drones

Article

PVswin-YOLOv8s: UAV-Based Pedestrian and Vehicle Detection for Traffic Management in Smart Cities Using Improved YOLOv8

Noor Ul Ain Tahir [1], Zhe Long [1], Zuping Zhang [1,*], Muhammad Asim [2,3,*] and Mohammed ELAffendi [2]

[1] School of Computer Science and Engineering, Central South University, Changsha 410083, China; 214718021@csu.edu.cn (N.U.A.T.); williamlon@csu.edu.cn (Z.L.)
[2] EIAS Data Science and Blockchain Laboratory, College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia; affendi@psu.edu.sa
[3] School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China
* Correspondence: zpzhang@csu.edu.cn (Z.Z.); asimpk@gdut.edu.cn or masim@psu.edu.sa (M.A.)

Abstract: In smart cities, effective traffic congestion management hinges on adept pedestrian and vehicle detection. Unmanned Aerial Vehicles (UAVs) offer a solution with mobility, cost-effectiveness, and a wide field of view, and yet, optimizing recognition models is crucial to surmounting challenges posed by small and occluded objects. To address these issues, we utilize the YOLOv8s model and a Swin Transformer block and introduce the PVswin-YOLOv8s model for pedestrian and vehicle detection based on UAVs. Firstly, the backbone network of YOLOv8s incorporates the Swin Transformer model for global feature extraction for small object detection. Secondly, to address the challenge of missed detections, we opt to integrate the CBAM into the neck of the YOLOv8. Both the channel and the spatial attention modules are used in this addition because of how well they extract feature information flow across the network. Finally, we employ Soft-NMS to improve the accuracy of pedestrian and vehicle detection in occlusion situations. Soft-NMS increases performance and manages overlapped boundary boxes well. The proposed network reduced the fraction of small objects overlooked and enhanced model detection performance. Performance comparisons with different YOLO versions (for example YOLOv3 extremely small, YOLOv5, YOLOv6, and YOLOv7), YOLOv8 variants (YOLOv8n, YOLOv8s, YOLOv8m, and YOLOv8l), and classical object detectors (Faster-RCNN, Cascade R-CNN, RetinaNet, and CenterNet) were used to validate the superiority of the proposed PVswin-YOLOv8s model. The efficiency of the PVswin-YOLOv8s model was confirmed by the experimental findings, which showed a 4.8% increase in average detection accuracy (mAP) compared to YOLOv8s on the VisDrone2019 dataset.

Keywords: YOLOv8; swin transformer; CBAM; soft-NMS; UAVs; pedestrian and vehicle detection

Citation: Tahir, N.U.A.; Long, Z.; Zhang, Z.; Asim, M.; ELAffendi, M.A. PVswin-YOLOv8s: UAV-Based Pedestrian and Vehicle Detection for Traffic Management in Smart Cities Using Improved YOLOv8. *Drones* **2024**, *8*, 84. https://doi.org/10.3390/drones8030084

Pablo Rodríguez-Gonzálvez

Received: 9 January 2024
Revised: 8 February 2024
Accepted: 24 February 2024
Published: 28 February 2024

1. Introduction

In smart cities, computer vision has gained popularity as a means of reducing traffic congestion and averting accidents in intelligent transportation systems (ITS) [1]. Congestion develops as traffic volume rises, which lowers object speeds and has negative effects such as higher consumption of gasoline, wasted time, more mental exertion, and increased air pollution [2]. Target location detection and categorization (vehicles, pedestrians, etc.) are crucial components of ITS that can help reduce traffic congestion. Notably, significant interest has been shown in unmanned aerial vehicles (UAVs), especially drones, in the domains of artificial intelligence, computer vision, and intelligent transportation [3]. The use of UAVs is expanding in a variety of fields, including agriculture, sports, the military, and traffic monitoring, due to the ongoing decrease in manufacturing costs and evolving flight control technologies. Pedestrian and vehicle detection plays a crucial role in the operations conducted by UAVs, which shows the significant importance of the current study. However, the wide field of vision offered by UAVs' high flying altitude presents difficulties, such as the occurrence

of extremely small, occluded entities and complicated backgrounds in the images that are taken. Because of this complexity, object detection becomes more challenging, and UAV-based object detection places a premium on improving detection performance.

The fundamental goal of the vital field of computer vision is to detect various objects in digital images [4]. Over recent years, a multitude of architectures, including convolutional neural networks (CNNs), fully connected networks (FCNs), and vision transformers, have been vital in driving the development of pedestrian and vehicle detection algorithms. In particular, there are two primary categories into which detectors based on CNNs can be generally divided as shown below.

- One-stage detectors (Fully Convolutional One-Stage (FCOS) [5], DEtection TRansformer (DETR) [6], EfficientDet [7], Single Shot Multibox Detector (SSD) [8], and You Only Look Once (YOLO1-8) [9];)
- Two-Stage detectors (Spatial Pyramid Pooling Networks (SPPNet) [10], centerNet [11], the R-CNN family [12–14], and Feature Pyramid Network (FPN) [15]).

In the domain of vehicle and pedestrian detection for UAVs, there is a growing trend to integrate both one-stage and two-stage object detection (OD) algorithms. These methods find application in scenarios demanding precise object detection and localization, showing promising results in this specialized field. However, pedestrians and vehicles in natural settings present a multiscale structure, with the UAV perspective often capturing small, low-contrast elements against the background and occluded features. Consequently, the conventional detection algorithms listed above are not directly applicable to the challenges faced in UAV object detection. Several initiatives have been undertaken to enhance detection performance, including attention mechanisms such as CBAM [16], and other mechanisms [17,18] that have emerged to improve detection performance by leveraging positional information. This improvement is achieved by using large-size convolutional kernels to decrease the input tensor channel dimension. In the area of computer vision, Transformer networks have recently been shown to offer qualitative performance. To capture relationships within a sequence, the conventional architecture of a visual transformer depends on a self-attention mechanism [19,20]. The Swin Transformer is a novel hierarchical vision backbone network that uses a multi-head self-attention system to encode global, local, and contextual data with flexibility while focusing on imagining patches [21]. Across a range of tasks involving computer vision, including pixel-level semantic segmentation [22], region-level OD [23], and image-level classification [24], it has shown remarkable performance. The Swin Transformer shows particular resilience in difficult situations such as extreme occlusions, erratic patch positions, and non-salient backdrop areas. However, despite its success, the Swin Transformer's distinct encoding–decoding structure, which sets it apart from ordinary CNNs, means that employing it alone for object detection requires a significant amount of processing power. To be more precise, every Swin Transformer encoder is made up of two sublayers, a completely connected layer and a multi-head attention layer, with residual connections being used in between. Through a self-attention mechanism, this design makes feature representation exploration easier [25,26]. Recent research on publicly available datasets, such as COCO [27], has substantiated the enhanced performance of the Swin Transformer, especially in scenarios involving substantial object occlusion [28]. Recently, the field of UAVs has also seen the utilization of different object detectors, Swin Transformers, and attention modules. We propose a novel method that we refer to as PVswin-YOLOv8s to address extremely small item detection and missed detection issues. This model leverages the effectiveness of YOLOv8s as the fundamental network, integrating the Swin Transformer for enhanced performance, and incorporates the CBAM module to improve pedestrian and vehicle detection when deployed on UAVs.

