higher is better.
Implementation Details: Input size = 640×640, batch size = 6.

By contrast, the ATSS assigner employs a more conventional iou-based statistical approach for sample selection, pairing it with the GFL to jointly optimize classification and bounding box quality estimation. While this combination offers computational efficiency and works well for general object detection, it lacks the explicit task-alignment mechanism that makes TAA particularly effective for complex cases. Through experimentation it was revealed that TAA+VFL consistently outperformed the ATSS+GFL approach in detection accuracy, especially for small objects and crowded scenes. The superior performance can be attributed to TAA+VFL's ability to maintain better harmony between classification and localization objectives during training, leading to fewer false positives and more precise detections. These findings suggest that for applications where detection precision is paramount, the task-aligned paradigm provides a more robust solution than traditional assignment strategies like ATSS paired with GFL. Table 4.2 summarizes the findings.

4.5 Model Training

The following sections detail the training protocol and performance analysis.

4.5.1 VisDrone Dataset

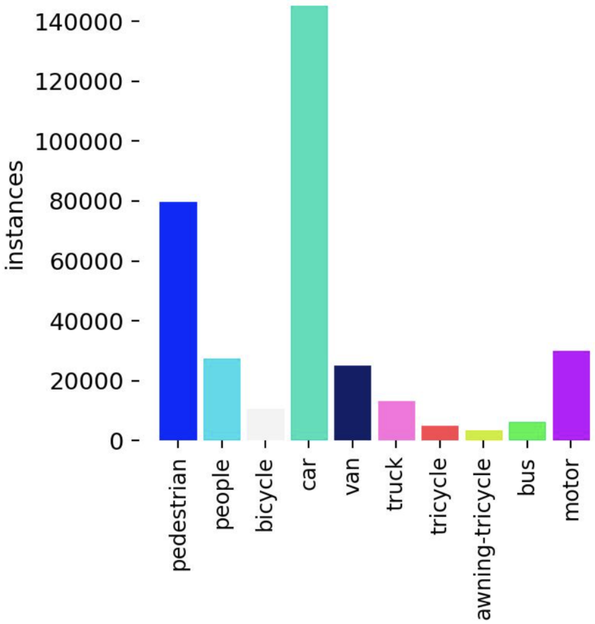


Figure 4.10: Class distribution of objects in VisDrone-2019 [1] dataset.

The VisDrone-2019 benchmark dataset is a large-scale aerial imagery collection containing 10 object categories (pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, and motor) captured across diverse urban and rural environments in China. The dataset comprises 261,908 frames extracted from 288 video clips and 10,209 static images acquired using various drone platforms under different weather and lighting conditions. These images are annotated with over 2.6 million bounding boxes, including detailed attributes for occlusion levels and visibility states. Fig. 4.10 shows the distribution of the labels in the dataset.

The dataset is mutually exclusive, divided into training (6,471 images), validation (548 images), and test sets (1,610 images), providing comprehensive coverage of challenging aerial scenarios ranging from sparse to highly crowded scenes. Collected across 14 cities with multiple drone models, VisDrone-2019 represents one of the most extensive and varied benchmarks for drone-based object detection, particularly for studying small object detection in complex environments.

4.5.2 AI-TOD Dataset

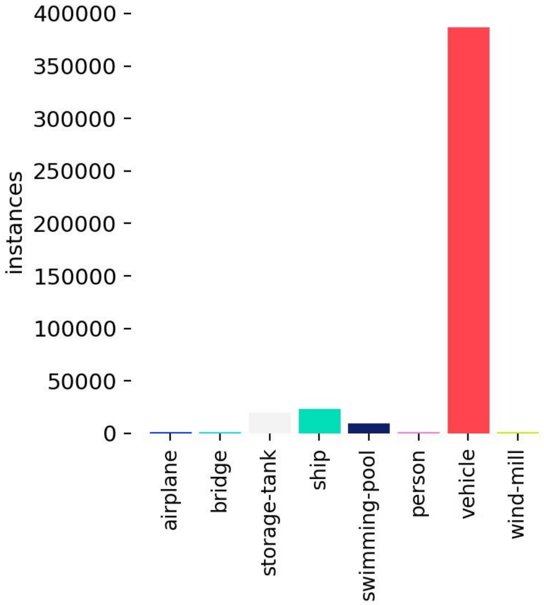


Figure 4.11: Class distribution of objects in AI-TOD [2] dataset.

The AI-TOD (Aerial Image Tiny Object Detection) dataset is a specialized benchmark designed for advancing tiny object detection in aerial imagery. It focuses on extremely small objects (often less than 16×16 pixels) captured in high-resolution satellite and drone images. The dataset includes 28,036 images annotated with 700,621 bounding boxes across 8 object categories, which are airplane, bridge, person, ship, swimming-pool, storage-tank, vehicle and wind-mill. These objects exhibit significant scale variations and are densely distributed in complex backgrounds, posing unique challenges for detection algorithms. The dataset is widely used to evaluate state-of-the-art detectors for small objects, particularly in remote sensing and surveillance applications.

While VisDrone focuses on drone-captured urban scenes with objects like pedestrians and cars, AI-TOD targets satellite/UAV imagery with sub-16px objects, emphasizing extreme scale variations. Both datasets address aerial detection but cater to distinct research needs (general urban objects vs. tiny structured objects).

