Detection of Small Objects in UAV Images via an Improved Swin Transformer-based Model

by

Weidong Liang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2023

© Weidong Liang 2023

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Automated detection of small objects such as vehicles in images of complex urban environments taken by unmanned aerial vehicles (UAV) is one of the most challenging tasks in computer vision and remote sensing communities, with various applications ranging from traffic congestion surveillance to vision systems in intelligent transportation. Deep learning models, most of which are based on convolutional neural networks (CNNs), have been commonly used to automatically detect objects in UAV images. However, the detection accuracy is still often unsatisfactory due to the shortcomings of CNNs. For instance, CNN collects data from nearby pixels, but spatial information is lost due to the pooling operations. As such, it is difficult for CNNs to model certain long-range dependencies.

In this thesis, we propose a Swin Transformer-based model that incorporates convolutions with the Swin Transformer to extract more local information, mitigating the problem of small object detection from complex backgrounds in UAV images and further improving the detection accuracy. By using the Swin Transformer, our model leverages both the local feature extraction of convolutions and the global feature modeling of transformers. The framework was designed with two main modules, a local context enhancement (LCE) module and a Residual U-Feature Pyramid Network (RSU-FPN) module. The LCE module is used to implement dilated convolution and increase the receptive field of each image pixel. By combining with the Swin Transformer block, it can efficiently encode various spatial contextual information and detect local associations and structural information within UAV images. In addition, the RSU-FPN module is designed as a two-level nested U-shaped structure with skip connections to integrate multi-scale feature maps. A loss function combining normalized Gaussian Wasserstein distance and L1 loss is also introduced, which allows the model to be trained using imbalanced data. The proposed method was compared with the state-of-the-art methods on the UAVDT dataset and Vis-Drone dataset. Our experimental results obtained on the UAVDT dataset indicated that our proposed method increased the average precision (AP) by 21.6%, 22.3% and 25.5% over Cascade R-CNN, PVT and Dynamic R-CNN detectors, respectively, demonstrating its effectiveness and reliability on small object detection from UAV images.

Acknowledgements

I would like to thank Prof. Jonathan Li for the support during my master's study. I would also like to thank Prof. Linlin Xu, Prof. John Zelek, and Prof. Michael A. Chapman for being my thesis committee members.

I would like to thank Kyle Gao and Dening Lu in Geospatial Intelligence and Mapping Lab for supporting me make this thesis possible.

Dedication

This is dedicated to my parents and friends.

Table of Contents

Author's Declaration ii

Abstract iii

Acknowledgements iv

Dedication v

List of Figures ix

List of Tables xi

List of Abbreviations xii

1 Introduction 1

1.1 Statement of Problems 1

1.2 Objectives and Contributions 3

1.3 Thesis Outline 4

2 Literature review 5

2.1 Traditional Object Detection 5

2.2 Generic Object Detection 6

2.3 Object Detection in UAV Images 9

2.4 Feature Fusion Networks . . . 10
2.5 Data Augmentation . . . 11
2.6 Chapter Summary . . . 12

3 Improved Swin Transformer-based Model 13
3.1 Overview Framework . . . 13
3.2 Local Context Enhancement Swin Transformer . . . 14
3.2.1 Swin Transformer Block . . . 14
3.2.2 Local Context Enhancement Module . . . 15
3.2.3 Local Context Enhancement Swin Transformer-Based Network Structure . . . 17
3.3 RSU-FPN . . . 18
3.3.1 U-block Architecture . . . 18
3.3.2 Residual U-FPN Network . . . 19
3.4 Bounding Box Regression Loss Function . . . 19
3.4.1 Normalized Gaussian Wasserstein Distance Loss . . . 19
3.4.2 L1 Loss . . . 20
3.4.3 Combined Regression Loss . . . 21
3.5 Experiments . . . 21
3.5.1 Datasets . . . 21
3.5.2 Implementation . . . 24
3.5.3 Evaluation Metrics . . . 25

4 Results and Discussions 26
4.1 Quantitative evaluation . . . 26
4.2 Qualitative evaluation . . . 30
4.3 Ablation study . . . 35
4.3.1 Effect of LCEST backbone . . . 36
4.3.2 Effect of RSU-FPN . . . 39
4.3.3 Effect of Combined Regression Loss . . . 39

5 Conclusions **41**

5.1 Summary 41

5.2 Future Work 42

References **43**

List of Figures

1.1 RGB UAV image with bounding boxes of vehicles overlay in blue. 3

3.1 The overall architecture of Swin-RSUFPN. 13
3.2 Illustration of the shifted window in Swin Transformer Block. 15
3.3 Two Successive Local Context Enhancement Swin Transformer Blocks. . . 16
3.4 The overall architecture of Local Context Enhancement Swin Transformer. 17
3.5 Architecture of U-block. 18
3.6 Annotated examples of different scenarios in UAVDT dataset. (a) Night with poor visibility. (b) Nightlight condition. (c) Day with good visibility. (d) Smoggy weather. 22
3.7 Distribution of UAVDT dataset. 23
3.8 Annotated examples of different scenarios in VisDrone dataset. 24

4.1 The precision-recall curves exhibit superior performance in each category on the VisDrone dataset. 29
4.2 Detection results of different methods on VisDrone dataset. 31
4.3 Detection results of different methods under congested intersection road conditions. 32
4.4 Detection results of different methods in smoggy weather conditions. 33
4.5 Detection results of different methods in the nighttime scenarios. 34
4.6 The visualization of detection results on the UAVDT dataset by progressively incorporating LCEST backbone, RSU-FPN, and Combined loss to the baseline. 37

4.7 The visualization of detection results on the VisDrone dataset by progressively incorporating LCEST backbone, RSU-FPN, and Combined loss to the baseline. 38

List of Tables

3.1 Distribution of the bounding boxes in VisDrone train and validation set. . 23

4.1 Evaluation metrics (%) for different models on the UAVDT dataset. 26
4.2 Evaluation metrics (%) for different models on the Vis-Drone dataset. . . . 27
4.3 Evaluation metrics (%) for ablation study on UAVDT dataset. 35
4.4 Evaluation metrics (%) for ablation study on VisDrone dataset. 35

List of Abbreviations

ClusDet Clustered Detection Network

CRENet Cluster Region Estimation Network

DMNet Density-Map Guided Object Detection Network

Faster R-CNN Faster Region-based Convolutional Neural Network

FPN Feature Pyramid Netw

FS-SSD Scaling-Based Single Shot Detector

GANs Generative Adversarial Neural Networks

HOG Histogram of Oriented Gradients

HRDNet High-Resolution Detection Network

LCEST Local Context Enhancement Swin Transformer

MPFPN Multi-Branch Parallel Feature Pyramid Network

R-CNN Region-based Convolutional Neural Networks

RPN Region Proposal Network

SSD Single Shot Multi-Box Detector

SURF Speeded Up Robust Features

SIFT Scale-Invariant Feature Transform

SW-MSA Shifted Window Multi-head Self-Attention

SVM Support Vector Machine

ViT Vision Transformer

W-MSA Window Multi-head Self-Attention

Yolo You Only Look Once

Chapter 1

Introduction

1.1 Statement of Problems

With the popularization and commercialization of drones, the cost of using airborne remote sensing decreases year-by-year. The significance of UAV images has permeated many industries, including disaster monitoring [1],[2], ecological environmental conservation [3], mineral resource exploration, and public safety. The UAV image contains an extensive amount of useful information that can be retrieved and used in various sorts of research, both in science and technology. Automatic object detection is an essential direction for computer vision since it can reduce human resource consumption while processing images efficiently and intelligently. In the field of computer vision, traditional object detection methods are usually based on low-level features from a class-specific learner (e.g. SVM) to predict a single class of images using sliding window methods and constructed features such as Histogram of Oriented Gradient (HoG) [4] and Scale-Invariant Feature Transform (SIFT) [5]. However, real-world surroundings may be too complicated and variable for traditional object detection methods to process. Objects may be covered by additional items or environmental elements such as shadows and reflections. Furthermore, influenced by various factors, for instance, the weather condition, light intensity, and the parameters of sensors, the current demands for object detection tasks in UAV imagery cannot be satisfied by conventional approaches.

In recent years, the field of object identification of natural images has made significant strides forward with the advance of deep learning. The utilization of deep learning techniques has revolutionized the area of object detection, providing a precise and effective method to recognize objects in complicated and constantly changing surroundings.

Convolutional neural networks (CNNs) possess remarkable capabilities for adapting to new data and extracting meaningful features and outperform traditional object detection methods. Two-stage methods, for instance, R-CNN [6], Faster R-CNN [7], Cascade R-CNN [8], and Dynamic R-CNN [9] use region proposal and object classification. These methods identify potential regions in the image and use a classifier to assess the presence of objects within the regions identified in the first stage, and further categorize the object. However, most CNN-based models prioritize complex architectural designs like residual learning [10, 11] and dense connections [12]. CNN-based models are intended to process specific, nearby areas of an image and might not be proficient at obtaining a holistic understanding of the global contextual information in which an image appears.

With attention mechanisms [13] making a big splash, transformer-based models became more and more utilized in object detection tasks [14, 15] to compensate for the incapacity of CNN models to model long-range dependencies between objects and features in an image. Dosovitskiy [16] produces a method without convolutions called vision transformer (ViT), which uses self-attention mechanisms to capture global dependencies between image patches. Swin Transformer [17] utilizes a shifted window scheme that facilitates the modeling of long-range dependencies in the image data.

Although gratifying results have been achieved in general object detection, the ability of transformers to detect small objects on UAV images remains limited. The performance of the transformer-based model on the UAV image dataset is far from satisfactory in terms of accuracy and efficiency. Several special challenges cause difficulties in aerial image detection: Compared with the image, the target usually has a smaller scale. Unlike objects with appropriate proportions, small objects pose more significant challenges because of their small size and the complex background, which will seriously deteriorate the feature representation, thus challenging the most advanced object detector [18]. Compared to natural images, the performance of representation learning for UAV images is negatively impacted by scale differences. As images are transmitted in the models, their resolution may degrade and the images may become blurred after being encoded and decoded, resulting in small objects merging into the background. Hence, detectors must be designed to distinguish small objects from the surrounding background. Moreover, it is typical for the small targets to be sparsely and unevenly distributed across the entire image. Figure 1.1 shows the examples captured by a UAV platform with a resolution of 1080 x 540 pixels.

In the realm of object detection, one of the most significant hurdles is effectively processing features across multiple scales. Early methods of detection relied on pyramid feature hierarchies extracted from backbone networks to make predictions. Feature Pyramid Network (FPN) [19] was one of the first techniques to propose a top-down approach for combining multi-scale features. Since then, other methods such as PANet [20], STDN [21],

Figure 1.1: RGB UAV image with bounding boxes of vehicles overlay in blue.

M2det [22], and BiFPN [23] have been developed to further improve multi-scale feature fusion. More recently, NAS-FPN [24] utilized neural architecture search to automatically design feature network topology. However, this approach required significant computational resources and produced a network that was challenging to interpret. In this thesis, we present a novel RSU-FPN network, which provides a more intuitive and systematic approach to enhancing feature fusion across multiple scales.

1.2 Objectives and Contributions

We intend to develop a new model to mitigate the problem of detecting small-scaled objects which appear relatively small in size and have fine details that may be difficult to discern in UAV images. Moreover, we intend to conduct a comparative analysis on two public datasets. Furthermore, we plan to implement quantitative analysis and compare the metrics to show the extent of improvement to existing models.

In this work, we develop an object detection method for recognizing UAV images based on Swin-Transformer [17], which detects not only the bounding box of identified small objects but returns their classification. We propose a Local Enhancement Module in conjunction with Swin Transformer to boost the network's ability of perceiving the local

context. The motivations of this work are (1) to develop a new model for better detection of small objects in UAV images to support urban surveillance and (2) to design a better loss function for quantifying the similarity of two bounding boxes.

The major contributions of this work are listed below:

First, a Local Context Enhancement module is introduced into Swin Transformer to enhance the local perception.

Next, we propose a new multiscale feature fusion module, i.e., RSU-FPN for small object detection from UAV images.

In addition, we propose a new loss function by combining L1 loss with Normalized Gaussian Wasserstein Distance for training unbalanced samples.

Finally, our method achieves state-of-the-art performance on UAVDT [25] dataset and Vis-Drone [26] dataset.

1.3 Thesis Outline

This thesis is divided into five chapters. Each chapter is constructed as follows. Chapter 1 introduces the statement of challenges of object detection on UAV images, the objectives of the study, and the main contributions in this work. Chapter 2 presents a summary of frequently used deep learning approaches for object identification in both natural images and UAV images, coupled with comparisons of the strengths and shortcomings of various methods. Chapter 3 illustrates the conceptual framework of the proposed method as well as the evaluation metrics. Two mainstream datasets are used to evaluate the performance of our method. The results obtained from the quantitative evaluation, qualitative evaluation, and ablation study are exhibited in Chapter 4. Finally, Chapter 5 summarizes the conclusions and recommendations of the work and indicates several potential improvements in the directions for future works.

Chapter 2

Literature review

This section provides a review of relevant research in relation to the proposed study, primarily covering five topics: traditional object detection methods, generic object detection, object detection in UAV images, feature fusion networks, and data augmentation.

2.1 Traditional Object Detection

Conventional approaches for object detection rely on manually crafted and extracted features. Additionally, simpler machine learning techniques such as SVM, decision trees, and random forests are commonly used. These methods follow a common architecture that can be divided into three main stages: proposal creation, feature representation extraction, and classification. In the proposal creation stage, the algorithm often employs sliding windows to locate areas within the image that are presumed to contain objects of interest. The feature representation extraction stage involves extracting feature vectors from the regions of interest identified in the previous stage. During the feature extraction stage, feature vectors are obtained and subsequently encoded using descriptors like HOG [4], Haar [27], SIFT [5], or Speeded Up Robust Features (SURF) [28]. Finally, the classification stage is trained to establish a correlation between object labels and the regions that have been suggested and encoded by feature descriptors.

P. Viola and M. Jones [29] utilizes Haar-like [27] features, integral image calculations, Adaboost, and a cascading classifier. To detect Haar-like features, the algorithm slides a small window across the input image and computes integral images to reduce the computational complexity. Each Haar-like feature is then evaluated using a trained Adaboost

classifier, which identifies the classifier associated with that feature. Finally, the classifiers are combined using a cascading approach, in which each stage has a set of classifiers that are applied in sequence to filter out negative samples.