Our primary contributions to this work are listed below.

1. We introduce the PVswin-YOLOv8s model by utilizing YOLOv8s as a baseline model with a Swin Transformer block for pedestrians and vehicle detection based on UAVs. This integration involved replacing the last C2f layer from the backbone network of

YOLOV8s with a Swin Transformer block. The models incorporate global feature extraction for detecting extremely small items;
2. Then, to overcome the issues of missed detection, we incorporate the CBAM module into the YOLOv8s neck network. This works well for extracting feature information flow inside the network;
3. We implement Soft-NMS in YOLOv8s as a replacement for NMS to improve detection achievement in occlusion scenarios. Occlusion is a major problem in the detection of objects for UAVs, and NMS methods frequently lead to missed identifications. Soft-NMS enhances accuracy for detection and manages overlapped bounding boxes with efficacy.

The rest of the paper is structured as follows: Section 2 contains the related work. The methods and structure of PVswin-YOLOv8s are covered in Section 3. A thorough description of the tests and findings along with detailed discussion is given in Section 4. Finally, the conclusions are presented in Section 5.

2. Related Work

The Visdrone dataset serves as a critical resource for UAV research, offering a rich collection of small, obscured objects set against intricate backgrounds. This dataset, specifically designed for UAV applications, encompasses scenarios that are pivotal for the traffic management of smart cities. Our model, termed 'PVswin-YOLOv8s', is inspired by the dataset's diversity, which includes the detection of individuals, pedestrians, and various vehicles. The 'PVswin' nomenclature underscores our focus on addressing the multifaceted challenges of traffic congestion in smart cities. This literature review section provides a detailed examination of prior research efforts within the UAV domain, particularly emphasizing OD methodologies utilizing the Visdrone dataset. We critically assess the limitations of existing algorithms, such as their struggles with occlusions, small object detection, and missed identifications, which our proposed model aims to overcome. By situating our research within this broader context, we underscore the novelty and significance of our contributions to the field.

In the succinct overview of prior research efforts within the UAV domain, particularly emphasizing using the Visdrone2019 dataset, Payal et al. [29] proposed a Dilated RCNN-based model and Feature Fusion Module. The system was applied to the VisDrone dataset, which achieved a 35.04% mAP. Liu et al. [30] proposed an enhanced vehicle detection algorithm in intelligent transportation applications that addresses challenges posed by dense distribution and scale variations in UAV images. The proposed model excels in small object detection, surpassing the baseline YOLOX by an impressive 3%. However, its ability to detect extremely small objects is inadequate.Deng et al. introduced LAI-YOLOv5s [31], an optimized YOLOv5s variant for aerial image object detection, which incorporates DFM-CPFN for feature fusion and a VoVNet module for enhanced feature extraction, achieving a 40.0% mAP:0.5. This lightweight algorithm demonstrates improved detection accuracy and computational efficiency compared to its predecessors. Lou et al. [32] introduced the DC-YOLOv8 model for small object detection, incorporating a downsampling technique to preserve context features. The model includes an enhanced feature fusion network for the effective integration of shallow and deep information, along with a newly proposed network structure. These innovations collectively contribute to a 2.5% increase in detection accuracy compared to YOLOv8s, addressing challenges associated with human fatigue in complex scenes.To improve feature extraction and localization classification, Li et al. [33] developed STF-YOLO, an advanced UAV remote sensing image small target detection algorithm that combines the Swin Transformer with CNNs. This improved detection performance over previous approaches was demonstrated by metrics like 3.9% mAP and 2.0% AP50 on the VisDrone dataset. Tang et al. [34] presented HIC-YOLOv5, an enhanced YOLOv5 model that employs a small object detection head for high-resolution feature maps, an involution block for channel information enhancement, and CBAM for feature importance. As a result, the proposed model achieved 36.95% mAP:0.5. Sirisha and

Sudha [35] introduced a novel object detection framework known as PvSAMNet for UAV imagery, which employed a Transformer backbone and a split attention module to enhance feature extraction and object detection. The model achieves a mAP of 38.74%, addressing the challenges of UAV-based object detection.

The literature review showcases the effectiveness of the Swin Transformer and CBAM in enhancing object detection, yet their integration into UAV-based models has not been fully optimized. Building on these findings, our research introduces a refined integration of these components into the YOLOv8s framework, tailored to enhance detection accuracy for UAV imagery. This approach aims to capitalize on their proven benefits while streamlining the model for optimal performance in UAV-based traffic management scenarios.

3. Proposed Method

3.1. Network Framework of YOLOv8s

One of the most popular networks for object detection is YOLOv8 [36], the newest model in the YOLO family of detection models. With its increased accuracy and faster detecting speed, it performs better than many of the prior models. YOLOv8 offers scalability, with network models available in different sizes to meet various usage requirements. Leveraging these advantages, this study aims to employ the enhanced YOLOv8s for the detection of pedestrians and vehicles. The backbone network, neck, and head network are the three main parts of the YOLOv8 algorithm design, as Figure 1a illustrates.