4.5.3 Training of the Proposed Model

For systematic and comprehensive model development and comparison, this study trains and tests the baseline model and the proposed model. The proposed model aims to minimize three losses: Varifocal (VFL) as in 4.4, Bounding box regression (DFL) as defined in (4.5), and Bounding box overlap loss (GIoU) as expressed in (4.6).

$$\mathcal{L}_{\text{VFL}} = \begin{cases} -q(q\log(p) + (1-q)\log(1-p)) & \text{if } q > 0 \\ -\alpha p^{\gamma}\log(1-p) & \text{otherwise} \end{cases} \tag{4.4}$$

where p is the predicted classification score (sigmoid output, p$\in [0,1]$). y is a binary label ($y = 1$ for positive, $y = 0$ for negatives samples) and β is a modulating factor (typically $\beta = 2$) to balance easy/hard examples.

$$\text{DFL}(S_i, S_{i+1}) = -\left((y_{i+1} - y)\log(S_i) + (y - y_i)\log(S_{i+1})\right), \tag{4.5}$$

where y is the bounding box coordinate (the target value), y_i and y_{i+1} the nearest two values satisfying $y_i \leq y \leq y_{i+1}$, and S_i, S_{i+1} are the predicted probabilities for y_i and y_{i+1}.

$$\text{GIoU} = \text{IoU} - \frac{C \setminus (A \cup B)}{C}, \tag{4.6}$$

where IoU is standard Intersection over Union, A: Area of predicted bounding box, B: Area of ground truth bounding box, C: Area of the smallest enclosing convex shape containing both A and B, $\setminus$: Set difference operator. The final loss (TAL) is a combination of the above losses and can be summarized as:

$$\text{Loss}(TAL) = 0.5 \times \text{VFL} + 1.5 \times \text{DFL} + 7.5 \times \text{Loss}_{\text{GIoU}} \tag{4.7}$$

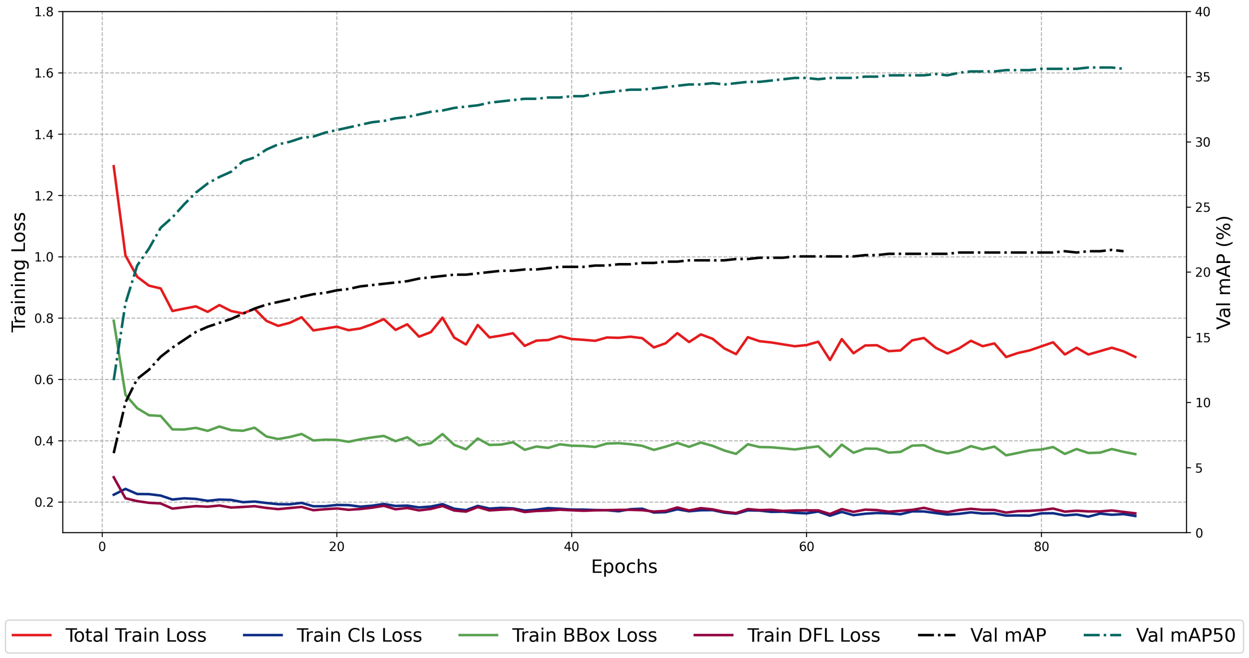
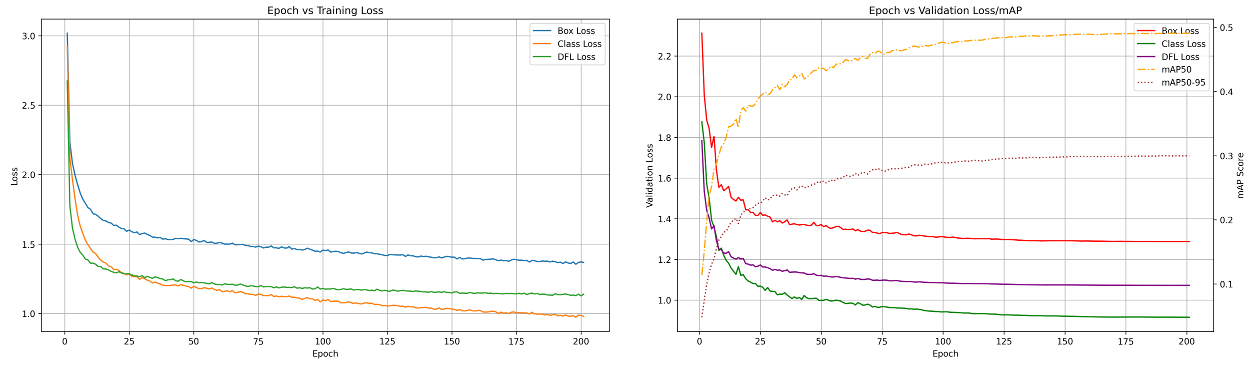


Figure 4.12: Training progress of the baseline GFL model with respect to mAP and loss versus training epochs on the VisDrone benchmark dataset.