Navneet Dalal and Bill Triggs [4] made substantial progress in feature extraction and object detection by enhancing the Scale-Invariant Feature Transform (SIFT) [5] algorithm and introducing a new feature descriptor known as the Histogram of Oriented Gradients (HOG) [4]. After the image has been divided into small, interconnected regions(cells), a histogram is created to represent the angle directions or edge orientations of the pixels within each cell. The histogram is generated based on the gradient orientation. Subsequently, the cells are partitioned into distinct bins. Adjacent cells are then clustered together within the same spatial region. Finally, the collection of normalized histograms forms the block histogram, and the set of these block histograms constitutes the descriptor, which serves as the basis for histogram aggregation and normalization in the HOG method.

Despite the fact that traditional object detection algorithms are used less frequently in contemporary research, they continue to be influential in history. The techniques and concepts introduced by these algorithms have served as crucial guidance and inspiration for the advance of modern deep learning algorithms. For instance, the Non-Maximum Suppression [30] algorithm facilitates the elimination of duplicated prediction anchors. However, the traditional object detection method has some drawbacks that limit its effectiveness in modern applications. An example of a limitation of the traditional object detection method is that it often generates a significant number of proposal anchor boxes that are either redundant or invalid. These boxes may not accurately match with any actual object of interest and can result in increased processing time and higher computational costs. Furthermore, manually designed features are often based on prior knowledge or assumptions about the data, and may not adapt to new or changing environments. As a result, traditional object detection methods may struggle to achieve high accuracy in challenging or unfamiliar scenarios.

2.2 Generic Object Detection

Deep learning has led to considerable progress in object detection in recent years. Deep neural network-based object detection techniques can be categorized into two-stage and one-stage detectors [31]. The two-stage method generates region proposals and predicts categories for each region proposal. For one stage detector, the process of object classification and bounding-box regression is performed directly on the image without relying on pre-generated region proposals.

Two-stage detectors divide the process into two distinct steps: region proposal and object classification. Ren et al. [7] put forward RPN which uses the convolutional neural network to generate region proposals directly and identifies the objects based on the proposals extracted from RPN. As it increased the efficiency and precision of object identification tasks, it is regarded as an important advancement in computer vision. A region proposal network (RPN) is used by Faster R-CNN [7] to provide region proposals for the objects in an image. The object proposals are then classified and improved using a region-based convolutional neural network (R-CNN), which receives the input from these proposals. While the RPN and R-CNN share convolutional layers, the model is quicker and more effective. In comparison to earlier models, this method enables Faster R-CNN to provide state-of-the-art results on a number of benchmark datasets at a much lower computational cost. The invention of the Faster R-CNN has inspired following object identification research and is now a well-liked option in practical applications, including as self-driving automobiles, video surveillance, and face recognition. Terrail [32] utilizes the Faster R-CNN algorithm to identify vehicles in the infrared images in VEDAI [33] dataset, and their approach yields an average precision of 77.8% and a recall of 31.04% in detecting objects.

Mask R-CNN [34] is a sophisticated model that builds upon the Faster R-CNN model. It employs a two-stage approach like Faster R-CNN and incorporates a segmentation branch to improve the primary model. The model begins by using a region proposal network (RPN) to generate proposals for regions of interest in the image. Then, a region-based convolutional neural network (R-CNN) is utilized to classify and refine the proposed regions. Additionally, the model enhances ROI Pooling with the use of ROI Align, which employs bilinear interpolation to acquire pixel values. Overall, Mask R-CNN represents a significant advancement in object detection and segmentation technology. Mask R-CNN has the advantages of simple structure, good flexibility, and remarkable effect. Vemula et al. [35] perform Mask RCNN on the powerline dataset and achieve good performance.

In the cascade [8] architecture, the initial detector produces a substantial number of region proposals using less strict thresholds. The subsequent detector applies more rigorous thresholds to filter the proposals generated by the previous stage. Finally, the most stringent detector only accepts a few proposals for further processing and object detection. The Cascade R-CNN detection network uses the initial regression of the anchor to obtain the input Region of Interest (RoI) for the first detection head. Nevertheless, the direct use of multiple classifiers makes it challenging to train a regressor that produces accurate results for potentially perplexing categories [36].

Dynamic R-CNN [9] is a SOTA object detection model which is proposed to autonomously modify the criteria for assigning labels and the loss function based on the candidate proposals. The dynamic architecture enables more efficient utilization of the

training data and challenges the model to accommodate a greater variety of high-quality data inputs.

While the two-stage algorithm is known for its high accuracy in object detection, it is computationally expensive and can hinder its real-world application. Unlike two-stage detectors, single-stage object detectors consider the object detection problem as a regression task. One-stage object detection methods, such as those in the YOLO [37] family, have gained significant popularity in recent years due to their efficiency and high speed.

Redmon [37] introduces a novel approach that merges the feature extraction and localization part into a holistic structure. The YOLO architecture partitions the input into a grid and instructs each grid cell to predict the category probabilities and the localization of the object within that cell. Furthermore, it does not require external region proposal techniques to generate potential object locations, which speeds up the inference time. However, the limitation of the Yolo network is that it can only identify a finite number of objects in each grid, which makes it challenging to achieve dense predictions and it is not accurate in identifying small targets. Redmon et al. propose an improved version YOLOv2 [38] which involves the use of Darknet for implementation and achieves a mean average precision (mAP) of 76.8% on the Pascal VOC 2007 [39] dataset. In addition, the concept of feature pyramid networks (FPN) is introduced in YOLOv3 [40] to detect objects at different scales and smaller objects that are not easily detected by earlier versions of YOLO.

Ning et al. [41] add Pyramidal Feature Hierarchy, predicting the object on a feature map with different receptive fields. The SSD model employs a multi-scale feature map technique to find objects in the image at various sizes and positions and utilizes pre-defined anchor boxes of varying sizes and aspect ratios to anticipate the locations of objects. Soleimani [42] uses an SSD detector to generate object regions of interest from low-altitude aerial images. One significant challenge faced by the SSD network is the computational complexity associated with detecting feature maps of different scales. Furthermore, when confronted with occlusion and background noise in highly dense crowds, SSD will achieve unsatisfactory results. SSD involves utilizing low-level feature maps to predict small objects, due to their limited receptive fields, they cannot effectively capture high-level semantic information from the surrounding, resulting in the inaccurate detection of partially obstructed objects. To address the issue, Wang et al. integrate the channel-wise attentional module into the existing SSD model. Lu et al. [43] propose a feature fusion SSD to enhance the detection performance on NWPU VHR-10 dataset [44]. These algorithms are aimed at improving the accuracy of detecting small targets. As the literature mentioned above, one-stage detectors are appropriate for balancing speed and accuracy. However, the size of the object is a significant challenge when it comes to object detection in UAV images.

2.3 Object Detection in UAV Images

Despite significant advancements have been achieved in object detection in natural scenes, the performance in high-resolution UAV images is not satisfactory. UAV image object detection poses greater challenges compared to general object detection due to several factors. First, small objects comprise a larger portion of the UAV image dataset. In addition, UAV images typically offer a top-down perspective and broad coverage area because of the elevated viewpoint from high altitudes. Due to variations in camera viewpoint, the size of objects can vary greatly both within an image and across different categories. Therefore, researchers specialize in searching for frameworks for detecting small objects.

ClusDet [31] is designed as an end-to-end cluster detection framework to concurrently tackle the challenges associated with detecting objects in aerial images, including the variations in object size and the sparse distribution of objects in the scene. Furthermore, an efficient ScaleNet is introduced to mitigate the heterogeneous size problem in densely populated regions, leading to the improvement of object detection. A multi-object image is divided into numerous images with fewer objects and extends or fills the photographs based on the respective sizes of the split images. However, the effectiveness of this method is heavily influenced by model parameter selection, including but not limited to the determination of the number of clusters (N) which must be decided beforehand and cannot be adjusted later [45]. If the overall count of objects within the image is greater than N, certain regions will undoubtedly be disregarded, forcing these overlooked zones to be inaccurately identified, leading to a decline in detection accuracy. In addition, ClusDet is specifically designed for object detection in aerial images, it may not perform as well on other datasets.

General object detectors often struggle to accurately detect and count the number of people in highly dense environments. To cope with the issue, Li et al. [46] incorporate density maps into aerial image object detection. A density map-guided image cropping method is introduced to leverage contextual and spatial information among objects, leading to superior object detection accuracy. Moreover, an efficient algorithm is put forward for producing image crops without the need for extra deep neural network training. The density map is subjected to a sliding window to obtain the total pixel intensity, which is then compared to a preset threshold to create a mask, and areas with density values exceeding the threshold are selected to form image crops. The output is produced by combining the results from the image crops and the original image. The limitation of the method is that the establishment of reliable density maps is a complicated procedure and may call for extra resources and knowledge. As a result, the usefulness of this approach may be restricted in some circumstances.

The shooting altitude of UAVs may vary substantially. As a consequence, objects in the same category might vary significantly in size, making anchor-based detectors difficult to establish the anchor size. Thus, it is important to minimize the variation in object size among images. To solve the issue, clustering algorithms are used in CRENet [47] to search for regions containing dense targets. By calculating the difficulty value of each clustered region, difficult regions are mined, and simple clustered regions are eliminated to improve detection speed. The experimental results indicate that CRENet achieves good performance on the VisDrone [48] dataset. However, the selection of the clustering parameters results in a significant influence on the method's performance and the adaptive searching mechanism utilized to discover object candidates is critical to the efficacy of this approach. Omitted detections and erroneous object localizations may occur if the adaptive searching procedure is not sufficiently strong.

Moreover, the motion blur caused by flying at high speeds and low altitudes can affect densely packed objects, making it even more challenging to distinguish individual objects accurately. To solve the issue, TPH-YOLOv5 [49] is added an additional prediction head to identify distinct-scale objects and the Transformer Prediction Heads (TPH) are introduced to investigate the predictive capabilities of the self-attention mechanism. Additionally, the convolutional block attention model (CBAM) is incorporated to identify regions of attention on the scenes containing high-density targets. However, the use of an additional prediction head results in a significant requirement for computing resources and can be time-consuming.

2.4 Feature Fusion Networks

To enhance the recognition of tiny and dense objects in UAV images, researchers have suggested several ideas such as enhancing feature maps, utilizing context information, and employing data enhancement techniques. By implementing these concepts, numerous multi-scale object detection techniques have been developed, including FS-SSD [50], HRDNet [51], and MPFPN [52].

Feature Pyramid Network(FPN) [19] takes input images of any size and produces feature maps at multiscale that are proportional to the input image size and employs a combination of feature extraction and feature aggregation through lateral connections to integrate high-resolution and on a semantic level deficient characteristics alongside low-resolution and on a semantic level robust characteristics [50]. Nuri [53] utilizes Mask R-CNN with FPN to extract trees from high-resolution RGB UAV images, which sustains a considerable degree of accuracy in the detection.

Liu et al. [51] introduce a new high-resolution detection model, which takes image hierarchy and feature hierarchy into account. HRDNet creates the MD-IPN, a module with multiple depths and streams, to achieve a balance in performance across small, medium, and large objects. Additionally, another novel module, MS-FPN, is proposed to appropriately integrate the multiscale features, which combines diverse semantic representations from these feature groups at various scales and achieve SOTA performance on MS COCO2017 [54] dataset.

Liang et al. [50] introduce a single-shot detector that employs feature fusion and scaling. FS-SSD creates two feature pyramids to detect small objects through the optimization of the feature fusion module and the incorporation of the deconvolution operation alongside the average pooling operation. The addition of average pooling can aid in mitigating network overfitting by reducing the overall number of parameters and offering background image information.

Current object detectors that rely on deep learning typically use feature extraction networks with a significant down-sampling factor to get high-level characteristics. However, this often results in the loss or disappearance of feature information for smaller objects because of the limited number of pixels in low-resolution feature maps. Therefore, two extra parallel branches are added in MPFPN [52], which perform up-sampling and lateral connections from their respective layers. The subsequent pyramid layers are merged to collect their features. The Supervised Spatial Attention Module (SSAM) integrated into MPFPN is intended to counter the presence of complicated background noise and accentuate the foreground information.

2.5 Data Augmentation

Data augmentation is an effective method for overcoming a dearth of training data and is now extensively employed in a variety of deep learning applications. It artificially extends the dataset by generating more equivalent data from a limited amount of data, thereby increasing the size and diversity of the dataset. The commonly used technique for addressing the problem of class imbalance is data augmentation. Along with conventional techniques such as flipping and rotating [55], various new data augmentation methods have also been put forward. Currently, common methods of data augmentation such as Mosaic [56], Mixup [57], and CopyPaste use various approaches to merge the pixel information of several images into one single image in order to enhance the overall information available in the image. A method called SamplePairing [58] has been introduced, which involves

overlaying two different images and computing the average pixel values. The approach using Generative Adversarial Neural Networks (GANs) [59] involves mixing and combining different images to generate fresh training samples. AdaResampling [60] utilizes a segmentation CNN to produce a contextual map beforehand and arranges the objects based on their scale and position. Manifold Mixup [61] involves training neural networks on the combined linear representations of hidden features from the training examples. In AlignMixup [62], feature tensors are explicitly aligned to create soft correspondences between two images.

2.6 Chapter Summary

In accordance with the findings of the literature review, deep learning methods outperform traditional object detection methods. While one-stage detectors are known for their high speed, they often exhibit lower detection accuracy compared to two-stage detectors. Therefore, in this thesis, we decide to propose a new two-stage model which utilizes the advantages of the deep learning method and the feature fusion network to further improve the accuracy and performance on UAV image datasets.

Chapter 3

Improved Swin Transformer-based Model

3.1 Overview Framework

In this section, a Swin-RSU network that utilizes the standard backbone–neck–head structure is introduced. Figure 3.1 demonstrates the architecture of Swin-RSUFPN network. The input image is processed by the proposed Local Context Enhancement Swin Transformer (LCEST) backbone. Next, an RSU-FPN is designed to extract multi-scale feature information for fusion to improve the accuracy of tiny object detection in the neck. Finally, the most widely used head of Faster R-CNN is utilized to perform feature map classifications and bounding box regression.