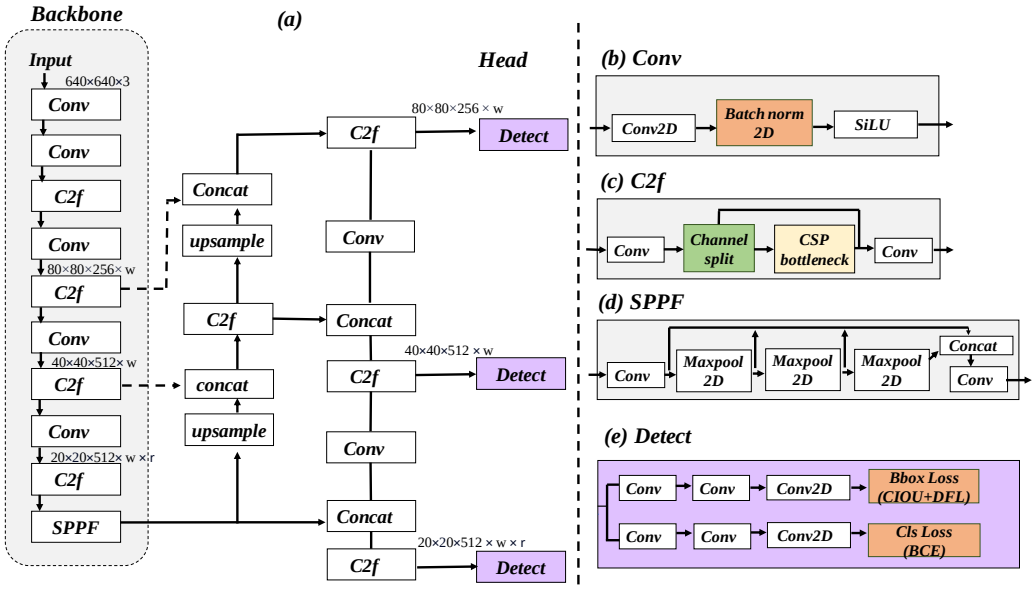


Figure 1. (**a**) The network structure of YOLOv8, where w stands for width and r for ratio, and the feature map size are represented by the parameters. (**b**) represents the architecture of Conv layers (**c**) represents the structure of C2f layers (**d**) represents the structure of SPPF (**e**) represents the architecture of detection layers in the head section.

First, the input image is resized to maintain a uniform dimension of 640 × 640 for all input images. The backbone network creates three-layer feature maps (80 × 80, 40 × 40, and 20 × 20) by extracting feature maps from the input images using repeated convolutions. The neck network then incorporates feature fusion layers to efficiently combine the image features and reduce information loss. The merging process incorporated the distinctive pyramid structure of the Feature Pyramid Network (FPN) and Path Aggregation Network (PANet). This integration facilitates the transfer of strong semantic features to top-down-level feature maps, using the FPN structure to enhance information flow. Finally, the joint use of the PAN and FPN structures improves the neck network feature fusion capacity. A detection head is used to obtain the final detection outcomes. More specifically, the core elements of YOLOv8s consist of the CBS module shown in Figure 1b (Convolution, Sigmoid-

weighted Linear Unit (SiLU) activation, and Batch Normalization (BN)), the Spatial Pyramid Pooling Fusion (SPPF) module depicted in Figure 1d, and the C2F module represented in Figure 1c (which incorporates features from ELAN [37] and is inspired by C3 to provide lightweight functionality). The CBS structural combination makes it possible to select models with excellent accuracy and efficiency. The module tackles gradient dispersion using feature reuse, which guarantees that the original data are preserved to a large extent. Moreover, the SPPF module increases classification accuracy by extracting and fusing high-level features and gathering a range of high-level semantic characteristics through several maximum pooling methods. By adopting the desired decoupling head technique, YOLOv8 sets itself apart from previous YOLO architectures [38]. Figure 1e illustrates the detection head, which consists of regression and classification branches using binary cross-entropy (BCE) loss for classification and distributed focal loss (DFL) for regression (localization) [39]. The DFL features separate sections for localization and categorization and is designed for one-stage object detectors to enhance detection performance. By modeling box locations as broad distributions, it focuses the network's attention on values that are close to the ground-truth label [40]. In addition, a task-aligned assigner is used by the YOLOv8 head to assign ground-truth objects to expected bounding boxes [41,42] by combining both classification and localization information.

3.2. Pedestrian and Vehicle Detection Network (PVswin-YOLOv8s)

We have introduced PVswin-YOLOv8s, a detection model that integrates YOLOv8s, Swin Transformer, and CBAM to enhance the robustness of pedestrian and vehicle detection. We selected the small version of YOLOv8 for its improvements over other variants based on its balance of computational efficiency and detection accuracy, which are critical for UAV applications. YOLOv8s, being a smaller model, offers a lower number of parameters and less memory usage, which is essential for the constrained computational resources of UAVs. Our model strategically replaces the last C2f layer in the YOLOv8s backbone with a Swin Transformer block, as illustrated in Figure 2a. This integration operates on low-resolution feature maps (20×20), which reduces computational load and memory requirements. It addresses the YOLOv8s limitation in capturing global and contextual information, a common challenge for CNNs with limited receptive fields [43], by leveraging the Swin Transformer's ability to capture long-distance dependencies and diverse local information [21]. The Swin Transformer block employs a default patch size of 4×4 pixels for initial patch embeddings, which is then dynamically expanded through the shifted window mechanism, effectively increasing the receptive field size without adding computational complexity. This integration ensures the model retains the speed and precision of YOLOv8s while enhancing its capability to detect objects in complex UAV imagery, making it an effective solution for traffic management in smart cities. Then, CBAM is introduced into the neck network of YOLOv8 to improve feature information by utilizing its dual-channel nature, as described in Figure 2b. Additionally, we implement soft-NMS for better detection of overlapping objects as a replacement for NMS, which is used in the YOLOv8 model to refine candidate boxes for pedestrians and vehicle detection, with the threshold balance affecting accuracy in the scenario of UAVs. By addressing occlusion issues, soft-NMS integration offers an adaptable method that maximizes detection. This comprehensive approach aims to maximize the detection of the quantity and location information for pedestrians and vehicles. The network is shown in Figure 2.