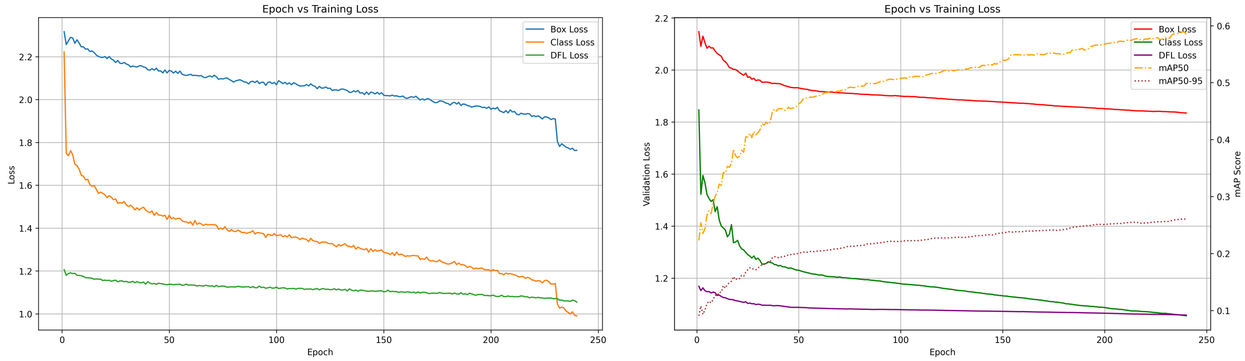


(a) Training loss versus epochs.

(b) Validation loss and mAP versus epochs.

Figure 4.13: Training progress of the proposed model on the VisDrone benchmark dataset, showing the convergence behaviour of the proposed model

Figures 4.12, 4.13, and 4.14 present the training plots for the baseline and proposed model on VisDrone and AI-TOD, respectively, based on the minimization of the combined loss. The VisDrone-2019 training loss shows a steady decline across epochs, with box loss converging faster than class loss, indicating stable learning of object localization. In contrast, AI-TOD exhibits more volatile loss fluctuations, particularly in DFL, reflecting the inherent challenges of optimizing for



(a) Training loss versus epochs.

(b) Validation loss and mAP versus epochs.

Figure 4.14: Training progress of the proposed model on the AI-TOD benchmark dataset, showing the convergence behaviour of the proposed model

extremely small objects. Both datasets demonstrate initial high losses that gradually stabilize, though AI-TOD's convergence is slower due to its fine-grained detection requirements. All models were trained with data augmentation and evaluated using the official mAP metric. For fair comparison, all baseline and ablation variants were trained under identical settings.

4.6 Experimental Analysis

4.6.1 Environment

All models were developed using Python 3.11.5 with PyTorch 2.1, TorchVision, and MMEngine and Ultralytics frameworks. The experiments were executed on Compute Canada's Narval cluster, utilizing four NVIDIA A100-SXM4-40GB GPUs (40GB HBM2 VRAM each) with 3rd Gen AMD EPYC processors and 128GB system memory.

4.6.2 Quantitative Analysis

VisDrone

Referring to Table 4.4, the proposed approach demonstrates substantial performance gains in object categories characterized by small physical dimensions and densely packed spatial arrangements,

Table 4.3: Performance Comparison on VisDrone-2019

Model	Input Size	Params (M) ↓	GFLOPS ↓	AP (%) ↑	AP50 (%) ↑
YOLO-Based Detectors					
YOLOv8-M [98]	640 × 640	25.9	78.9	24.6	40.7
YOLOv8-L [98]	640 × 640	43.7	165	26.1	42.7
YOLOv9-S [99]	640 × 640	17.2	26.7	22.9	38.3
YOLOv9-M [99]	640 × 640	20.1	76.8	25.2	42.0
UAV Object Detectors					
QueryDet [100]	2400 × 2400	33.9	212	28.3	48.1
Cascade RCNN+ [1]	736 × 736	-	-	17.67	34.9
Transformer Based Detectors					
DETR [101]	1333 × 750	60.0	187	24.1	40.1
Deformable DETR [63]	1333 × 800	40.0	173	27.1	42.2
Sparse DETR[102]	1333 × 800	40.9	121	27.3	42.5
RT-DETR-R50[103](SOTA)	640 × 640	42.0	136	28.4	47.0
Our Methods					
GFL-ResNet101(Baseline)	640 × 640	51.3	112	22.4	36.8
TAL-HRNet-DSSFP	640 × 640	31.6	115	27.6	46.4

Note: ↓ (Lower is better), ↑ (Higher is better).

Table 4.4: AP Scores on VisDrone-DET2019 Test Set by Category

Method	**Object Categories**									
	Ped.	Pers.	Bike	Car	Van	Truck	Tric.	Awn.	Bus	Motor
CornerNet	20.4	6.5	4.5	40.9	20.2	20.5	14.0	9.2	24.3	12.1
RetinaNet	9.9	2.9	1.3	28.9	17.8	11.3	10.9	8.0	22.2	7.0
Our Methods										
GFL-ResNet101(Baseline)	12.3	4.2	6.0	44.1	31.8	36.2	14.3	12.6	50.1	12.1
TAL-HRNet-DSSFP	20.4	12.0	9.7	57.1	37.0	36.4	17.7	16.1	48.0	21.7

such as persons and bicycles. The observed improvements underscore the effectiveness of the proposed pyramid architecture in retaining fine-grained local details while simultaneously aggregating contextual cues from broader receptive fields. This balance is crucial for detecting small and densely located objects, which often rely on subtle visual cues that may be lost in traditional downsampling operations.

Furthermore, as illustrated in Table 4.3, the model not only achieves accuracy that is competitive with, and in some cases superior to, state-of-the-art and contemporary detection frameworks, but it also does so with favorable computational efficiency. This trade-off between accuracy and resource consumption is particularly important in real-world UAV-based applications, where onboard processing power and memory are constrained. Overall, these results validate the effectiveness of the feature pyramid enhancements and highlight the model's practical utility in scenarios involving small object detection under resource-limited conditions.