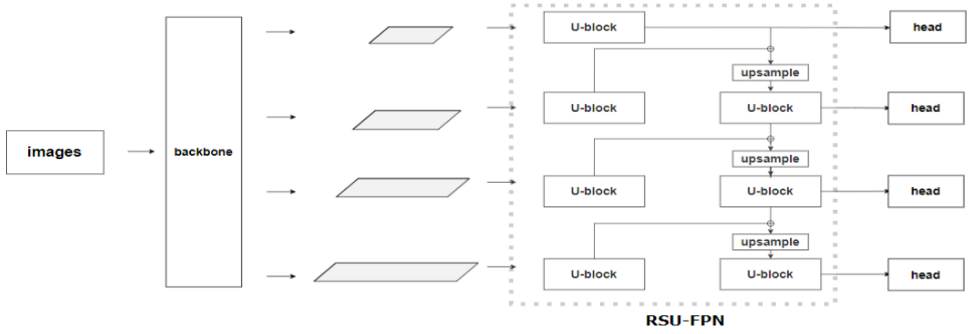


Figure 3.1: The overall architecture of Swin-RSUFPN.

3.2 Local Context Enhancement Swin Transformer

3.2.1 Swin Transformer Block

The Swin Transformer [17] block is a fundamental component of the Swin Transformer architecture's backbone. It is a variation of the ordinary Transformer block that processes input features using multi-head self-attention and feedforward neural networks. Several essential modules are introduced in the Swin Transformer block, for instance, the layer normalization module, multi-layer perceptrons module, and multi-head self-attention (MSA) modules that may be modified using either the conventional windowing (W-MSA) or shifting windowing (SW-MSA) technique. To process an input with feature size H×W×C, Swin Transformer initially separates it into distinct windows that do not overlap one another., each consisting of M×M patches. where $\frac{H}{M} \times \frac{W}{M}$ refers to the overall count of windows. Self-attention is calculated for every window to create the attention output of W-MSA. To establish connections between windows, the self-attention operation is applied to each individual window by moving the feature by $(\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$ before partitioning. With a local window feature $X \in \mathbb{R}^{M^2 \times C}$, the Q, K and V matrices are constructed as follows:

$$Q = XP_Q, \quad K = XP_K, \quad V = XP_V, \tag{3.1}$$

The matrices P_Q, P_K, and P_V are utilized in the Swin Transformer for projection purposes and are common to multiple windows, allowing for efficient computation and reduced parameterization. As a result, employing the mechanism of self-attention within a specific local window, the attention matrix is computed as:

$$\text{Attention}(Q, K, V) = \text{SoftMax}(QK^T/\sqrt{d} + B)V, \tag{3.2}$$

the relative positional encoding B is capable of being trained.

The Swin Transformer utilizes non-overlapping local windows to perform self-attention computation by shifting the window partition between two subsequent levels in the hierarchical map [63]. As shown in Figure 3.2, both layers on the left and right sides possess the same window size. The left layer adopts the window partitioning method, commencing from the upper left corner, and dividing 8x8 feature maps evenly into 2x2 windows, with each window size of 4x4. As shown on the right side in Figure 3.2, shifted window technique is used to generate new windows. These cross-window connections across overlapping windows are like the behavior of the CNN.

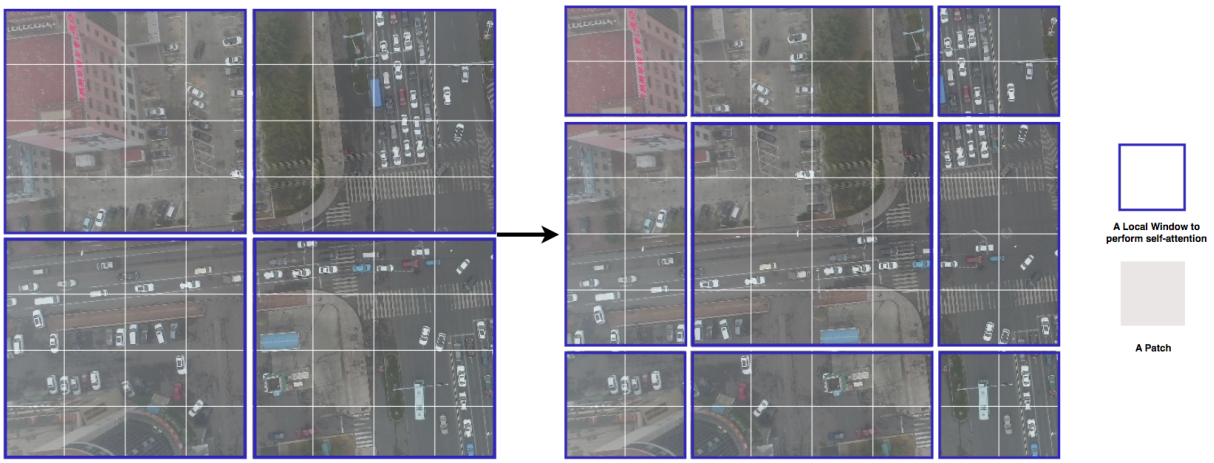


Figure 3.2: Illustration of the shifted window in Swin Transformer Block.

3.2.2 Local Context Enhancement Module

Position coding in the transformer is unsatisfactory for detecting local associations and structural information in images. Despite having a shifted window approach with sequential layers and a hierarchical structure, the Swin transformer still struggles to effectively encode a wide variety of spatial context information. To solve this issue, we proposed Local Context Enhancement Module that can be utilized in combination with the Swin Transformer block. As illustrated in Figure 3.3, a dilated convolution layer followed by a batch normalization layer and a RELU activation function is inserted in front of the Swin Transformer block. Dilated convolution can widen the receptive field of spatial images, making it possible to more accurately encode the multi-scale contextual information, effectively resolving the issue of inadequate spatial information extraction. A bigger number for the receptive field of a neuron indicates that it has access to a broader range of original images, suggesting that it may include information at a higher semantic level.

With the Local Context Enhancement Module applied, the notation of consecutive

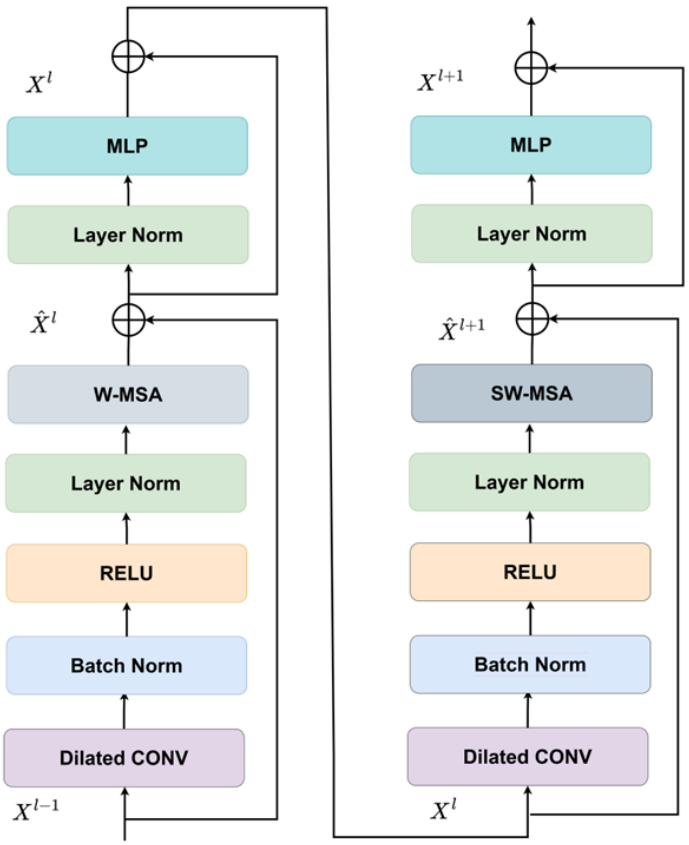


Figure 3.3: Two Successive Local Context Enhancement Swin Transformer Blocks.

Local Context Enhancement Swin Transformer blocks is computed as

$$
\begin{aligned}
\tilde{\mathbf{x}}^l &= \text{RELU}(\text{BN}(DCONV(\mathbf{x}^{l-1})), \\
\hat{\mathbf{x}}^l &= \text{W-MSA}\left(\text{LN}\left(\tilde{\mathbf{x}}^l\right)\right) + \mathbf{x}^{l-1}, \\
\mathbf{x}^l &= \text{MLP}\left(\text{LN}\left(\hat{\mathbf{x}}^l\right)\right) + \hat{\mathbf{x}}^l, \\
\tilde{\mathbf{x}}^{l+1} &= \text{RELU}(\text{BN}(DCONV(\mathbf{x}^l)), \\
\hat{\mathbf{x}}^{l+1} &= \text{SW-MSA}\left(\text{LN}\left(\tilde{\mathbf{x}}^{l+1}\right)\right) + \mathbf{x}^l, \\
\mathbf{x}^{l+1} &= \text{MLP}\left(\text{LN}\left(\hat{\mathbf{x}}^{l+1}\right)\right) + \hat{\mathbf{x}}^{l+1},
\end{aligned} \tag{3.3}
$$

where $\tilde{\mathbf{x}}^l$, $\hat{\mathbf{x}}^l$ and $\mathbf{x}^l$ indicate the output features of the RELU activation function, (S)W-MSA module and the MLP module, respectively; W-MSA and SW-MSA indicate Windows Multi-head Self-Attention and Shifted Windows Multi-Head Self-Attention, respectively.

The scale and computational complexity issues of high-resolution images are resolved by the Swin Transformer using a window-based hierarchy. The proposed backbone network fully exploits the ability to express global features, captures a wealth of contextual information and learns more discernible features.

3.2.3 Local Context Enhancement Swin Transformer-Based Network Structure

Fig 3.4 shows the network architecture designed with Local Context Enhancement Swin Transformer backbone. First, a patch partition layer is used to divide the input image with the size of HxWx3 into non-overlapping segments. Each patch has a size of 4x4, which is considered as a "token" and the feature dimension of the patch is 4x4x3. The feature of each patch is specified by concatenating the RGB values of the individual pixels. Then, by utilizing a linear embedding operation, the characteristics are converted to the desired dimension. To generate a hierarchical representation of the features, building on the proposed Local Context Enhancement Swin Transformer(LCEST) backbone, the hierarchical structure comprises four levels. The output resolution of "Stage 1", "Stage 2", "Stage 3" and "Stage 4" is $\frac{H}{4} \times \frac{H}{4}$, $\frac{H}{8} \times \frac{H}{8}$, $\frac{H}{16} \times \frac{H}{16}$ and $\frac{H}{32} \times \frac{H}{32}$, respectively.

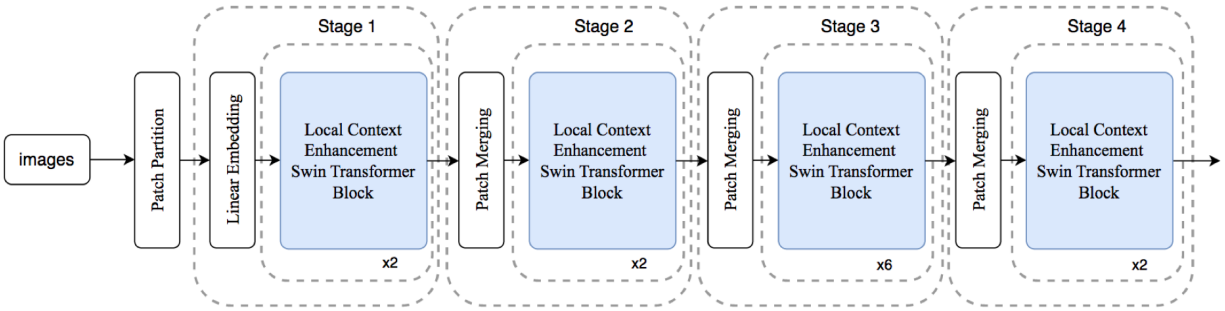


Figure 3.4: The overall architecture of Local Context Enhancement Swin Transformer.

3.3 RSU-FPN

3.3.1 U-block Architecture

We proposed a U-block which is created as a symmetrical U-shape encoder-decoder structure. It contains a number of 3x3 convolution layers placed in a series. A set of filters is applied to the input feature map in the first stage of the U-block to enhance the features with a varied number of channels. The filters are refined to recognize certain input characteristics, such as edges, corners, or textures. Multi-scale features are obtained by downsampling feature maps in a gradual manner and the characteristics are subsequently converted into feature maps of high resolution through a range of methods, such as progressive upsampling, concatenation, and convolution. The combination of convolution, batch normalization, activation, downsampling, and upsampling is used in the design of U-block to capture intricate interactions between the input and output.

A convolutional input layer takes the input feature map x, which has dimensions of H×W×C_{in}, and converts it to an intermediate map F1(x) with the number of channels equal to C_{out}. As demonstrated in Figure 3.5, the symmetric encoder-decoder structure is designed to take in the F1(x) feature map generated at an earlier stage of the network, the model is capable of extracting and encoding contextual information Φ(F1(x)) at multiple scales in a way that has been specifically trained. Φ refers to the symmetrical U-shape structure. Moreover, the local features and multi-scale features are fused together using a residual connection through summation, which is represented as F1(x) + Φ(F1(x)).

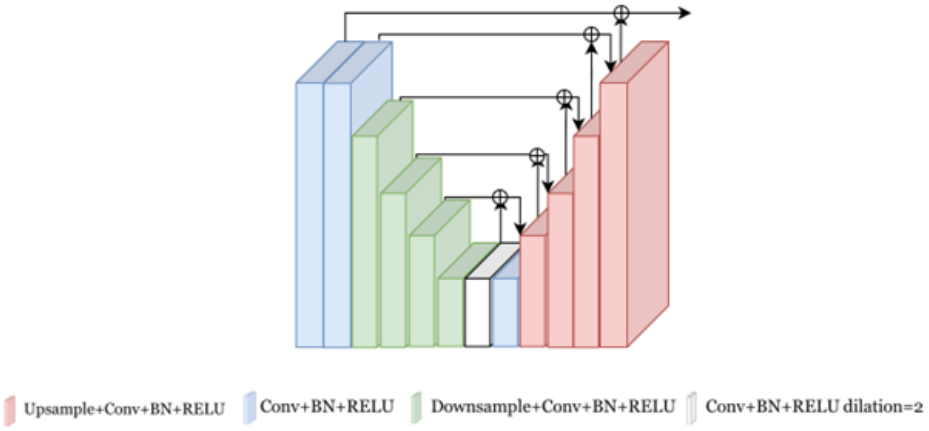


Figure 3.5: Architecture of U-block.