3.3. Swin Transformer Block

To address the problem of the varying sizes of pedestrians and vehicles in UAV imagery, this work incorporates the sliding window multi-head self-attention mechanism of the Swin Transformer, which improves the ability of the model to capture and process global features effectively. Swin Transformer [21], which is based on the Transformer [44] design, has shown remarkable performance in computer vision applications like detection, segmentation, and classification. An overview of the Swin Transformer architecture is

shown in Figure 3. Using a patch division module, the input image is split up into discrete, non-overlapping pieces. Every patch is regarded as a "token", with features created by joining the raw values of its pixels. With the 4 × 4 patch size used in this investigation, each patch has a feature dimension of 4 × 4 × 3 = 48. Then, as shown in Figure 3a, a linear embedding layer projects the raw value feature to an arbitrary dimension, represented by the design element C. Figure 3b depicts this approach. To construct the Swin Transformer, a modified Multi-head Self Attention (MSA) module based on "shifted windows" (SW-MSA) and "windows" (W-MSA) was installed in place of the normal MSA module in a standard Transformer block. The other layers were left unchanged. This module is replaced by a two-layer Multi-Layer Perceptron (MLP) with nonlinearity between Rectified Linear Units (ReLU). Before and after each MSA module and MLP layer, LayerNorm, the normalization layer, and a residual connection were applied.

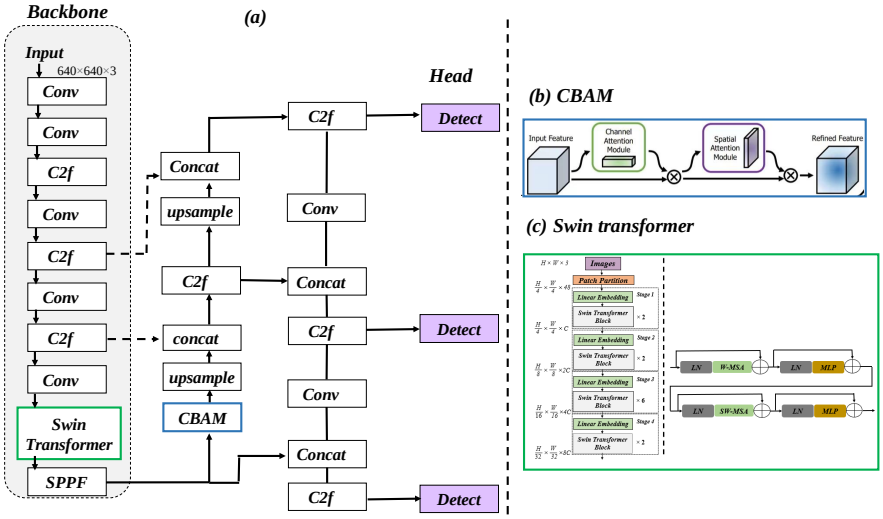


Figure 2. (**a**) Pedestrian and vehicle detection model (PVswin-YOLOv8s) (**b**) represents the structure of CBAM module (**c**) represents the structure of swin transformer.

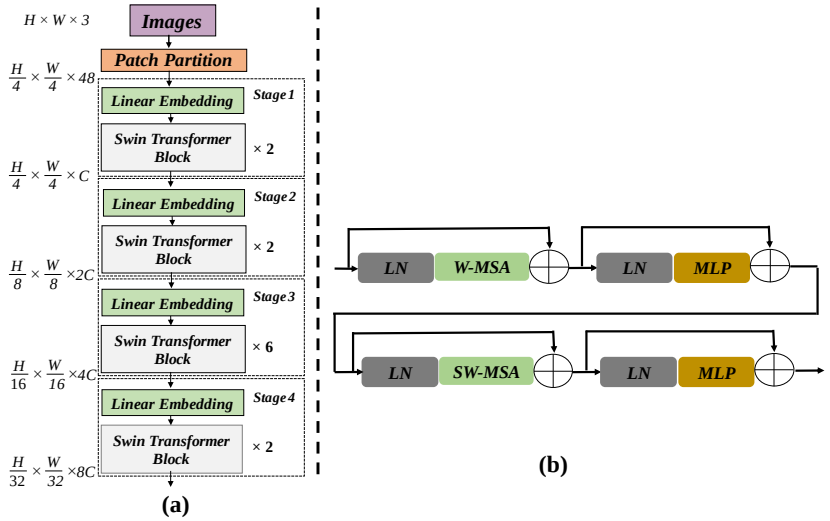


Figure 3. (a) Swin transformer model (b) Two successive swin transformer Blocks

3.4. CBAM

Based on the decomposition of the attention mechanism in a three-dimensional feature map into channel attention and spatial attention, the feed-forward convolutional neural network attention module known as the Convolutional Block Attention Module (CBAM [16]) is illustrated in Figure 2b. The spatial attention module should be included after the channel attention module to achieve the best results. This lightweight module is

easily implemented into any CNN architecture for thorough training. Channel attention entails applying methods like maximum pooling and average pooling within each channel to compress the spatial dimension of the feature map F into a one-dimensional vector. The resulting aggregated information is processed through a shared network to compile spatial data, The channel attention mechanism, expressed in Equation (1), incorporates the $f(\sigma)$, $W_0 \in \mathbb{R}C \times C/r$, and $W_1 \in \mathbb{R}C \times C/r$. The shared MLP weights for both inputs denoted as W_0' are succeeded by the activation function of the ReLU, applied to W_1. The input of the spatial attention module is the feature map that results from multiplying the output of the channel attention module by F. Max pooling and average pooling between the channel dimensions were used to produce channel compression, and the compressed results were then concatenated. The final weight is calculated using the Sigmoid coefficients and a 7×7 fusion operation, as per Formula (2), where σ represents the sigmoid function and $f^{7\times7}$ is the convolution operation with a 7×7 filter size.

$$\begin{aligned} M_c(F) &= \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \\ &= \sigma\Big(W_1\Big(W_0(F^c_{\text{avg}}) + W_1(W_0(F^c_{\text{max}})\Big)\Big) \end{aligned} \tag{1}$$

$$M_s(F) = \sigma(f^{7\times7}(F^s_{\text{avg}}, F^s_{\text{max}})) \tag{2}$$

3.5. Soft-NMS

We opt to integrate the Soft-NMS algorithm with the Gaussian reset method as a replacement for the traditional NMS algorithm within the YOLOv8 framework. This choice offers several advantages, notably its simplicity and versatility. The Soft-NMS algorithm requires minimal modification to the traditional NMS approach, eliminating the need for additional parameters and ensuring ease of implementation. Despite its enhanced functionality, Soft-NMS maintains the same algorithmic complexity as traditional NMS, thereby avoiding an increase in computational burden. Furthermore, its seamless integration into existing object detection pipelines facilitates its adoption across various applications without the need for additional training. Soft-NMS enhances the robustness and accuracy of object detection systems by preventing object missed detection resulting from overlapping proposals [45]. The usage of NMS will result in objects in the overlapping region being miss identified because of the overlapping features of UAV imagery. By adding a confidence decay factor, Soft-NMS preserves the confidence information of overlapping objects and resolves overlapping concerns in occlusion. Soft-NMS improves the localization accuracy of high-confidence bounding boxes by adjusting suppressing based on confidence as well as overlap, in contrast to classic NMS (Non-Maximum Suppression), which primarily depends on IoU for suppression and may mistakenly remove essential objects. Specifically designed to handle occlusion scenarios, Soft-NMS has the following mathematical expression (3):

$$s_i = \begin{cases} s_i, & iou(\mathcal{M}, b_i) < N_t \\ s_i(1 - \text{iou}(\mathcal{M}, b_i)), & iou(\mathcal{M}, b_i) \geq N_t \end{cases} \tag{3}$$