AI-TOD

Table 4.5: Performance Comparison on AI-TOD Dataset

Model	**AP50 (%)** ↑	**AP (%)** ↑
Cascade R-CNN [104]	30.8	13.8
FCOS [60]	24.1	9.8
CenterNet [61]	39.2	13.4
DetectoRS [105]	32.9	14.8
FSANet [106]	41.4	15.2
SP-YOLOv8s [107]	48.4	22.7
MSFE-YOLO [108]	50.1	22.8
SOD-YOLOv8n [109]	50.7	23.4
FM-RTDETR [110]	56.3	26.9
TAL-HRNet-DSSFP	59.9	27.8

Table 4.5 reveals that DSSFP not only surpasses contemporary frameworks, achieving 59.9% AP50 and 27.8% AP, a +3.6% AP50 margin over the transformer-based FM-RTDETR, but also maintains computational efficiency critical for UAV deployments. While FM-RTDETR [110] shows competitive accuracy, DSSFP's hybrid design balances precision and resource constraints, making it viable for edge devices. For instance, it outperforms anchor-free methods like FCOS by +35.8% AP50 without incurring the memory overhead of dense attention mechanisms.

The results in table 4.6 uncover persistent challenges in low-frequency categories like windmills and low-contrast objects like swimming-pools, where even DSSFP struggles due to limited training data and ambiguous visual features. However, its consistent gains across structured ob-

Table 4.6: AP Scores on AI-TOD Test Set by Category

Object Categories	Method			
	FCOS [60]	CenterNet [61]	Cascade RCNN [104]	**DSSFP**
Airplane	14.30	17.43	25.57	37.9
Bridge	4.75	9.46	7.47	20.4
Storage-tank	19.8	25.93	23.33	47.2
Ship	22.24	21.86	23.55	40.5
Swimming-pool	0.65	6.21	10.81	16.2
Vehicle	12.51	16.54	14.09	34.5
Person	3.98	8.12	5.34	18.3
Windmill	0.17	1.94	0.00	0.08

jects, like bridges with a +15.7% AP over FCOS, highlight its robustness to scale variation, a key advantage for aerial imagery where object sizes vary drastically.

4.6.3 Qualitative Analysis

To complement our quantitative results, we conduct a qualitative analysis of model behaviour through visual examples and failure cases. Predictions are visualized on both VisDrone (Fig. 4.15) and AI-TOD (Fig. 4.16) datasets. The results demonstrate our model's ability to accurately detect small objects in dense, cluttered environments by effectively utilizing multi-scale contextual information. However, we observe that certain heavily occluded objects or those in extremely congested areas remain challenging cases that are occasionally missed.

4.7 Chapter Summary

This chapter presented an enhanced object detection framework that addresses critical challenges in small object detection through three key innovations: (1) an adaptive receptive field expansion mechanism for multi-scale feature capture, (2) a hierarchical feature refinement module to preserve fine-grained spatial details, and (3) a global context aggregation strategy to mitigate information loss in downsampling operations.

On VisDrone, the model achieves 26.3% mAP, marking a 17.4% relative improvement over the baseline while maintaining a lightweight parameter footprint. Notably, this method approaches the SOTA (28.4%), demonstrating its efficacy in detecting small and densely packed objects.

On AI-TOD, it sets a new SOTA of 27.8% mAP, a 3.4% absolute gain over prior methods (FM-RTDETR: 26.9%), demonstrating unparalleled capability for extremely small-object detection.

These findings demonstrate the effectiveness of the proposed architecture and provide insights for real-world object detection. However, the model struggles with specific categories, likely due to their extreme smallness and limited representation. Future work will focus on optimizing inference speed while maintaining detection performance and evaluating generalization on a dataset with varying object scales.

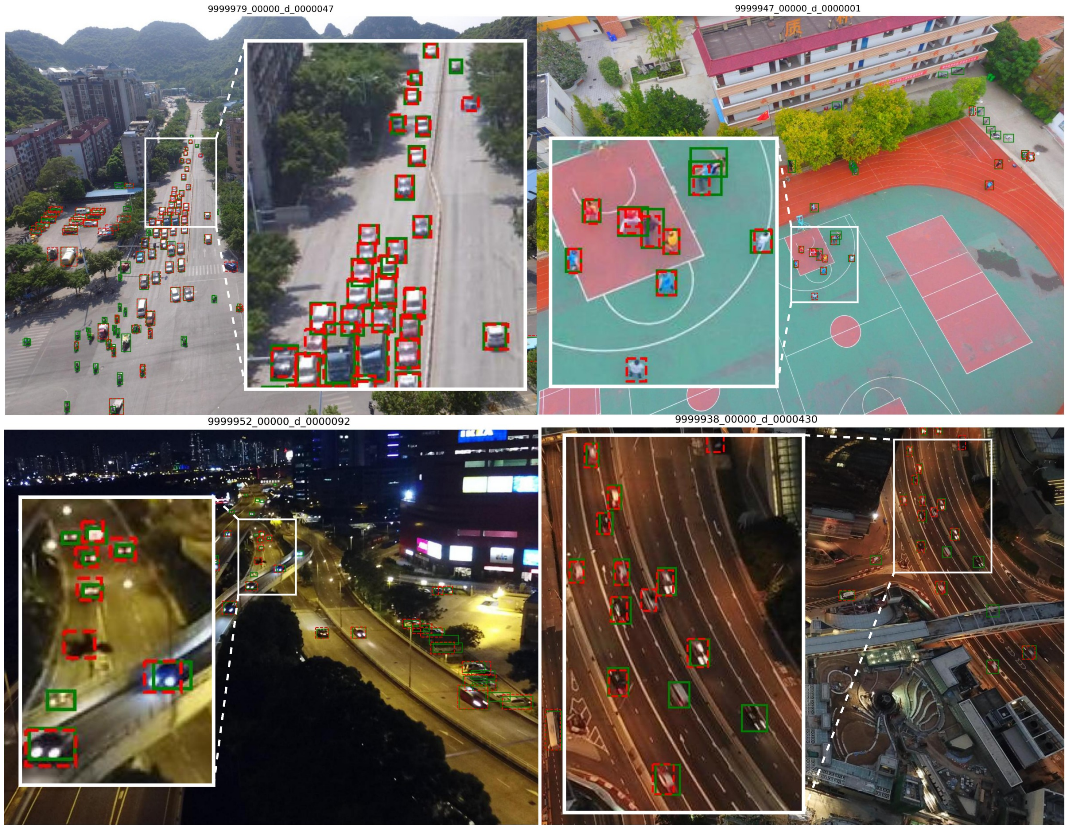


Figure 4.15: Qualitative results of the proposed method on the VisDrone test set. Only predictions with confidence scores larger than 0.3 are demonstrated. Legend:— Prediction — Ground-truth.