3.3.2 Residual U-FPN Network

In Figure 3.1, our study proposes the implementation of a Top-Down pathway to generate high-resolution features through the process of upsampling the maps of characteristics originating from higher pyramid stages. which are semantically more meaningful but have a lower spatial resolution. Additionally, we adopt U-block as the foundational architecture and use the notation {P2, P3, P4, P5} to represent the feature levels produced by our improved Swin Transformer backbone. Each feature map {P2, P3, P4, P5} passes through a U-block and generates a new feature map {N2, N3, N4, N5}, respectively. The spatial resolution is gradually reduced by a factor of down-sampling as we progress from P2 to P5. Furthermore, our framework deploys an enhancement of the Top-Down Path. The newly generated feature maps that correspond to {N2, N3, N4, N5} are represented by {F2, F3, F4, F5}. The U-block in Top-Down Path receives the input as the takes the concatenation of feature maps that have been upsampled from the corresponding level Ni. Noted that N5 is equivalent to F5, with no processing involved in F5.

3.4 Bounding Box Regression Loss Function

Since real objects are rarely precise rectangles, the bounding boxes of tiny objects often contain background pixels. The bulk of the foreground pixels in these bounding boxes is in the center, whereas most of the background pixels are concentrated at the edges. [64]. Additionally, the efficacy of anchor-based small object detectors is significantly dependent on the quality of the training samples that are chosen [65]. Unfortunately, given the restricted number of pixels available to effectively represent the object's attributes, selecting acceptable training examples becomes more difficult.

3.4.1 Normalized Gaussian Wasserstein Distance Loss

The Wasserstein distance can be employed to calculate the distance between two distributions. In the case of two normal distributions, $\mu_1 = \mathcal{N}(\boldsymbol{m}_1, \boldsymbol{\Sigma}_1)$ and $\mu_2 = \mathcal{N}(\boldsymbol{m}_2, \boldsymbol{\Sigma}_2)$, the following is the formula for the Wasserstein distance of the second order between these two distributions. [66]:

$$W_2^2(\mu_1, \mu_2) = \|\mathbf{m}_1 - \mathbf{m}_2\|_2^2 + \left\|\boldsymbol{\Sigma}_1^{1/2} - \boldsymbol{\Sigma}_2^{1/2}\right\|_F^2 \tag{3.4}$$

Moreover, if we consider Gaussian distributions N_a and N_b that are generated based on the bounding boxes A = (x_a, y_a, w_a, h_a) and B = (x_b, y_b, w_b, h_b), it is possible to simplify the equation as:

$$W_2^2(\mathcal{N}_a, \mathcal{N}_b) = \left\| \left(\left[x_a, y_a, \frac{w_a}{2}, \frac{h_a}{2} \right]^{\mathrm{T}}, \left[x_b, y_b, \frac{w_b}{2} \cdot \frac{h_b}{2} \right]^{\mathrm{T}} \right) \right\|_2^2 \tag{3.5}$$

By using the exponential form normalization, the Normalized Wasserstein Distance is calculated as:

$$NWD\left(\mathcal{N}_a, \mathcal{N}_b\right) = \exp\left(-\frac{\sqrt{W_2^2\left(\mathcal{N}_a, \mathcal{N}_b\right)}}{C} \right) \tag{3.6}$$

where C represents a custom fixed value. Therefore, the Normalized Gaussian Wasserstein Distance loss function is defined as:

$$\mathcal{L}_{NWD} = 1 - NWD(\mathcal{N}_a, \mathcal{N}_b) \tag{3.7}$$

The Gaussian distribution model for the prediction box A is denoted as N_a, while the one for the ground truth box B is denoted as N_b.

However, the Wasserstein distance lacks scale invariance and may not be the most appropriate option in situations where there is a wide range of object scales within the dataset.

3.4.2 L1 Loss

L1 Loss is a measure of errors between prediction and ground truth. The equation is shown below:

$$\mathcal{L}_{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{3.8}$$

Even if the error is minimal, the L1 loss function still modifies the model using a constant loss value, which causes the absolute value of the error derivative with respect to the predicted value to remain at 1 when the predicted value is only slightly different from the ground truth during the later stages of training. If the learning rate remains constant, it will oscillate around the steady value of the loss function, making it challenging to achieve higher accuracy through continued convergence.

3.4.3 Combined Regression Loss

$$\mathcal{L}_C = \mathcal{L}_{NWD} + \lambda \mathcal{L}_{MAE} \tag{3.9}$$

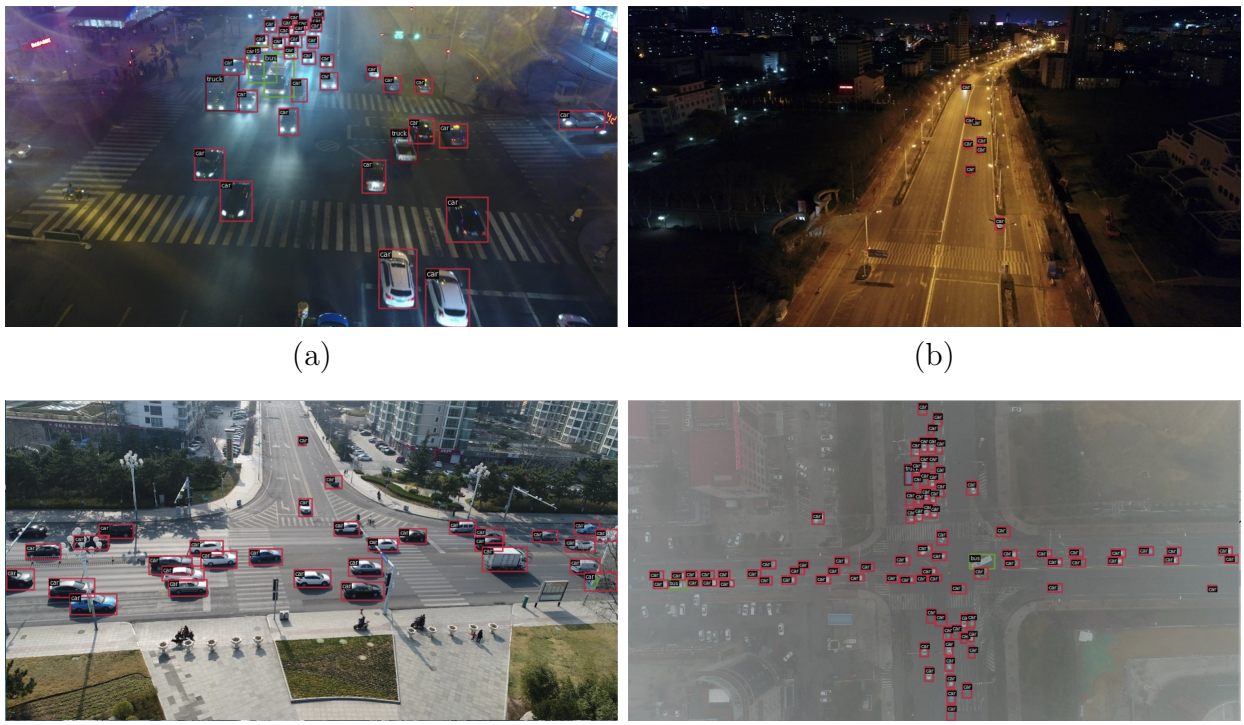
We define a combined loss function as a linear combination of the individual loss functions, where λ is the trade-off parameter. All experiments have their respective modal weights set to 5 to ensure consistency.

3.5 Experiments

3.5.1 Datasets

To evaluate the effectiveness of our proposed method, we perform our experiments on two mainstream datasets: UAVDT [25] Dataset and Vis-Drone2019 [48] Dataset.

UAVDT: The UAVDT dataset is a compilation of data collected by a UAV platform in several metropolitan locations and acts as a standard for UAVs. It covers a range of scenarios, including plazas, main roads, motorways, toll booths, and intersections. This dataset includes 40735 images. 60% of the dataset is randomly selected as the training set, the validation set comprises 10%, and the test set comprises 30%. The image has a resolution of 1080 $\times$ 540 pixels. The UAVDT dataset contains images with a more complicated background than the VisDrone dataset. The latter dataset is composed of three separate vehicle types that were evaluated in a total of fourteen unique situations. These scenarios differ in terms of the prevailing weather conditions, camera views, flight altitudes, and occlusions. Figure 3.6 shows examples of different scenarios in UAVDT dataset. Figure 3.7 shows the distribution of UAVDT dataset.



(a) (b)

(c) (d)

Figure 3.6: Annotated examples of different scenarios in UAVDT [25] dataset. (a) Night with poor visibility. (b) Nightlight condition. (c) Day with good visibility. (d) Smoggy weather.

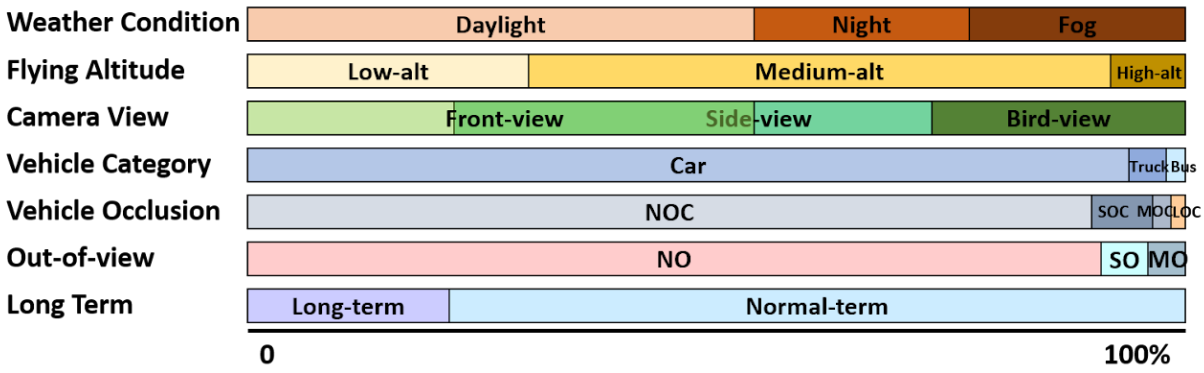
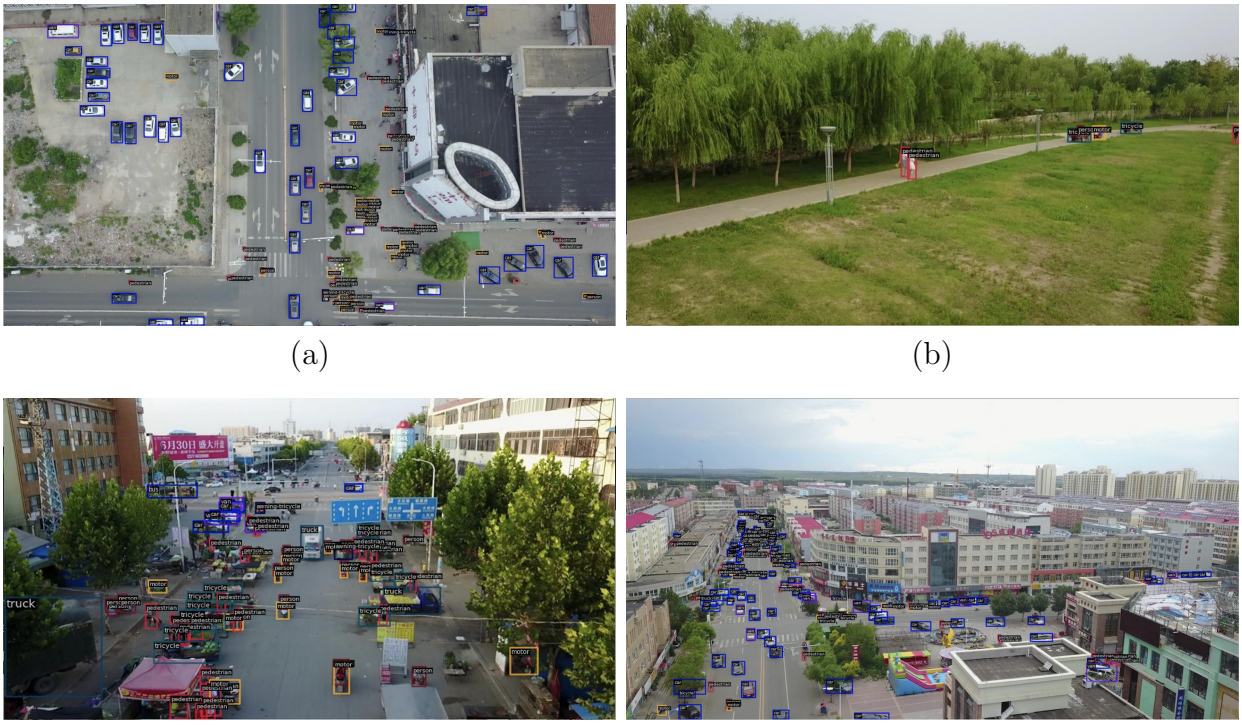


Figure 3.7: Distribution of UAVDT dataset. [25]

Vis-Drone2019: The dataset contains 10209 drone-captured images in a variety of locations and scenarios, including urban areas, forests, coastlines, and highways, with potentially significant differences in quality, scale, and perspective. The object detection dataset has been divided into 3 subsets, 6471 training images, 548 validation images, and 3190 testing images. There are ten accessible evaluation categories, all of which contain comprehensive annotations. The images have a resolution of approximately 2000 $\times$ 1500 pixels. The images were captured on days with clear skies and excellent visibility, as well as clouds and reduced visibility. In addition, the dataset comprises images collected at various times of day, spanning from early morning to late afternoon, which might change the lighting conditions of the scene. The VisDrone dataset is a tough benchmark for computer vision algorithms because of its variable weather and lighting conditions. Unfortunately, the testing phase cannot be completed since the evaluation server is no longer accessible. Thus, like earlier research, our method is assessed using the validation set. Table 3.1 demonstrates the initial row displays the range of pixel sizes for the objects, and the data in the second row presents information regarding the quantity of objects that pertain to the respective size category. Figure 3.8 demonstrates some examples of different scenarios in VisDrone dataset.

Table 3.1: Distribution of the bounding boxes in VisDrone train and validation set.

Size(pixel)	$<200^2$	$200^2 \sim 400^2$	$>400^2$	$<32^2$	$32^2 \sim 64^2$	$>96^2$
number	487887	2035	42	306262	159999	23703



(a) (b)

(c) (d)

Figure 3.8: Annotated examples of different scenarios in VisDrone dataset [26]. (a) Bird view of the road. (b) Forest. (c) Congested condition. (d) Specific perspective.