The expression iou (.) measures the intersection over union ratio within the i-th candidate box M, N_t indicates an established limit, and s_i indicates the result of the i-th candidate box. M and b_i represent the coordinates of the candidate box alongside the greatest rating and the coordinates of the i-th candidate box, respectively.

4. Experimental Results

4.1. Overview of the Experiment

In this section, we first go over the dataset and parameters that were utilized for this research, and then we introduce the evaluation metrics that are connected to our experimental findings.

4.1.1. Dataset and Hyper Parameters

VisDrone2019 [46] is a noteworthy group of UAV aerial photos. The AISKYEYE data-mining team and Tianjin University collaborated to create this dataset, which features a range of urban settings from multiple Chinese cities. Using different UAVs, the dataset provides rich and extensive resources by capturing imagery from a variety of situations, viewpoints, and tasks. Among the noteworthy features are the large variety of detection object categories (from highly diverse to monotonous), varying quantities of detection objects, sparse and dense distributions of detection objects, and observations made in both day and night lighting conditions. Without compromising originality, this large and diverse dataset offered insightful information for the study and development of UAV-related applications. Ten different class types were covered by VisDrone2019, including pedestrians and vehicles. This study uses a dataset partitioning method akin to the VisDrone 2019 challenge, dividing the data into three separate subsets: 6471 instances in the training subset, 548 instances in the validation subset, and 1610 instances in the test subset. The insights obtained from manual labeling of the dataset are presented in Figure 4. The item distribution is highlighted in the first subfigure, which shows that pedestrians and vehicles are predominant. The second subplot showcases the dimensions of object bounding boxes in the dataset, maintaining consistent positioning of object box centers. The prevalence of numerous small-sized objects is apparent from the observed sizes of the object bounding boxes. The third subplot illustrates the distribution of center point coordinates within object bounding boxes, with the majority concentrated in the central and lower-right regions of the image data area. The fourth subplot, a scatter plot representing the width and height of object bounding boxes, highlights the dominance of small objects in the dataset. VisDrone2019 offers a dataset that is densely distributed and full of extremely small objects. Unlike standard computer vision datasets, it is a demanding dataset for UAV-related computer vision tasks because of its unique qualities, which include numerous scales, scenery, and viewpoints.

The VisDrone 2019 dataset offers a distinctive challenge for computer vision problems connected to UAVs due to its dense distribution of small objects and diverse range of scales, landscapes, and viewpoints. This dataset is a challenging testbed for creating reliable detection algorithms because it deviates greatly from normal computer vision datasets. To overcome these obstacles and guarantee that our model operates at peak efficiency for UAV applications, we set the image size to 640 $\times$ 640 pixels. This resolution enables efficient deployment on edge devices by striking a compromise between computing efficiency and maintaining essential visual characteristics. Together with an Intel (R) CoreTM i5-6300U processor and 24 GB of RAM, our experimental configuration included Torchvision 0.10.1 for developing the model and the PyTorch 2.0.1 framework as shown in Table 1. The baseline version of YOLOV8 employed was Ultralytics 8.0.25. Pretraining weights were not used in any of the comparative investigations of training procedures to guarantee the comparable nature and integrity of the model impacts. To address the specific requirements of UAV-based applications, where target recognition and reasoning must be performed efficiently, the model must be optimized for minimal parameters, a low memory footprint, and swift inference. Consequently, YOLOv8s was selected as the baseline model for enhancement. This variant is characterized by its constrained network depth and width, yet it retains the core features of the YOLOv8 series, making it well-suited for the resource-limited environment of UAVs. To comprehensively evaluate the performances, we compared the proposed PVswin-YOLOv8s with previous versions of YOLO (YOLOv3-tiny, YOLOv5, YOLOv6, and YOLOv7), different sizes of YOLOv8 (YOLOv8n, YOLOv8s, YOLOv8m, and YOLOv8l), and some classical models (RetinaNet, CenterNet, cascade R-CNN, and faster R-CNN). Table 2 lists the primary hyperparameters that were employed in this study.

Table 1. The experimental environment for our model.

Parameters	Configuration
CPU	I5-6300U
GPU	NVIDIA A10, 22,592 MiB
GPU memory	24 GB
Python	3.10.13
Ultralytics	YOLOv8.0.25
DL architecture	Pytorch2.0.1 + cu117, CUDA

Table 2. The key parameters we used during model training.

Parameters	Setup
Epochs	200
Learning rate	0.01
Image size	640 × 640
Batch size	8
optimizer	SGD
Weight decay	5×10^{-4}
Momentum	0.932
Mosaic	1.0
Patience	50
Close mosaic	last 10 epochs