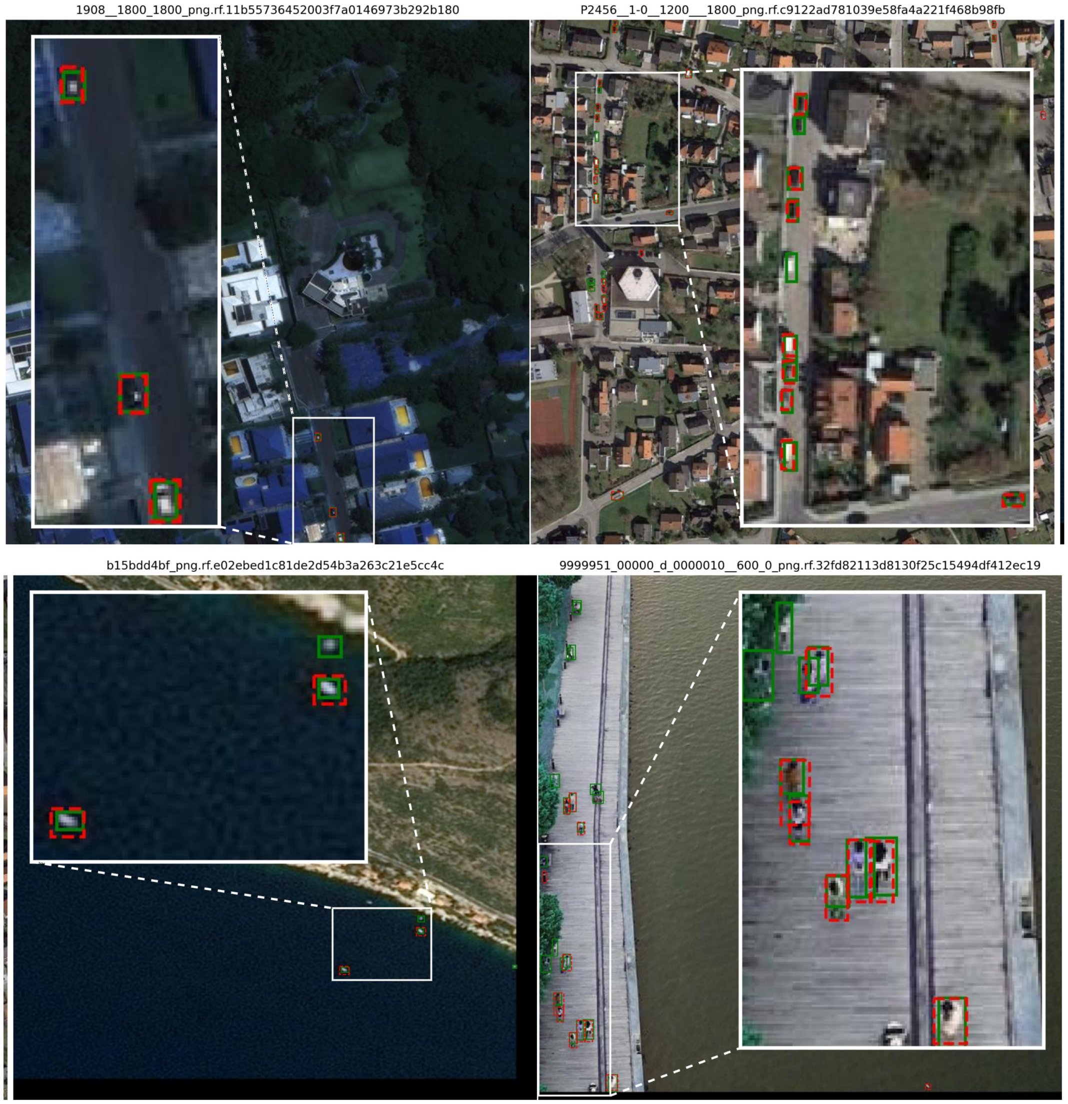

1908__1800_1800_png.rf.11b55736452003f7a0146973b292b180
P2456__1-0__1200___1800_png.rf.c9122ad781039e58fa4a221f468b98fb
b15bdd4bf_png.rf.e02ebed1c81de2d54b3a263c21e5cc4c
9999951_00000_d_0000010__600_0_png.rf.32fd82113d8130f25c15494df412ec19

Figure 4.16: Qualitative results of the proposed method on the AI-TOD test set. Only predictions with confidence scores larger than 0.3 are demonstrated. Legend:— Prediction — Ground-truth.

Chapter 5

Conclusion

Small object detection is challenging due to limited resolution, weak feature representation, and increased background noise associated with small-scale targets. Recognizing its importance in safety-critical domains such as autonomous driving and aerial surveillance, this thesis proposes two methodologies to improve detection performance.

Although the attention design in the first methodology was intentionally simple, its integration into the detection architecture demonstrates that even modest enhancements, when guided by cross-task learning, can lead to meaningful improvements in small object localization and hard-to-identify classes.

Furthermore, enhancing feature pyramid structures by introducing dilated strip depth-wise convolutions is effective at addressing the resolution–semantics trade-off that often hinders small object detection performance.