3.5.2 Implementation

We conducted our experiments based on the MMDetection [67] toolbox and Pytorch framework. To keep the training results consistent and comparable, we used identical hyperparameters for the training process. For the training phase, we used AdamW optimizer with a weight decay of 0.05 and the initial learning rate was set to 0.0001. Furthermore, we used standard data augmentation techniques for both the training and testing datasets, including a horizontal random flip with a 0.5 probability. All the networks in this thesis are trained for 20 epochs since the average precision (AP) shows little improvement after 15 epochs of training. All the experiments are processed on an NVIDIA A100 GPU with a memory of 48 GB.

3.5.3 Evaluation Metrics

To verify our proposed method, we utilize precision, recall, Average Precision, AP_{50}, AP_{75}, AP_s, AP_m, and AP_l as the evaluation metrics. The precision metric calculates the ratio of true positives out of all the positive detections. Recall refers to a statistic that determines the ratio of true positives made in comparison to the total number of real objects present in an image. The F_1-score is a statistical metric, calculated as the harmonic mean of the precision and recall scores, with 1 being the best possible score and 0 representing the worst possible score. Intersection over Union (IoU) is a metric used to measure the size of the intersection of two boxes divided by the size of their union. Average precision is the area under the precision–recall curve. AP represents the average AP value when the IoU is set at 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, and 0.95. AP_{75} and AP_{50} denote the AP values corresponding to IoU thresholds of 0.75 and 0.50, respectively.

The formulas to calculate these metrics are presented below:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{3.10}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{3.11}$$

$$\text{F}_1 = \frac{\text{2 x Precision x Recall}}{\text{Precision+Recall}} \tag{3.12}$$

where TP indicates a correctly categorized positive sample and FP indicates an erroneously classed positive sample. FN represents a negative sample that is wrongly classified, and TN represents a negative sample that is correctly classified.

Chapter 4

Results and Discussions

In this section, we present a comprehensive assessment of our suggested technique through quantitative evaluation and qualitative evaluation. Furthermore, we conduct an ablation study to explore the effect of LCEST backbone, the effect of RSU-FPN neck, and the effect of the combined loss we proposed.

4.1 Quantitative evaluation

We conduct a comparative analysis to assess the effectiveness of our proposed method compared with the SOTA object detection networks under the same settings, such as Faster R-CNN, Dynamic R-CNN, and Cascade R-CNN.

Table 4.1: Evaluation metrics (%) for different models on the UAVDT dataset.

Models	**AP**	**AP**$_{50}$	**AP**$_{75}$	**AP**$_s$	**AP**$_m$	**AP**$_l$	**Precision**	**Recall**	**F**$_1$
FRCNN(ResNet-50)	55.7	69.8	66.9	20.2	79.8	84.6	96.3	61.8	75.3
FRCNN(PVT2-B0)	60.8	71.5	69.8	25.6	84.7	91.6	97.8	70.4	81.9
Cascade-RCNN	61.5	73.5	71.7	30.7	83.8	89.3	98.0	71.5	82.7
Dynamic R-CNN	57.6	65.3	64.6	14.3	86.9	**92.7**	**98.5**	64.3	77.8
Proposed Method	**83.1**	**98.8**	**96.4**	**76.3**	**87.5**	91.8	98.1	**95.8**	**97.0**

The comparative results in Table 4.1 demonstrate that our method achieves AP of 83.1%, which is greater than Faster R-CNN with the ResNet-50 [68] backbone, Faster

R-CNN with the PVT2-B0 backbone, Cascade R-CNN, and Dynamic R-CNN by 27.4%, 22.3%, 21.6%, and 25.5%, respectively. Furthermore, the proposed method achieves 98.8% AP_{50} and 96.4% AP_{75}. Our proposed method has the best performance among these five models in the evaluation metrics except AP_l, which is defined as the AP of test results with object frame sizes larger than 96 pixels. Since our study primarily focuses on the accuracy of tiny items in UAV images, the accuracy of larger objects may be neglected. The proposed approach is particularly noteworthy as it shows significant enhancement in detecting objects of small size. In comparison to the Faster R-CNN that uses the ResNet-50 as the backbone network, this new method achieves an improvement of 56.1%, 7.7%, and 7.2% AP for small, medium, and large object detection, respectively. In addition, the proposed method improved on both AP_s and AP_m. Cascade R-CNN can be regarded as an expanded version of Mask R-CNN that operates in multiple stages and shows an improvement of 5.8% on AP compared with Faster R-CNN. We get the unsatisfactory performance of dynamic R-CNN models in detecting small targets, with 57.6% AP, lower than Cascade R-CNN by 3.9%. Moreover, our method achieves the highest recall and F_1-score, with 95.8% and 97.0%, respectively. The utilization of our technique additionally empowers the model to identify more objects and achieve improved recall metrics.

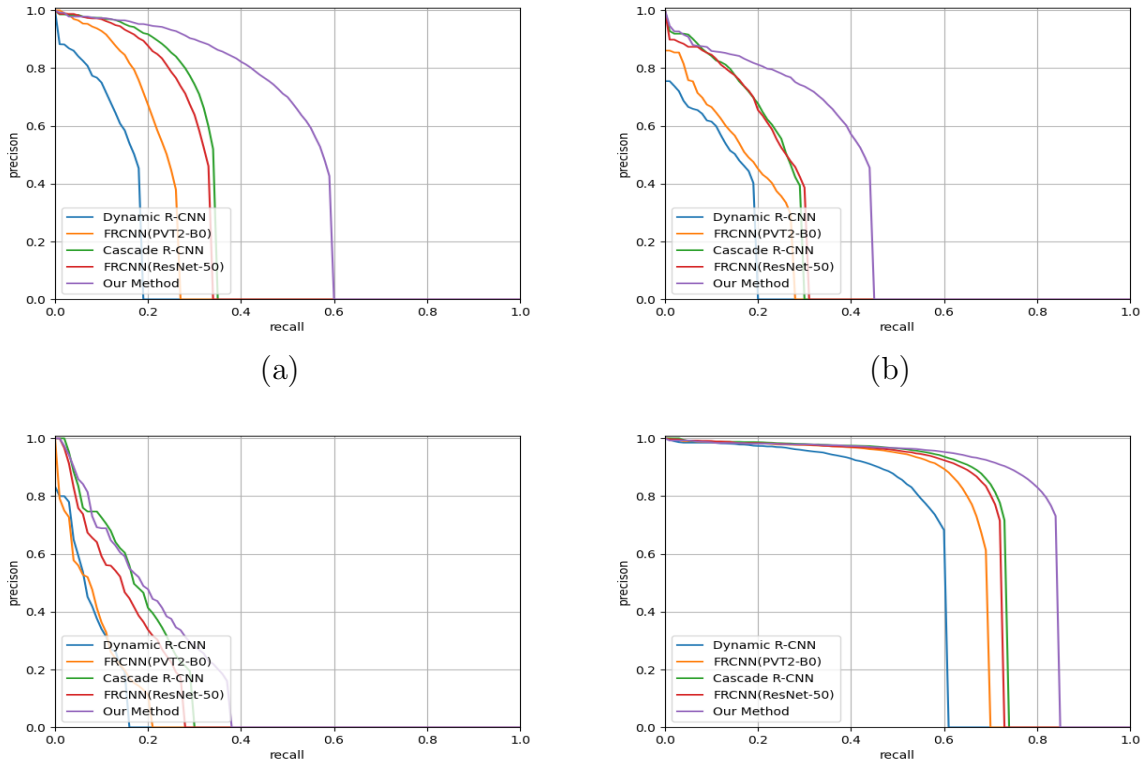
Table 4.2: Evaluation metrics (%) for different models on the Vis-Drone dataset.

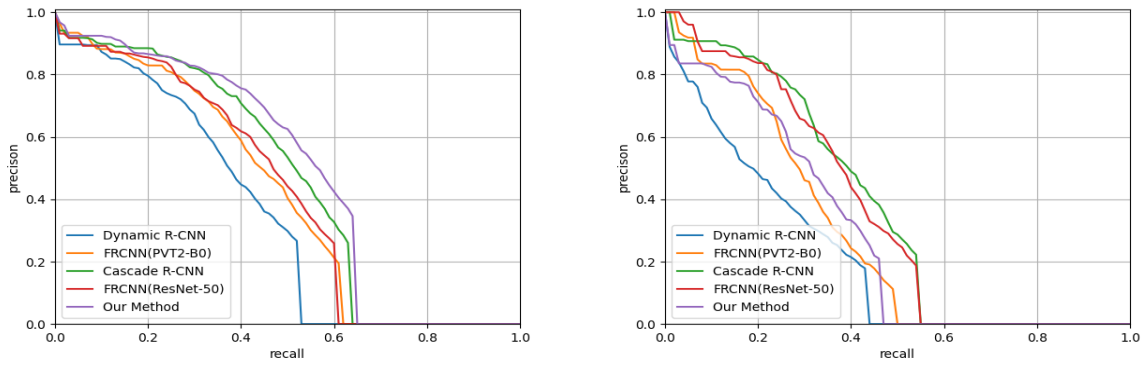
Models	**AP**	**AP_{50}**	**AP_{75}**	**AP_s**	**AP_m**	**AP_l**	**Precision**	**Recall**	**F_1**
FRCNN(ResNet-50)	22.4	37.2	23.7	11.0	36.4	40.9	52.1	29.7	37.8
FRCNN(PVT2-B0)	19.4	31.8	20.8	9.1	32.3	37.1	47.2	25.2	32.9
Cascade-RCNN	25.2	40.1	26.8	12.5	**40.3**	**47.7**	**56.0**	31.9	40.6
Dynamic R-CNN	13.7	24.7	13.5	6.4	25.2	15.0	55.4	18.2	27.4
Proposed Method	**26.4**	**45.5**	**27.0**	**18.0**	36.5	39.4	50.1	**36.8**	**42.4**

Table 4.2 illustrates the comparison results between different methods combined with the respective backbones on the Vis-Drone2019 dataset. The proposed method achieves superior performance with the highest AP and demonstrates a notable increase in the precision of object detection, exceeding that of vanilla Faster R-CNN by 3.5% AP. Both Dynamic R-CNN and Faster R-CNN with PVT2-B0 backbone demonstrate worse performance in UAV image object detection. The values of AP_{50}, AP_{75}, and AP_s are improved from Faster R-CNN to our proposed method, with an increase of 8.3%, 3.3%, and 7.0%, respectively. Furthermore, the improvement primarily results from the detection of smaller-sized objects. The increase in AP_s by 7.0% is particularly remarkable, which means that

our method achieved significant results in detecting small objects, effectively improving the precision of detection. The enhancement of AP_m and AP_l is not significant, the possible reason is that the smallest resolution feature map from the backbone goes through only one U-block and is not connected by a shortcut between the upper and lower layers of feature maps. It is notable that our method achieves the highest recall and F_1-score which can be regarded as a significant improvement in the ability to detect small objects compared to other models.

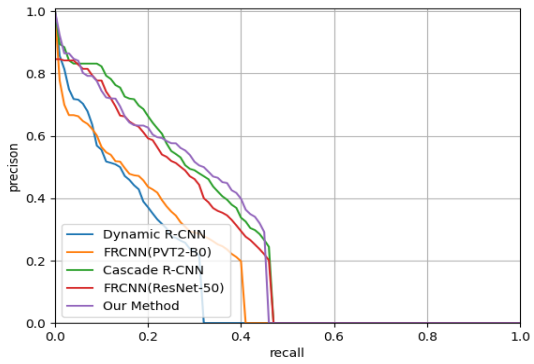
In addition, we assess the precision and recall for each class, and the PR curves are depicted in Figure 4.1. When comparing models at the same recall level, the model with higher precision has better performance. Our model outperforms in most categories and aids in achieving greater precision for identical recall levels. However, for the truck and awning-tricycle categories, our model does not perform as well as other methods. One of the potential reasons we speculate is that Transformer-based models need more training data. The quantity of these classifications is exceedingly limited within the confines of the dataset.



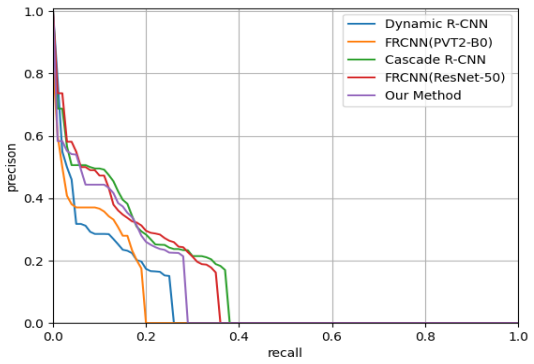


(e)

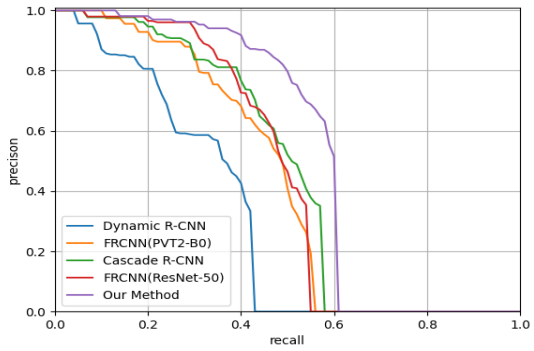
(f)



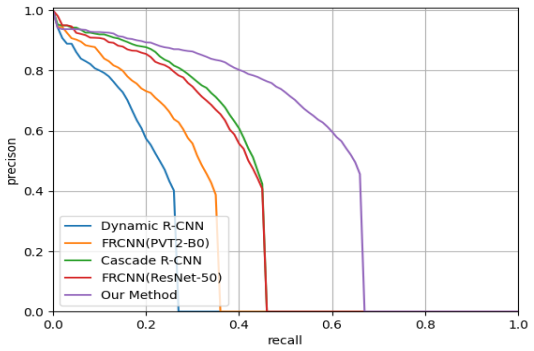
(g)



(h)



(i)



(j)

Figure 4.1: The precision-recall curves exhibit superior performance in each category on the VisDrone dataset. (a) pedestrian. (b) people. (c) bicycle. (d) car. (e) van. (f) trucks (g) tricycle. (h) awning-tricycle. (i) bus. (j) motors.