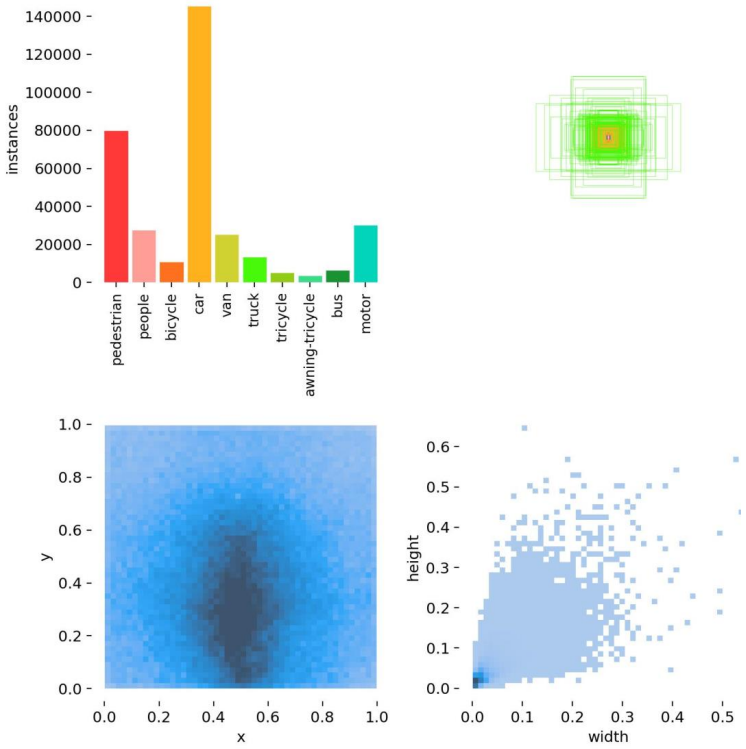


Figure 4. Information regarding the manually labeled objects in the VisDrone2019 dataset.

4.1.2. Evaluation Matrix

We used model size, detection speed per image, precision (P), recall (R), F1-score, mAP0.5, and mAP0.5:0.95, in addition to other measures, to assess the detection abilities of our proposed PVswin-YOLOv8s model. The formulation for some of these evaluation metrics involves specific parameters: True Positive (TP) denotes instances where the prediction correctly identifies the positive class; False positives (FP) indicate instances where the prediction incorrectly identifies the Positive class; and False Negative (FN) represents instances in which the prediction incorrectly identifies the negative class. Meanwhile, the ratio of their intersection to their union, or the intersection over union (IoU), indicates the percentage of contact between the bounding box and the true box.

4.1.3. Precision and Recall

As Equation (4) illustrates, precision [47] is calculated as the ratio of positively predicted samples by the model to the total number of observed samples.

$$Precision = TP/TP + FP \tag{4}$$

The percentage of positively predicted samples that the model properly recognized about the total number of positive samples present is known as recall [47]. Equation (5) is utilized in the computation of recall.

$$Recall = TP/TP + FN \tag{5}$$

The precision–recall curve, or P–R Curve, is a useful representation for visualizing model performance. The study by Boyd et al. [48] displays a plot of recall on the x-axis and precision on the y-axis.

4.1.4. F1-Score

The algorithm performance can potentially be accurately evaluated by using the F1-score [47], which considers both Precision and Recall. Equation (6) shows how the F1-score is calculated.

$$F1 = 2 * Precision * Recall / Precision + Recall \tag{6}$$

In image recognition tasks, the F1-score is a statistic that is frequently used to evaluate an algorithm's object detection ability. As a result, we chose to employ the F1-score as the primary indicator for assessing our model efficacy.

4.1.5. Mean Average Precision (mAP)

The area under the precision–recall curve is the same as the average precision (AP), which may be calculated using the formula below.

$$AP = \int P(R)dR \tag{7}$$

The Mean Average Precision (mAP), a general indicator of the model's detection performance, is computed using the weighted average of the AP values for each sample category.

$$mAP = \frac{1}{N}\sum_{t=1}^{N} AP_i \tag{8}$$

In Equation (8), APi represents the AP value corresponding to category index i and N indicates the total number of categories in the training dataset (n = 10 in our case). The mean average accuracy is represented by mAP0.5 when the detection model's IoU is set to 0.5. Conversely, the mean average accuracy is represented by mAP0.5:0.95 when the IoU values vary from 0.5 to 0.95.

4.2. Results and Analysis

We conducted four sets of comparative studies to demonstrate the superiority and efficacy of our proposed PVswin-YOLOv8s model. The first set involved a thorough comparison with the baseline model YOLOv8s. In the second set, we compared our model with four versions of YOLOv8. The third set involved a comparison with various YOLO series versions. In the final set, we extended our comparisons to include other classical models renowned for their exceptional performance.

4.2.1. Comparison with YOLOv8s

We conducted comparative experiments between our proposed PVswin-YOLOv8s and the baseline YOLOv8s model in the validation set of the VisDrone2019 dataset to highlight the enhanced detection performance of our model. Table 3 presents mAP values for individual classes and mAP0.5 values across all classes, showcasing the improved performance of our model compared to YOLOv8s. The proposed model exhibited a 4.8% enhancement in mAP, as indicated in Table 3.

The variation trends in important evaluation measures for our proposed PVswin-YOLOv8s model and YOLOv8s during the training phase are shown in Figure 5. We noticed that the model halted at the 162nd epoch during the training procedure. To conserve computer power, we decided that all model training should have 200 epochs. The findings show that our proposed model performs better than YOLOv8s in terms of the precision, recall, and mAP0.5 detection measures. Furthermore, our approach outperforms YOLOv8s in terms of detecting capabilities and training speed.