Collectively, these methodologies contribute to a flexible and modular architecture that can be adapted to other vision tasks or detection backbones. The techniques explored in this thesis highlight the value of combining semantic guidance, multi-scale feature fusion, and spatially adaptive receptive fields to address the inherent challenges of small object detection. The resulting architecture balances performance and efficiency, making it a strong candidate for deployment in practical, resource-constrained environments.

While the results are promising, several limitations remain. The quality of semantic features imputed to the detection model continues to constrain the overall performance ceiling. Additionally, although the architecture is lightweight, the added modules introduce computational overhead that must be considered for real-time applications. These findings point to important directions for future research.

Thus, future research will focus on three main directions: (i) improving the segmentation backbone to enhance the quality of semantic features, (ii) reducing computational latency for real-time deployment, and (iii) evaluating the methodologies across a broader set of datasets, such as COCO and PASCAL VOC, to test their generalizability across varying object densities and scales. Moreover, further investigation into advanced attention mechanisms could help improve segmentation and detection performance without increasing reliance on labelled data.

In conclusion, this thesis contributes a practical and extensible architecture through carefully designed feature fusion and context-aware enhancements. It offers valuable insights for improving the detection of small objects and lays the groundwork for future advances in multi-task learning and real-world object detection systems.

Bibliography

[1] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, “Detection and tracking meet drones challenge,” *IEEE Trans. on Patt. Analy. and Machine Intelligence*, vol. 44, no. 11, pp. 7380–7399, 2021.

[2] J. Wang, W. Yang, H. Guo, R. Zhang, and G.-S. Xia, “Tiny object detection in aerial images,” 2021, pp. 3791–3798.

[3] Z. M. Research, “Drone Market Size, Share, Demand, Trends and Growth 2030.” [Online]. Available: https://www.zionmarketresearch.com/report/drone-market-size

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

[5] A. Mathew, P. Amudha, and S. Sivakumari, “Deep learning techniques: an overview,” *Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2020*, pp. 599–608, 2021.

[6] 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*, vol. 45, no. 11, pp. 13 467–13 488, 2023.

[7] A. Rozantsev, “Vision-based detection of aircrafts and uavs,” Ph.D. dissertation, EPFL, 2017.

[8] J.-S. Lim, M. Astrid, H.-J. Yoon, and S.-I. Lee, "Small object detection using context and attention," in *2021 international Conference on Artificial intelligence in information and Communication (ICAIIC)*. IEEE, 2021, pp. 181–186.

[9] X. Ying, "An overview of overfitting and its solutions," in *Journal of physics: Conference series*, vol. 1168. IOP Publishing, 2019, p. 022022.

[10] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

[11] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 113–123.

[12] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[13] S. K. Ravindran and C. Tomasi, "Randmsaugment: A mixed-sample augmentation for limited-data scenarios," *arXiv preprint arXiv:2311.16508*, 2023.

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

[15] J. Redmon, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conf. on computer vis. and pattern recognition*, 2016.

[16] R. Padilla, S. L. Netto, and E. A. Da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 international conference on systems, signals and image processing (IWSSIP)*. IEEE, 2020, pp. 237–242.

[17] 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.

[18] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, “Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2874–2883.

[19] Y. Imamura, S. Okamoto, and J. H. Lee, “Human tracking by a multi-rotor drone using hog features and linear svm on images captured by a monocular camera,” in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, 2016, pp. 8–13.

[20] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, “Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 2778–2788.

[21] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, “Object detection with deep learning: A review,” *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[22] G. Cao, X. Xie, W. Yang, Q. Liao, G. Shi, and J. Wu, “Feature-fused ssd: Fast detection for small objects,” in *Ninth international conference on graphic and image processing (ICGIP 2017)*, vol. 10615. SPIE, 2018, pp. 381–388.

[23] Y. Cao, K. Chen, C. C. Loy, and D. Lin, “Prime sample attention in object detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 583–11 591.

[24] C. Chen, M.-Y. Liu, O. Tuzel, and J. Xiao, “R-cnn for small object detection,” in *Asian conference on computer vision*. Springer, 2016, pp. 214–230.

[25] K. Li and L. Cao, “A review of object detection techniques,” in *2020 5th International Conference on Electromechanical Control Technology and Transportation (ICECTT)*. IEEE, 2020, pp. 385–390.

[26] B. R. Solunke and S. R. Gengaje, “A review on traditional and deep learning based object detection methods,” in *2023 International Conference on Emerging Smart Computing and Informatics (ESCI)*. IEEE, 2023, pp. 1–7.

[27] A. B. Amjoud and M. Amrouch, “Object detection using deep learning, cnns and vision transformers: A review,” *IEEE Access*, vol. 11, pp. 35 479–35 516, 2023.

[28] 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.

[29] Y. Liu, P. Sun, N. Wergeles, and Y. Shang, “A survey and performance evaluation of deep learning methods for small object detection,” *Expert Systems with Applications*, vol. 172, p. 114602, 2021.

[30] R. Kaur and S. Singh, “A comprehensive review of object detection with deep learning,” *Digital Signal Processing*, vol. 132, p. 103812, 2023.

[31] S. S. Vasekar and S. K. Shah, “A method based on background subtraction and kalman filter algorithm for object tracking,” in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. IEEE, 2018, pp. 1–4.

[32] J. D. Pulgarin-Giraldo, A. Alvarez-Meza, D. Insuasti-Ceballos, T. Bouwmans, and G. Castellanos-Dominguez, “Gmm background modeling using divergence-based weight updating,” in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 21st Iberoamerican Congress, CIARP 2016, Lima, Peru, November 8–11, 2016, Proceedings 21*. Springer, 2017, pp. 282–290.

[33] F. El Baf, T. Bouwmans, and B. Vachon, “Fuzzy statistical modeling of dynamic backgrounds for moving object detection in infrared videos,” in *2009 IEEE computer society*

conference on computer vision and pattern recognition workshops. IEEE, 2009, pp. 60–65.

[34] Y. Dong, T. Han, and G. N. DeSouza, “Illumination invariant foreground detection using multi-subspace learning,” *International journal of knowledge-based and intelligent engineering systems*, vol. 14, no. 1, pp. 31–41, 2010.