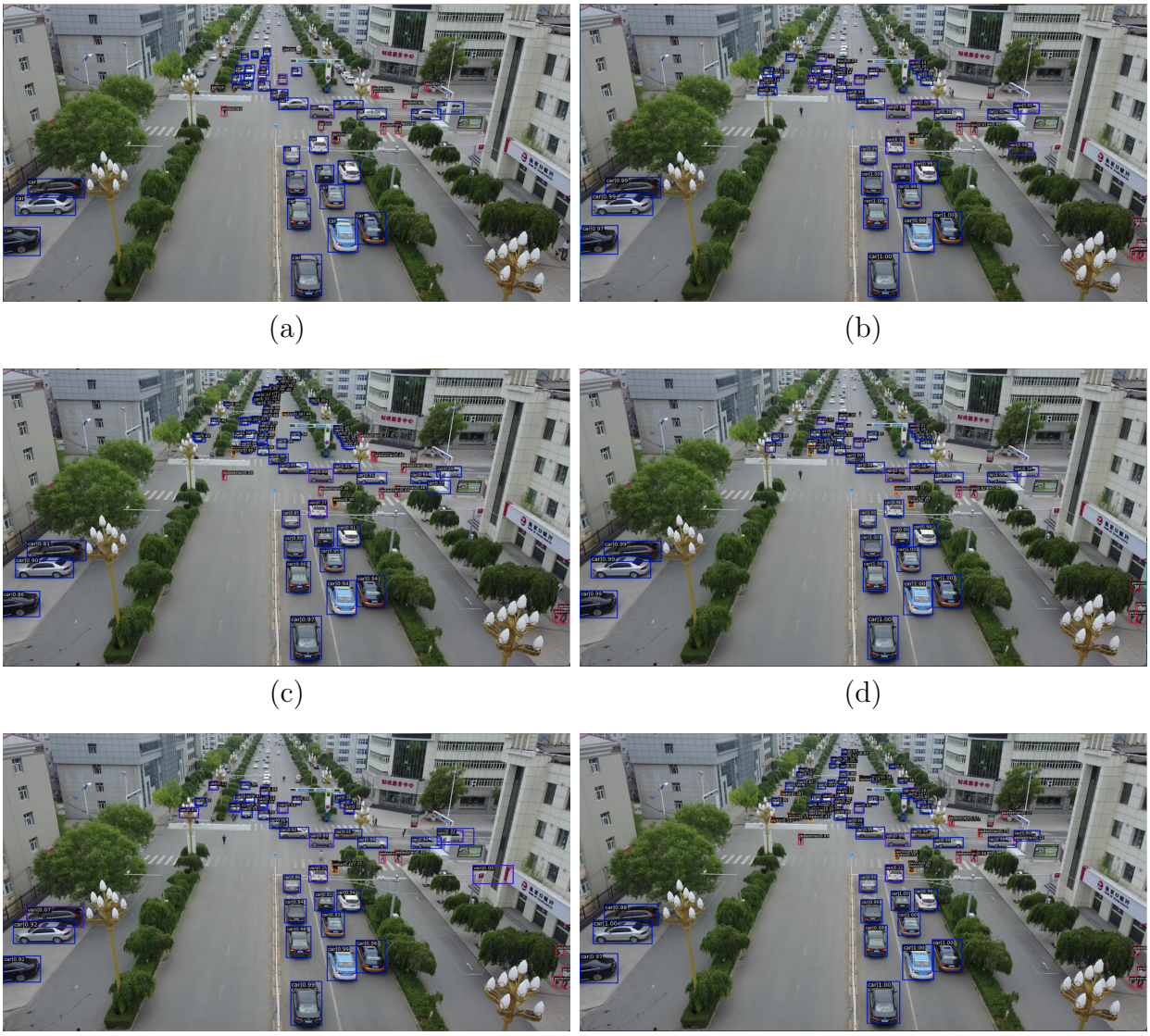
4.2 Qualitative evaluation

The visualization of the comparison experiment performed on the VisDrone dataset is shown in Figure 4.2. Our technique has shown to be very successful in highly populated areas when objects like vehicles are tightly grouped together. Furthermore, our model also detects several small-scale vehicles at the far end of the image, which are not included in the ground truth annotations. The accuracy of model training and experiment evaluation can be compromised by incorrect labeling of the ground truths.

For the UAVDT dataset, the results of our technique and four other SOTA models are depicted in Figure 4.3. These models are evaluated under crowded road circumstances. When it comes to accurately detecting automobiles in very congested junction road circumstances, our suggested technique is more accurate than the other common methods. In certain locations when automobiles of a similar colour are parked near one another, our approach is nevertheless able to accurately identify those vehicles. Some methodologies, such as the Faster R-CNN, exclude the cars from these zones. Since our technique correctly identifies most of the vehicles seen in the UAV picture, we can conclude that it is very effective at locating cars within the images captured by the UAV.

Moreover, as the visualization of detection results illustrated in Figure 4.4, our proposed method still achieves relatively good performance and determines the location and category of the vehicles even in hazy weather. On the other hand, we have also observed that our method occasionally makes errors in identifying vehicles in the images. Some areas covered in fog are mistaken as vehicles. However, the other four models have a limited ability to recognize small objects in this extreme weather. These models appear to struggle with identifying targets in foggy environments. Most of them are influenced by weather conditions and have inadequate learning ability for small targets, thereby misjudging the position information of small targets.

As illustrated in Figure 4.5, despite the effects of nightlight, our method still achieves superior performance. Due to the lighting conditions, the color of certain vehicles appears to merge with that of the road surface, making it difficult to distinguish them from their surroundings. Furthermore, for CNN-based models, these vehicles are likely to be blurred out during the convolution process, resulting in missing features and thus affecting the accuracy of the model. Compared to other CNN models, our model still has high recognition accuracy for smaller vehicles under nightlight conditions. Even in night light conditions, our model continues to learn the characteristics of small targets and locates them in the image.

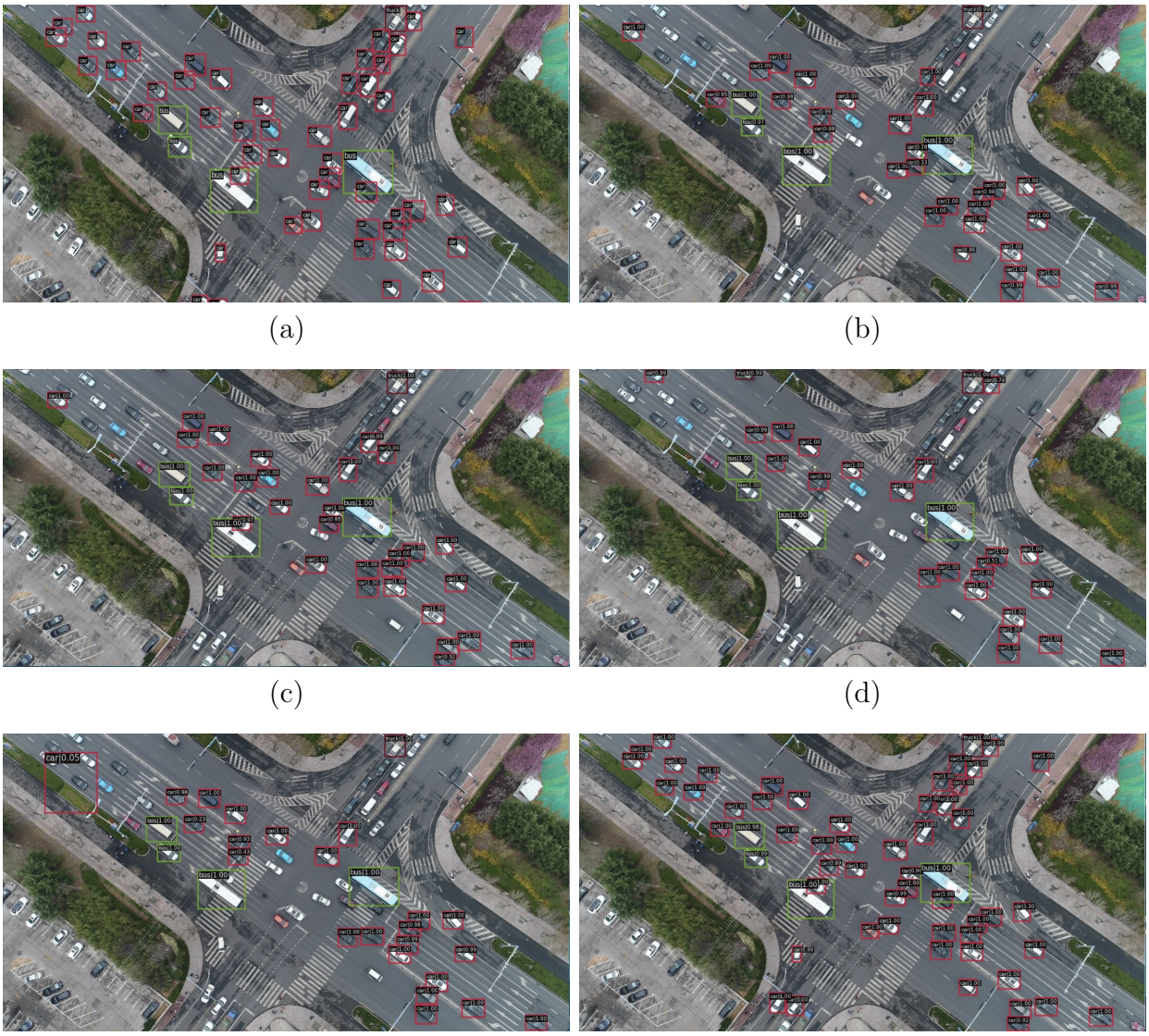


(a) (b)

(c) (d)

(e) (f)

Figure 4.2: Detection results of different methods on VisDrone dataset. The rectangles indicate the bounding boxes that have been drawn around each detected object, and the categories and confidence scores are marked within the bounding box. (a) Ground Truth. (b) Faster R-CNN(ResNet-50). (c) Faster R-CNN(PVT2-B0). (d) Cascade R-CNN. (e) Dynamic R-CNN. (f) Our Method.



(a) (b)

(c) (d)

(e) (f)

Figure 4.3: Detection results of different methods under congested intersection road conditions. The rectangles indicate the bounding boxes that have been drawn around each detected vehicle, and the categories and confidence scores are marked within the bounding box. (a) Ground Truth. (b) Faster R-CNN(ResNet-50). (c) Faster R-CNN(PVT2-B0). (d) Cascade R-CNN. (e) Dynamic R-CNN. (f) Our Method.

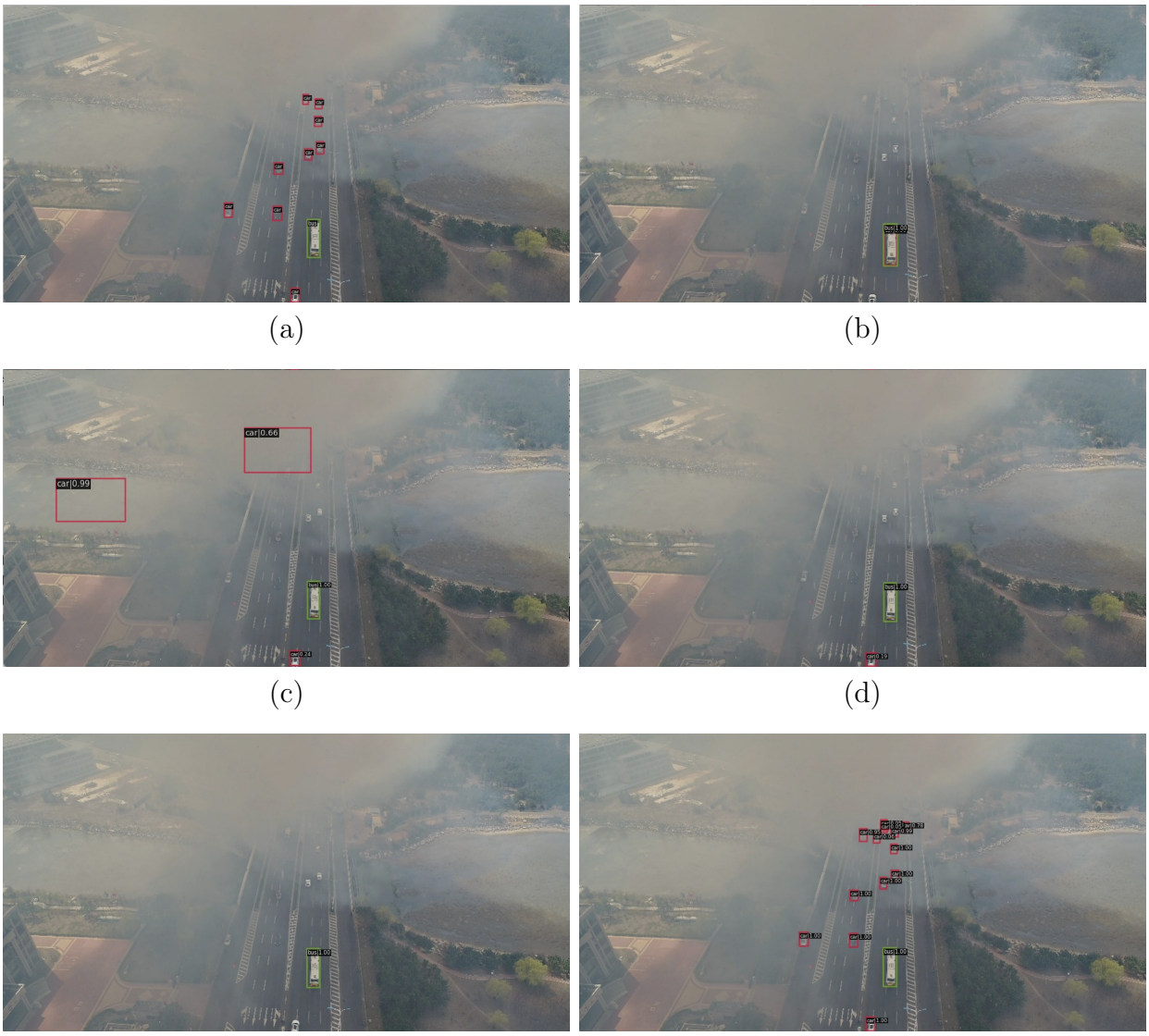
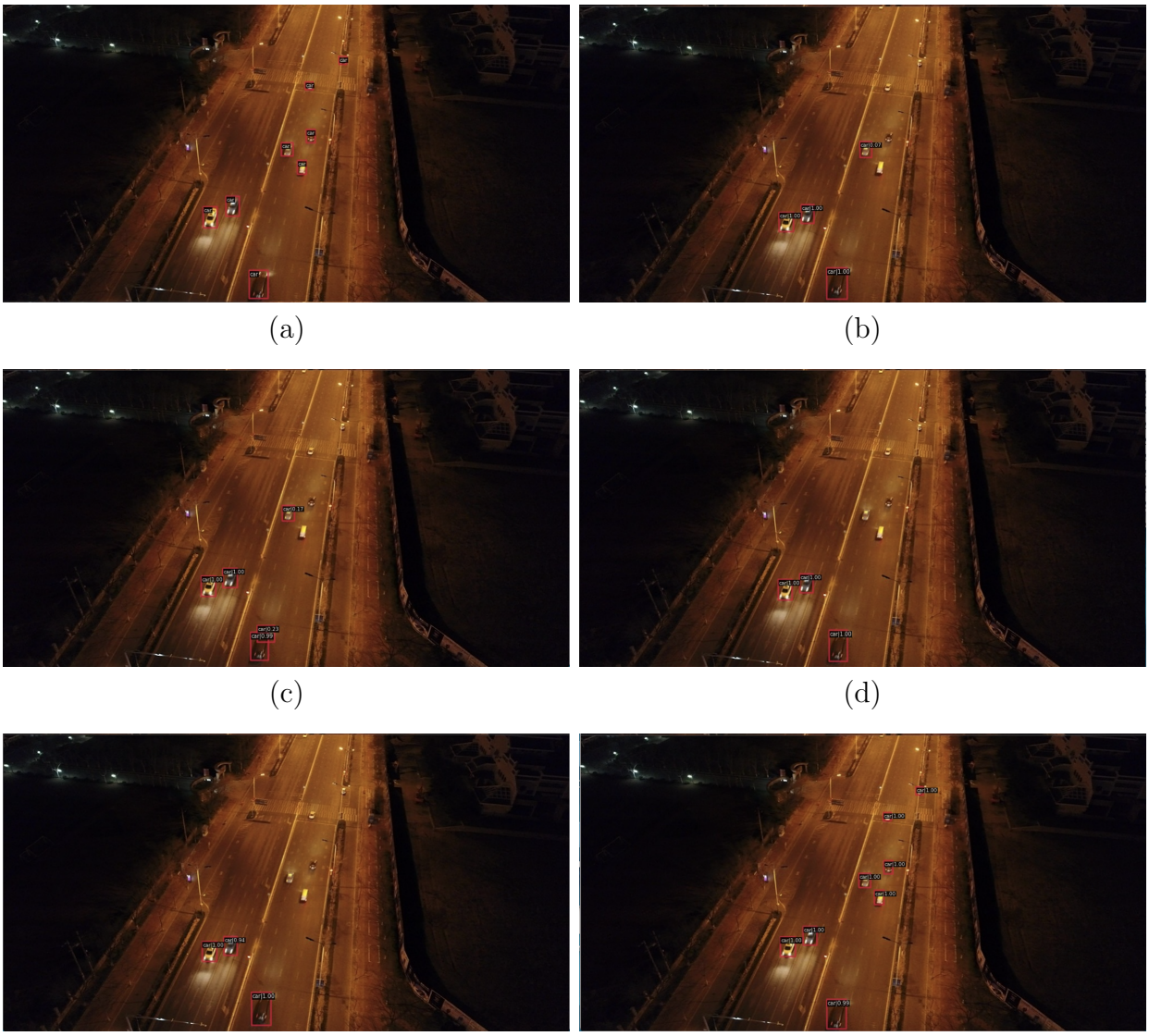