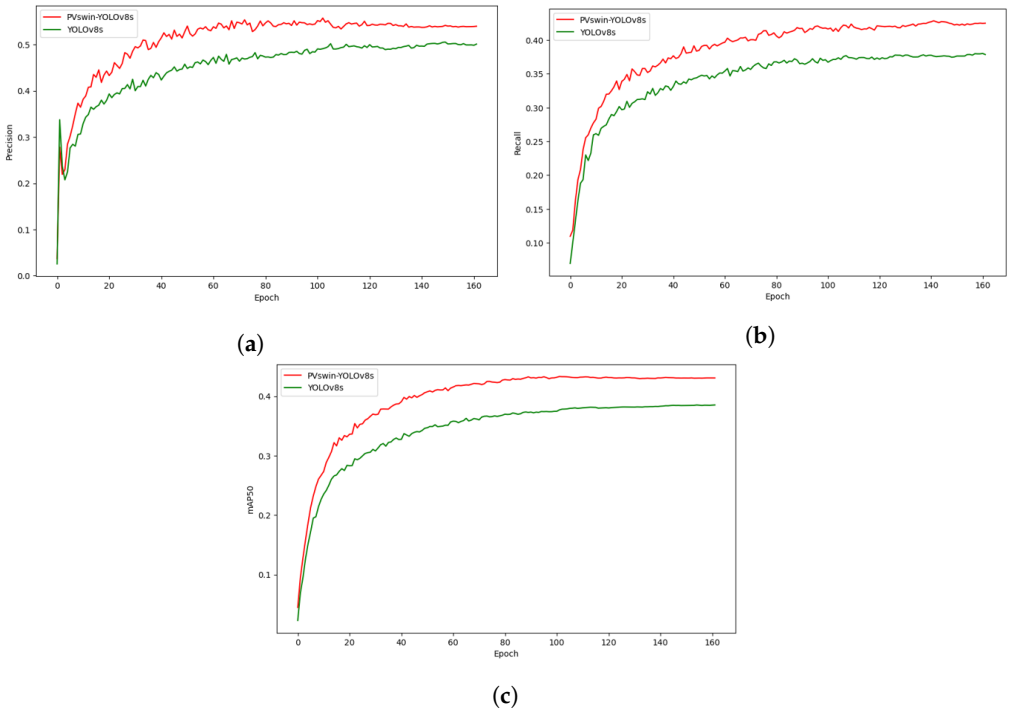
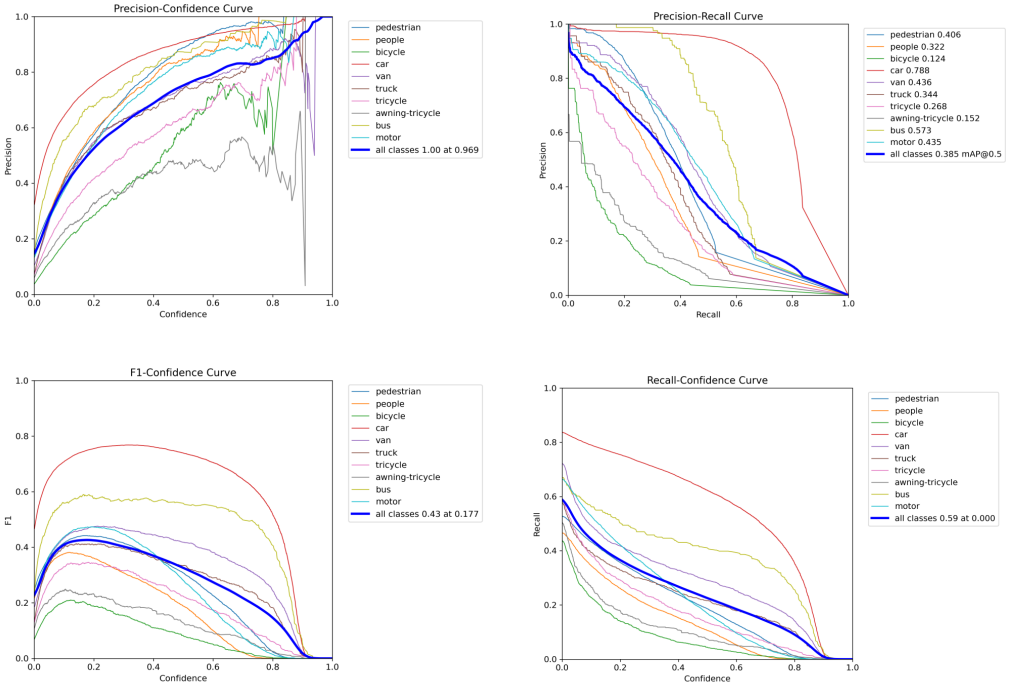


Figure 5. Training curves (**a**) showing precision between PVswin-YOLOv8s and YOLOv8s on the Visdrone2019-train dataset; (**b**) showing recall between PVswin-YOLOv8s and YOLOv8s; (**c**) showing a comparison between Pvswin-YOLOv8s and YOLOv8s in mAP.

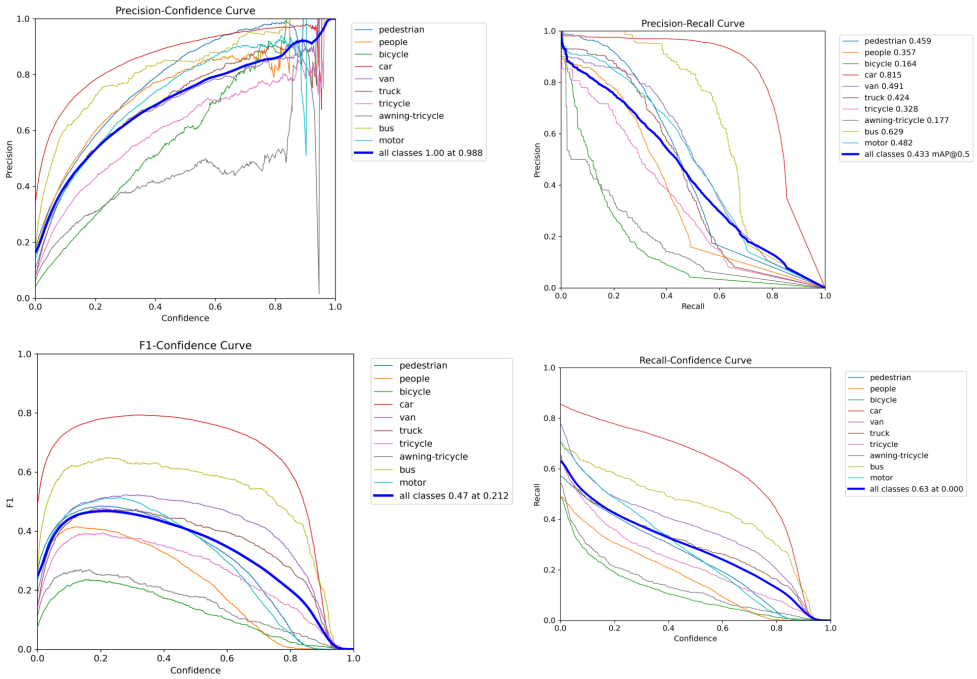
Figure 6 displays the experimental results of YOLOv8s and the proposed PVswin-YOLOv8s model on the VisDrone2019-val dataset. Because it covers a bigger region, the PVswin-YOLOv8s curve for the entire class has a higher F1 value, according to the F1–Confidence graph and the precision–recall Curve (P–R Curve). This discovery suggests that the enhancements were advantageous to the model.

Table 3. Comparative analysis of detection accuracy between the PVswin-YOLOv8s model and YOLOv8s in Visdrone2019-val. The bolded numbers are the most accurate indicators for each category.