[35] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)*, vol. 2. IEEE, 1999, pp. 246–252.

[36] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1. Ieee, 2001, pp. I–I.

[37] W. K. Mutlag, S. K. Ali, Z. M. Aydam, and B. H. Taher, “Feature extraction methods: a review,” in *Journal of Physics: Conference Series*, vol. 1591, no. 1. IOP Publishing, 2020, p. 012028.

[38] G. Kumar and P. K. Bhatia, “A detailed review of feature extraction in image processing systems,” in *2014 Fourth international conference on advanced computing & communication technologies*. IEEE, 2014, pp. 5–12.

[39] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, vol. 1. Ieee, 2005, pp. 886–893.

[40] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[41] R. Lienhart and J. Maydt, “An extended set of haar-like features for rapid object detection,” in *Proceedings. international conference on image processing*, vol. 1. IEEE, 2002, pp. I–I.

[42] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, pp. 273–297, 1995.

[43] L. Breiman, “Random forests,” *Machine learning*, vol. 45, pp. 5–32, 2001.

[44] Y. Freund and R. E. Schapire, “A desicion-theoretic generalization of on-line learning and an application to boosting,” in *European conference on computational learning theory*. Springer, 1995, pp. 23–37.

[45] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[46] A. A. M. Al-Saffar, H. Tao, and M. A. Talab, “Review of deep convolution neural network in image classification,” in *2017 International conference on radar, antenna, microwave, electronics, and telecommunications (ICRAMET)*. IEEE, 2017, pp. 26–31.

[47] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *2014 IEEE conf. on Computer vis. and Pattern Recognition*, 2014, pp. 580–587.

[48] R. Girshick, “Fast r-cnn,” in *Proceedings of the IEEE intel. conf. on computer vis.*, 2015, pp. 1440–1448.

[49] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[50] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proceedings of the IEEE conf. on computer vis. and pattern recognition*, pp. 770–778, 2016.

[51] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, “Deep high-resolution representation learning for visual recognition,” *TPAMI*, 2019.

[52] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer vis.–ECCV 2016: 14th European conf., Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[53] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[54] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *arXiv preprint arXiv:2410.17725*, 2024.

[55] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. of the IEEE Intl. Conf. on computer vision*, 2017, pp. 2980–2988.

[56] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," *Advances in neural information processing systems*, vol. 33, pp. 21 002–21 012, 2020.

[57] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9759–9768.

[58] L. Huang, Y. Yang, Y. Deng, and Y. Yu, "Densebox: Unifying landmark localization with end to end object detection," *arXiv preprint arXiv:1509.04874*, 2015.

[59] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 734–750.

[60] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: A simple and strong anchor-free object detector," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 4, pp. 1922–1933, 2020.

[61] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proc. of the IEEE/CVF Intl. Conf. on computer vision*, 2019, pp. 6569–6578.

[62] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conf. on computer vis.* Springer, 2020, pp. 213–229.

[63] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[64] 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," Aug. 2021, arXiv:2103.14030 [cs]. [Online]. Available: http://arxiv.org/abs/2103.14030

[65] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 568–578.

[66] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu *et al.*, "A survey on vision transformer," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 87–110, 2022.

[67] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM computing surveys (CSUR)*, vol. 54, no. 10s, pp. 1–41, 2022.

[68] A. Ivanov, N. Dryden, T. Ben-Nun, S. Li, and T. Hoefler, "Data movement is all you need: A case study on optimizing transformers," *Proceedings of Machine Learning and Systems*, vol. 3, pp. 711–732, 2021.

[69] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conf. on computer vis. and pattern recognition*, 2017, pp. 2117–2125.

[70] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, “Panet: Few-shot image semantic segmentation with prototype alignment,” in *proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9197–9206.

[71] G. Yang, J. Lei, Z. Zhu, S. Cheng, Z. Feng, and R. Liang, “Afpn: Asymptotic feature pyramid network for object detection,” in *2023 IEEE Intl. Conf. on Systems, Man, and Cybernetics (SMC)*. IEEE, 2023, pp. 2184–2189.

[72] M. Tan, R. Pang, and Q. V. Le, “Efficientdet: Scalable and efficient object detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 781–10 790.

[73] D. Hoiem, S. K. Divvala, and J. H. Hays, “Pascal voc 2008 challenge,” *World Literature Today*, vol. 24, no. 1, pp. 1–4, 2009.

[74] X. Hua, Z. Cai, X. Zhu, and O. Aijia, “Bikd-yolo: A small object detection algorithm from uav perspective based on biformer attention and knowledge distillation,” *2024 20th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 1–6, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:273226073

[75] Z. Nian, W. Yang, and H. Chen, “Aeffnet: Attention enhanced feature fusion network for small object detection in uav imagery,” *IEEE Access*, 2025.

[76] H. Zhang, K. Liu, Z. Gan, and G.-N. Zhu, “Uav-detr: Efficient end-to-end object detection for unmanned aerial vehicle imagery,” *arXiv preprint arXiv:2501.01855*, 2025.

[77] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in *Proc. of the IEEE Conf. on compu. vis. and pattern recognit.*, 2018, pp. 8759–8768.

[78] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proc. of the IEEE Intl. Conf. on computer vision*, 2017, pp. 2961–2969.

[79] Y. Su, Q. Liu, W. Xie, and P. Hu, “Yolo-logo: A transformer-based yolo segmentation model for breast mass detection and segmentation in digital mammograms,” *Computer Methods and Programs in Biomedicine*, vol. 221, p. 106903, 2022.

[80] X. Gao, G. Zhang, and Y. Xiong, “Multi-scale multi-modal fusion for object detection in autonomous driving based on selective kernel,” *Measurement*, vol. 194, p. 111001, 2022.

[81] 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*, 2017, pp. 5998–6008.