Figure 4.4: Detection results of different methods in smoggy weather conditions. The rectangles indicate the bounding boxes that have been drawn around each detected vehicle, and the categories and confidence scores are marked within the bounding box. (a) Ground Truth. (b) Faster R-CNN(ResNet-50). (c) Faster R-CNN(PVT2-B0). (d) Cascade R-CNN. (e) Dynamic R-CNN. (f) Our Method.



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

Figure 4.5: Detection results of different methods in the nighttime scenarios. The rectangles indicate the bounding boxes that have been drawn around each detected vehicle, and the categories and confidence scores are marked within the bounding box. (a) Ground Truth. (b) Faster R-CNN(ResNet-50). (c) Faster R-CNN(PVT2-B0). (d) Cascade R-CNN. (e) Dynamic R-CNN. (f) Our Method.

4.3 Ablation study

This section outlines an ablation study that is conducted to determine the extent that each component contributed to the accuracy of the entirety specifically on UAVDT dataset and VisDrone dataset. We perform a series of ablation experiments on the baseline to explore the significance of different modules by replacing the LCEST backbone, RSU-FPN and combined loss, respectively. We analyze three configurations that may have an impact on ultimate performance. Table 4.3 illustrates the gradual addition of modules at each level to the baseline to demonstrate their compatibility on UAVDT dataset. The baseline is ResNet-50 + FPN, LCEST stands for Local Context Enhancement Swin Transformer backbone, RSU-FPN stands for Residual U-FPN neck, the loss function combined Normalized Gaussian Wasserstein Distance and L1 loss is denoted as LC. "$\surd$" means the module is added. "$-$" means the module is not added. The metrics of the baseline are demonstrated in the first row of the table. AP/AP_{50} rises from 75.1%/94.2% in the second row to 83.1%/98.8% in the final row. The ablation experiments on the VisDrone dataset are displayed in Table 4.4.

Table 4.3: Evaluation metrics (%) for ablation study on UAVDT dataset.

Methods	LCEST	RSU-FPN	LC	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Baseline	–	–	–	55.7	69.8	66.9	20.2	79.8	84.6
Baseline	$\surd$	–	–	75.1	94.2	90.4	63.5	82.6	86.3
Baseline	$\surd$	$\surd$	–	75.6	94.5	91.0	64.3	82.9	85.8
Ours	$\surd$	$\surd$	$\surd$	**83.1**	**98.8**	**96.4**	**76.3**	**87.5**	**91.8**

Table 4.4: Evaluation metrics (%) for ablation study on VisDrone dataset.

Methods	LCEST	RSU-FPN	LC	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Baseline	–	–	–	22.4	37.2	23.7	11.0	36.4	40.9
Baseline	$\surd$	–	–	24.5	41.0	25.1	13.0	**38.1**	**47.0**
Baseline	$\surd$	$\surd$	–	25.9	44.5	26.9	17.0	36.6	24.0
Ours	$\surd$	$\surd$	$\surd$	**26.4**	**45.5**	**27.0**	**18.0**	36.5	39.4

4.3.1 Effect of LCEST backbone

To verify the effectiveness of our Local Context Enhancement Swin Transformer backbone on UAVDT dataset, we select Faster R-CNN(ResNet-50) + FPN as the baseline. An improvement is achieved by substituting the original ResNet + FPN backbone with LCEST + FPN. As shown in Table 4.3, from ResNet-50 to LCEST, all the scores increased significantly. LCEST is effective to improve the evaluation metrics. AP is increased by 14.9%. The AP_{50}, AP_{75}, AP_s, AP_m, and AP_l values are all significantly higher than the baseline values of 24.4%, 20.6%, 43.3%, 2.8%, and 1.7%, respectively. Figure 4.6(a) and (b) show the comparison on a night with poor visibility. The baseline can distinguish only the larger bus, while the smaller vehicles are completely ignored. It fails to differentiate automobiles from the background. However, LCEST captures the character of small vehicles, improving the model's capability to recognize diminutive entities. It suggests that the LCEST backbone may enhance the capacity to locate targets and can effectively use the detector's capabilities to improve the precision of object detection.

Table 4.4 illustrates that metrics' values exhibit a remarkable enhancement on VisDrone dataset. The advancement in the metrics from baseline to LCEST, is attributed to the integration of an improved Swin Transformer backbone. As compared to typical CNN models, the employment of the LCEST backbone improves the metrics of numerous experimental results. The AP increases from 22.4% to 24.5%, AP_{50} contributes more, from 37.2% to 41.0%. These experimental results demonstrate the efficacy of our methodology. Figure 4.7(b) shows that a number of densely clustered vehicles on a congested road are recognized accurately. However, the baseline is not discerning for dense targets, resulting in many missing objects. Limited by a large number of overlapping regions, CNN-based model is struggling to extract these features. The introduction of LCEST facilitates the detection of densely distributed objects. We explain it as our proposed backbone can extract global information and boost the model's ability to perceive local details.

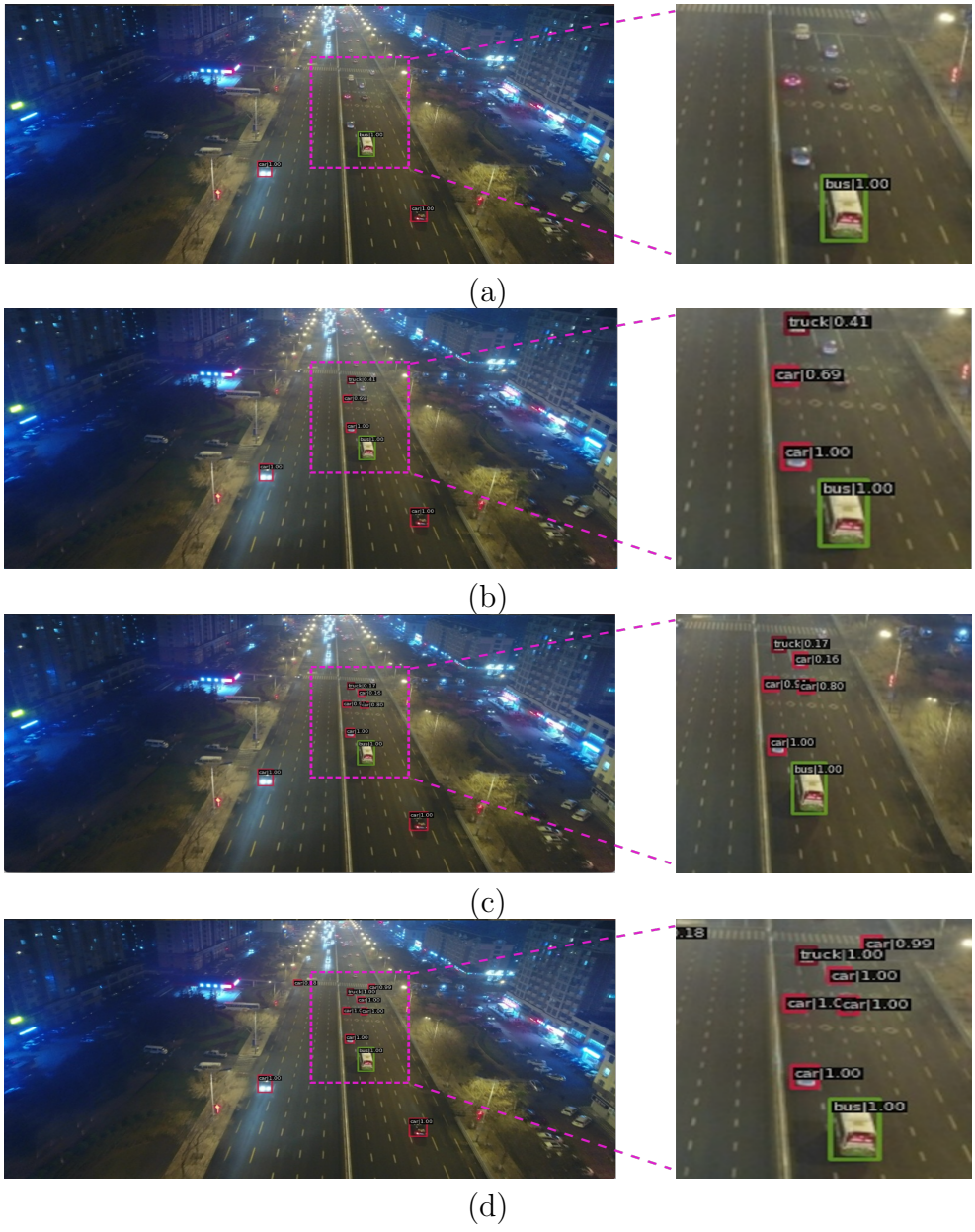


Figure 4.6: The visualization of detection results on the UAVDT dataset by progressively incorporating LCEST backbone, RSU-FPN, and Combined loss to the baseline. (a) Baseline. (b) Baseline + LCEST. (c) Baseline + LCEST + RSU-FPN. (d) Baseline + LCEST + RSU-FPN + Combined loss.

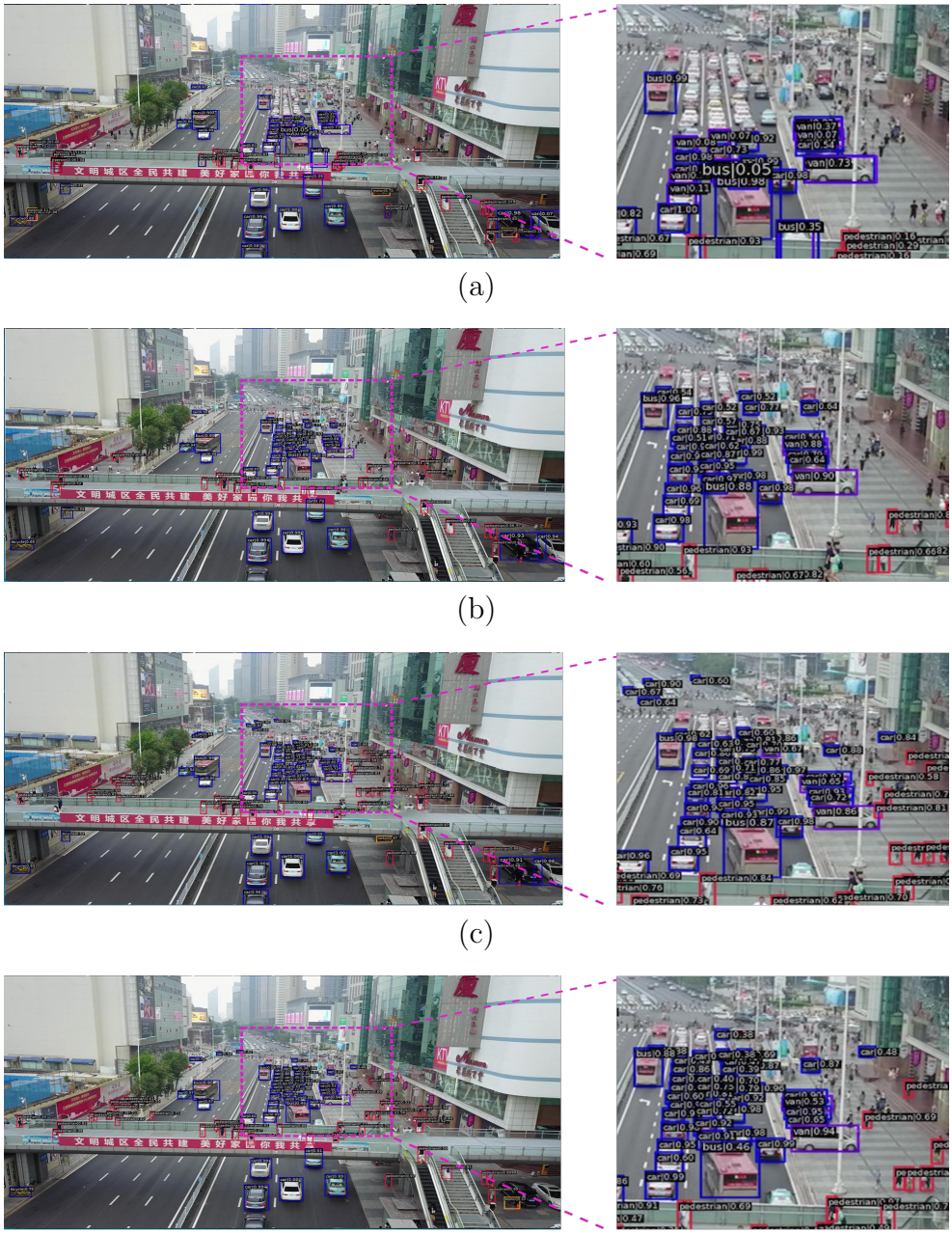


Figure 4.7: The visualization of detection results on the VisDrone dataset by progressively incorporating LCEST backbone, RSU-FPN, and Combined loss to the baseline. (a) Baseline. (b) Baseline + LCEST. (c) Baseline + LCEST + RSU-FPN. (d) Baseline + LCEST + RSU-FPN + Combined loss.

4.3.2 Effect of RSU-FPN

To verify that the RSU-FPN neck can enhance the fusion of features, we replace the original FPN [19] with our proposed RSU-FPN. As illustrated in Table 4.3, the AP of adding LCEST and RSU-FPN in the baseline is 0.5% higher than the baseline with LCEST. The improvement is observed consistently across small and medium scales. The application of RSU-FPN has slightly improved the AP_s by 0.8%. Furthermore, AP_{50}, AP_{75}, and AP_m values increase by 0.3%, 0.6%, and 0.3%, respectively, indicating that the proposed fusion module combines high-level features and low-level features effectively within each stage. Since the input of the RSU-FPN module is derived from the extracted features of the LCEST backbone, RSU-FPN is only used to integrate the characteristics. The insignificant improvements of the metrics still demonstrate a higher capability for feature fusion than the original FPN. In Figure 4.6(c), further improvement in vehicle identification accuracy is observed. Vehicles with a body colour like that of the road are also detected. The visualization result is still convincing to support the efficacy of RSU-FPN.

Table 4.4 depicts the increase of 1.4% AP, 3.5% AP_{50}, 1.8% AP_{75}, and 4.0% AP_s, which adequately demonstrates the advantages of the RSU-FPN. Furthermore, Figure 4.7 shows a discernible enhancement in the detection results through the utilization of RSU-FPN. In contrast to the detection results in Figure 4.7(b), we observe that most of the vehicles present in these images are precisely detected. In addition, more pedestrians on the right sidewalk are identified and several obscure vehicles at the end of the road are also marked out. Pedestrians in the image have the characteristic of small scale. Under blurred visual conditions, RSU-FPN still fuses the features of small objects and locates the bounding boxes and assists in recognizing overlooked things which are considered as background. Therefore, the experimental results demonstrate the exceptional value of our innovative RSU-FPN module, which has proven to be an effective module for fusing complex features from tiny objects.

4.3.3 Effect of Combined Regression Loss

We evaluate the effect of the L1 loss and our proposed combined loss on the model. Table 4.3 demonstrates that all scores have grown substantially. The implementation of the Combined loss has resulted in an enhancement of the box AP, elevating it from 75.6% to 83.1%. More specifically, the improvements in performance are observed in AP_s, with a substantial increase of 8.0 points in AP when compared to the corresponding baseline. In Figure 4.6(d), more small-scale vehicles are detected at night. The experimental results validate the positive impact of introducing combined loss.

The results in Table 4.4 shows an improvement of AP, AP_{50}, AP_{75}, and AP_s. The test results in Figure 4.7(d) are more consistent with the ground truth. As illustrated in Figure 4.7(c), we observe that some vehicles at the end of the road are marked out, but they are not annotated by ground truth. Mislabeled ground truth has a direct effect on the process of continuous model training and experiment assessment.

As a result, the experimental results highlight the importance of each configuration in enhancing performance. The effectiveness of our method is confirmed by the significant performance improvement achieved through each individual configuration, as well as their combination.

Chapter 5

Conclusions

5.1 Summary

In conclusion, object detection from UAV imagery is an essential technology with a wide range of applications in numerous fields. Efficient and precise object detection in UAV photography provides useful insights and data for decision-making processes in various domains, making it a vital method for a variety of industries. Current conventional methods and general deep learning-based methods do not yield adequate precision for high-precision applications. In this thesis, we propose a new approach for the automated detection of objects in images captured by unmanned aerial vehicles, named Local Context Enhancement Swin Transformer. Specifically, to amalgamate the strengths of Convolutional Neural Network (CNN) and Transformer, a Local Context Enhancement Module is proposed to be used in combination with Swin Transformer. In addition, an RSU-FPN neck is introduced to integrate intra-stage multi-scale features. The combination of L1 loss and normalized Gaussian Wasserstein distance loss can mitigate the problem of being impacted by small object location deviations. Furthermore, we conduct a comparative study and ablation study on both UAVDT dataset and VisDrone dataset to evaluate the performance of our proposed model. The experimental results demonstrate that our proposed method can increase the accuracy of detection for small-scale objects and achieve SOTA performance on UAVDT dataset and Vis-Drone dataset, with the AP of 83.1% and 26.4%, respectively.

5.2 Future Work

Limitations are also observed in experimental experiments. Each epoch of the training procedure takes a significant amount of time, and the training process demands high hardware resources. The proposed method may be improved further, and future efforts can concentrate on creating lightweight models and increasing performance, which could be possible by putting effective transformers in place and using advanced optimization methods. Furthermore, we may evaluate our proposed model on the newly published dataset, such as SODA-D dataset [69] and Manipal-UAV dataset [70].

References

[1] A. Ramachandran and A. K. Sangaiah, "A review on object detection in unmanned aerial vehicle surveillance," *International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 215–228, 2021.

[2] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging uavs for disaster management," *IEEE Pervasive Computing*, vol. 16, no. 1, pp. 24–32, 2017.

[3] C. Zhang, P. M. Atkinson, C. George, Z. Wen, M. Diazgranados, and F. Gerard, "Identifying and mapping individual plants in a highly diverse high-elevation ecosystem using uav imagery and deep learning," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 169, pp. 280–291, 2020.

[4] 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, 2005, pp. 886–893 vol. 1.

[5] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157 vol.2.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016.

[7] 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, 2017.

[8] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6154–6162.

[9] H. Zhang, H. Chang, B. Ma, N. Wang, and X. Chen, "Dynamic r-cnn: Towards high quality object detection via dynamic training," in *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV*, Berlin, Heidelberg, 2020, p. 260–275.

[10] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 105–114.

[11] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1132–1140.

[12] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2472–2481.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[14] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., 2020.

[15] X. Dai, Y. Chen, J. Yang, P. Zhang, L. Yuan, and L. Zhang, "Dynamic detr: End-to-end object detection with dynamic attention," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 2968–2977.

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7*, 2021.

[17] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9992–10 002.

[18] Y. Yuan, Y. Luo, L. Kang, J. Ni, and Q. Zhang, "Range alignment in isar imaging based on deep recurrent neural network," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[19] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.

[20] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768.

[21] P. Zhou, B. Ni, C. Geng, J. Hu, and Y. Xu, "Scale-transferrable object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 528–537.

[22] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2det: A single-shot object detector based on multi-level feature pyramid network," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9259–9266, 07 2019.

[23] M. Tan, R. Pang, and Q. Le, "Efficientdet: Scalable and efficient object detection," 06 2020, pp. 10 778–10 787.

[24] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "Nas-fpn: Learning scalable feature pyramid architecture for object detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7029–7038.

[25] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, and Q. Tian, "The unmanned aerial vehicle benchmark: Object detection and tracking," in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X*, ser. Lecture Notes in Computer Science, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11214, 2018, pp. 375–391.

[26] D. Du etal, "Visdrone-det2019: The vision meets drone object detection in image challenge results," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 213–226.

[27] 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, 2002, pp. I–I.

[28] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds.

[29] 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, 2001, pp. I–I.

[30] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3, 2006, pp. 850–855.

[31] F. Yang, H. Fan, P. Chu, E. Blasch, and H. Ling, "Clustered object detection in aerial images," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 8310–8319.

[32] J. O. D. Terrail and F. Jurie, "On the use of deep neural networks for the detection of small vehicles in ortho-images," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 4212–4216.

[33] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery : A small target detection benchmark," *Journal of Visual Communication and Image Representation*, vol. 34, pp. 187–203, 2016.

[34] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386–397, 2020.

[35] S. Vemula and M. Frye, "Mask r-cnn powerline detector: A deep learning approach with applications to a uav," in *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, 2020, pp. 1–6.

[36] H. Huang, L. Li, and H. Ma, "An improved cascade r-cnn-based target detection algorithm for uav aerial images," in *2022 7th International Conference on Image, Vision and Computing (ICIVC)*, 2022, pp. 232–237.

[37] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[38] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517–6525.

[39] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, pp. 98–136, 2015.

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

[41] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds.

[42] A. Soleimani and N. M. Nasrabadi, "Convolutional neural networks for aerial multi-label pedestrian detection," in *2018 21st International Conference on Information Fusion (FUSION)*, 2018, pp. 1005–1010.

[43] X. Lu, J. Ji, Z. Xing, and Q. Miao, "Attention and feature fusion ssd for remote sensing object detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2021.

[44] G. Cheng, J. Han, P. Zhou, and L. Guo, "Multi-class geospatial object detection and geographic image classification based on collection of part detectors," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 98, pp. 119–132, 2014.

[45] Z. Li, S. Sun, Y. Li, B. Sun, K. Tian, L. Qiao, and X. Lu, "Aerial image object detection method based on adaptive clusdet network," in *2021 IEEE 21st International Conference on Communication Technology (ICCT)*, 2021, pp. 1091–1096.

[46] C. Li, T. Yang, S. Zhu, C. Chen, and S. Guan, "Density map guided object detection in aerial images," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 737–746.

[47] Y. Wang, Y. Yang, and X. Zhao, "Object detection using clustering algorithm adaptive searching regions in aerial images," in *Computer Vision – ECCV 2020 Workshops*, A. Bartoli and A. Fusiello, Eds., 2020.

[48] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu, "Vision meets drones: A challenge, 2018."

[49] 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 *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021, pp. 2778–2788.

[50] X. Liang, J. Zhang, L. Zhuo, Y. Li, and Q. Tian, "Small object detection in unmanned aerial vehicle images using feature fusion and scaling-based single shot detector with spatial context analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1758–1770, 2020.

[51] Z. Liu, G. Gao, L. Sun, and Z. Fang, "Hrdnet: High-resolution detection network for small objects," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, 2021, pp. 1–6.

[52] Y. Liu, F. Yang, and P. Hu, "Small-object detection in uav-captured images via multi-branch parallel feature pyramid networks," *IEEE Access*, vol. 8, pp. 145 740–145 750, 2020.

[53] N. E. Ocer, G. Kaplan, F. Erdem, D. K. Matci, and U. Avdan, "Tree extraction from multi-scale uav images using mask r-cnn with fpn," *Remote Sensing Letters*, vol. 11, no. 9, pp. 847–856, 2020.

[54] 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*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[55] B. M. Albaba and S. Ozer, "Synet: An ensemble network for object detection in uav images," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 10 227–10 234.

[56] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *ArXiv*, vol. abs/2004.10934, 2020.

[57] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[58] H. Inoue, "Data augmentation by pairing samples for images classification," 2018.

[59] D. Liang, F. Yang, T. Zhang, and P. Yang, "Understanding mixup training methods," *IEEE Access*, vol. PP, pp. 1–1, 10 2018.

[60] C. Chen, Y. Zhang, Q. Lv, S. Wei, X. Wang, X. Sun, and J. Dong, "Rrnet: A hybrid detector for object detection in drone-captured images," *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 100–108, 2019.

[61] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *International Conference on Machine Learning*, 2018.

[62] S. Venkataramanan, E. Kijak, L. Amsaleg, and Y. Avrithis, "Alignmixup: Improving representations by interpolating aligned features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 19 174–19 183.

[63] F.-E. Jannat and A. R. Willis, "Improving classification of remotely sensed images with the swin transformer," in *SoutheastCon 2022*, 2022, pp. 611–618.

[64] J. Wang, W. Yang, H.-C. Li, H. Zhang, and G.-S. Xia, "Learning center probability map for detecting objects in aerial images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 4307–4323, 2021.

[65] 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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9756–9765.

[66] J. Wang, C. Xu, W. Yang, and L. Yu, "A normalized gaussian wasserstein distance for tiny object detection," *CoRR*, vol. abs/2110.13389, 2021.

[67] K. C. et al, "MMDetection: Open mmlab detection toolbox and benchmark, 2019."

[68] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[69] G. Cheng, X. Yuan, X. Yao, K. Yan, Q. Zeng, X. Xie, and J. Han, "Towards large-scale small object detection: Survey and benchmarks," 2023.

[70] A. K.R., K. A.K., S. S. B., P. P. K., C. V. Dhareshwar, and D. G. Johnson, "Manipal-uav person detection dataset: A step towards benchmarking dataset and algorithms for small object detection," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 195, pp. 77–89, 2023.

[71] R. Del Prete, M. D. Graziano, and A. Renga, "Retinanet: A deep learning architecture to achieve a robust wake detector in sar images," in *2021 IEEE 6th International Forum on Research and Technology for Society and Industry (RTSI)*, 2021, pp. 171–176.

[72] C. Chen, Y. Zhang, Q. Lv, S. Wei, X. Wang, X. Sun, and J. Dong, "Rrnet: A hybrid detector for object detection in drone-captured images," 10 2019, pp. 100–108.

[73] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Gcnet: Non-local networks meet squeeze-excitation networks and beyond," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 1971–1980.

[74] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 764–773.

[75] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2999–3007.