[82] W. Li, K. Liu, L. Zhang, and F. Cheng, “Object detection based on an adaptive attention mechanism,” *Scientific Reports*, vol. 10, no. 1, p. 11307, 2020.

[83] C. Shen, C. Ma, and W. Gao, “Multiple attention mechanism enhanced yolox for remote sensing object detection,” *Sensors*, vol. 23, no. 3, p. 1261, 2023.

[84] J.-S. Lim, M. Astrid, H.-J. Yoon, and S.-I. Lee, “Small object detection using context and attention,” in *2021 intel. conf. on Artificial intelligence in information and Communication (ICAIIC)*. IEEE, 2021, pp. 181–186.

[85] S. C. Yurtkulu, Y. H. Şahin, and G. Unal, “Semantic segmentation with extended deeplabv3 architecture,” in *2019 27th Signal Processing and Communications Applications conf. (SIU)*. IEEE, 2019, pp. 1–4.

[86] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “vis. meets robotics: The kitti dataset,” *The intel. Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[87] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset,” in *CVPR Workshop on the Future of Datasets in vis.*, vol. 2, 2015, p. 1.

[88] Y. Gefan and L. Yuchi, “Object detection in the kitti dataset using yolo and faster r-cnn.”

[89] J. Liu, Q. Cai, F. Zou, Y. Zhu, L. Liao, and F. Guo, “Biga-yolo: A lightweight object detection network based on yolov5 for autonomous driving,” *Electronics*, vol. 12, no. 12, p. 2745, 2023.

[90] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “Cbam: Convolutional block attention module,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[91] Q. Hou, L. Zhang, M.-M. Cheng, and J. Feng, “Strip pooling: Rethinking spatial pooling for scene parsing,” in *Proc. of the IEEE/CVF Conf. on compu. vis. and pattern recognit.*, 2020, pp. 4003–4012.

[92] X. Yuan, Z. Zheng, Y. Li, X. Liu, L. Liu, X. Li, Q. Hou, and M.-M. Cheng, “Strip r-cnn: Large strip convolution for remote sensing object detection,” *arXiv preprint arXiv:2501.03775*, 2025.

[93] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, “Ghostnet: More features from cheap operations,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1580–1589.

[94] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[95] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proc. of the IEEE Conf. on compu. vis. and pattern recognit.*, 2017, pp. 2117–2125.

[96] M. Hong, S. Li, Y. Yang, F. Zhu, Q. Zhao, and L. Lu, “Sspnet: Scale selection pyramid network for tiny person detection from uav images,” *IEEE geoscience and remote sensing letters*, vol. 19, pp. 1–5, 2021.

[97] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proc. of the IEEE/CVF Conf. on compu. vis. and pattern recognit.*, 2020, pp. 9729–9738.

[98] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," https://github.com/ultralytics/ultralytics, 2023, [Online].

[99] C.-Y. Wang, I.-H. Yeh, and H.-Y. Mark Liao, "Yolov9: Learning what you want to learn using programmable gradient information," in *European Conf. on computer vision*. Springer, 2024, pp. 1–21.

[100] C. Yang, Z. Huang, and N. Wang, "Querydet: Cascaded sparse query for accelerating high-resolution small object detection," in *Proc. of the IEEE/CVF Conf. on compu. vis. and pattern recognit.*, 2022, pp. 13 668–13 677.

[101] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conf. on computer vision*. Springer, 2020, pp. 213–229.

[102] B. Roh, J. Shin, W. Shin, and S. Kim, "Sparse detr: Efficient end-to-end object detection with learnable sparsity," *arXiv preprint arXiv:2111.14330*, 2021.

[103] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "Detrs beat yolos on real-time object detection," in *Proc. of the IEEE/CVF Conf. on compu. vis. and pattern recognit.*, 2024, pp. 16 965–16 974.

[104] Z. Cai and N. Vasconcelos, "Cascade r-cnn: High quality object detection and instance segmentation," *IEEE Trans. on pattern analysis and machine intelligence*, vol. 43, no. 5, pp. 1483–1498, 2019.

[105] S. Qiao, L.-C. Chen, and A. Yuille, "Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution," in *Proceedings of the IEEE/CVF Conf. on comp. vision and pattern recognition*, 2021.

[106] J. Wu, Z. Pan, B. Lei, and Y. Hu, "Fsanet: Feature-and-spatial-aligned network for tiny object detection in remote sensing images," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 60, pp. 1–17, 2022.

[107] M. Ma and H. Pang, “Sp-yolov8s: An improved yolov8s model for remote sensing image tiny object detection,” *applied sciences*, vol. 13, no. 14, 2023.

[108] S. Qi, X. Song, T. Shang, X. Hu, and K. Han, “Msfe-yolo: An improved yolov8 network for object detection on drone view,” *IEEE Geoscience and Remote Sensing Letters*, 2024.

[109] Y. Liu, Q. Ye, L. Sun, and Z. Wu, “Sod-yolov8n: Small object detection in remote sensing images based on yolov8n,” *IEEE Geoscience and Remote Sensing Letters*, 2025.

[110] Y. Yang, J. Dai, Y. Wang, and Y. Chen, “Fm-rtdetr: Small object detection algorithm based on enhanced feature fusion with mamba,” *IEEE Signal Processing Letters*, 2025.

Appendix

A Permission to Reprint

A.1 IEEE Permission to Reprint

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of Lakehead University's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html and https://www.ieee.org/publications/rights/author-rights-responsibilities.html to learn how to obtain a License from RightsLink.

A.2 Elsevier Permission to Reprint

In reference to Elsevier copyrighted material which is used with permission in this thesis, Elsevier does not endorse any of Lakehead University's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing Elsevier copyrighted material for advertising or promotional purposes, or for creating new collective works for resale or redistribution, please go to https://www.elsevier.com/about/policies/copyright/permissions to learn how to obtain a License from RightsLink.

B Source Code

The source codes of this thesis are available on GitHub.

For more information about the author’s publications, please refer to LinkedIn profile.

LinkedIn: LinkedIn.