Models	Pedestrian	People	Bicycle	Car	Van	Truck	Tricycle	Awning-Tricycle	Bus	Motor	mAP0.5 (%)
YOLOv8s	40.6	32.2	12.4	78.8	43.6	34.4	26.8	15.2	57.3	43.5	38.5
PVswin-YOLOv8s	**45.9**	**35.7**	**16.4**	**81.5**	**49.1**	**42.4**	**32.8**	**17.7**	**62.9**	**48.2**	**43.3**



(a) Evaluation factor of yolov8s



(b) Evaluation factors of Pvswin-YOLOv8s

Figure 6. Evaluation factors of Yolov8s and the proposed PVswin-YOLOv8s.

4.2.2. Comparison with Other Versions of YOLOv8

To demonstrate the efficacy of the proposed strategy, we conducted a comparative study between the proposed PVswin-YOLOv8s model and the other YOLOv8 variations (YOLOv8n, YOLOv8s, YOLOv8m, and YOLOv8l) using the VisDrone2019 validation and test datasets. Table 4 displays the experimental outcomes on the validation set. Precision and mAP0.5, the two evaluation indices, show the highest values for the upgraded model. Meanwhile, its mAP0.5:0.95 is only 0.1% lower than YOLOv8l, and its F1-score is comparable to YOLOv8l, suggesting that the detection performance is better than larger-scale models. Meanwhile, Table 5 displays the experimental outcomes on the test set. The precision, F1-score, and mAP0.5 are better for PVswin-YOLOv8s model than for all versions n, s, m, l of YOLOv8. The results of the experiment show that our proposed model significantly improves the accuracy of small object detection.

Table 4. The experimental findings use four distinct sizes of YOLOv8 and our proposed model on VisDrone2019-val. The bolded numbers are the most precise indicators.

Models	Precision (%)	Recall (%)	F1-Score	mAP0.5 (%)	mAP-0.5:0.9 (%)	Detection Time (*ms*)	Model Size (*MB*)
YOLOv8n	43.2	32.5	0.37	32.6	18.9	**4.4**	**6.2**
YOLOv8s	49.9	38	0.43	38.5	23	6.0	22.5
YOLOv8m	52.5	41	0.46	41.7	25.2	11.4	49.2
YOLOv8l	54.2	**42.4**	**0.47**	43	**26.5**	14.8	87.6
Proposed: PVswin-YOLOv8s	**54.5**	41.8	**0.47**	**43.3**	26.4	6.2	21.6

Table 5. The experimental findings use four distinct sizes of YOLOv8 and our proposed model on VisDrone2019-test. The bolded numbers are the most precise indicators.

Models	Precision (%)	Recall (%)	F1-Score	mAP0.5 (%)	mAP-0.5:0.9 (%)	Model Size (MB)
YOLOv8n	38.2	28.7	0.32	26.1	14.6	**6.2**
YOLOv8s	44.5	32.3	0.37	30.5	17.4	22.5
YOLOv8m	46.3	35.3	0.40	33.4	19.4	49.2
YOLOv8l	48.1	**37**	0.41	35	**20.6**	87.6
Proposed: PVswin-YOLOv8s	**49.7**	36.4	**0.42**	**35.2**	20.4	21.6

4.2.3. Comparison with Other Versions of YOLO

The YOLO family of algorithms is examined in this work, including YOLOv3-tiny [49], a more compact version of the method. With the use of mosaic data augmentation, YOLOv5 [50] improves the speed and accuracy of model training. YOLOv6 [51] introduces an efficient decouple head, while YOLOv7 [52] utilizes the ELAN efficient network architecture. Table 6 provides a full summary of the outcomes of these comparison models.

Based on the analysis presented in Table 6, it can be concluded that previous YOLO series algorithms, even the sophisticated YOLOv3-Tiny, have poorer detection accuracy than the proposed PVswin-YOLOv8s. The latter is a lightweight model but this comes at the cost of greatly reduced detection accuracy. Compared to previous YOLO versions, YOLOv5, which has smaller model sizes, performs better in detection. YOLOv7 performs worse regarding model size and detection performance but better in terms of detection speed compared to PVswin-YOLOv8s. PVswin-YOLOv8s shows the best overall detection performance and the highest average detection accuracy in comparable studies. Even with a longer inference time, the enhanced model can still detect targets with success.

Table 6. The comparative experimental outcomes of our proposed model examined against different versions of YOLO algorithm on VisDrone2019-val. The bolded results indicate the best values.

Models	Precision (%)	Recall (%)	mAP0.5 (%)	mAP0.5:0.95 (%)	Detection Time (ms)	Model Size (MB)
YOLOv3 (tiny)	37.2	24.1	23.2	12.9	2.7	24
YOLOv5	42.3	31.9	31.5	18	3.8	**5.3**
YOLOv6	39.8	29.4	29.1	17	3.3	8.7
YOLOv7	50.2	41.1	37.9	19.9	**1.9**	72
YOLOv8s	49.9	38	38.5	23	6.0	22.5
PVswin-YOLOv8s	54.5	**41.8**	**43.3**	**26.4**	6.2	21.6

4.2.4. Comparison with the Classical Model

Next, we carried out a comparison study to assess the accuracy of PVswin-YOLOv8 against other widely used models, namely, Faster RCNN [14], Cascade R-CNN [53], RetinaNet [54], and CenterNet [11]. Table 7 presents the experimental outcomes. The proposed model outperforms other notable models. Interestingly, two-stage algorithms like Faster R-CNN show poorer detection speeds than their one-stage counterparts. Their feature maps also have lower resolution, which affects the precision of small-object detection. Concentrating on sample determination, RetinaNet could overlook things in cluttered scenes or situations where extremely small objects overlap. Although CenterNet eliminates processing linked to anchors, it has difficulties with small-object occlusion and dense scenes. The multilevel detection employed by Cascade R-CNN improves overall performance but adds complexity and training difficulty. RetinaNet may have trouble identifying small objects even when it employs multiscale feature fusion. The model proposed in this research provides better outcomes in this situation without having the drawbacks of RetinaNet.

Table 7. Comparison of PVswin-YOLOv8s with some classical models. The bolded results show the optimal values.

Models	mAP0.5 (%)	mAP0.5:0.95 (%)
Faster RCNN	37.2	21.9
Cascade R-CNN	39.1	24.3
RetinaNet	19.1	10.6
CenterNet	33.7	18.8
PVswin-YOLOv8s	**43.3**	**26.4**

In summary, the proposed PVswin-YOLOv8s model shows better detection performance than other models in all four comparison studies. The small-object detection structure of our model offers unique benefits over the other models examined in these experiments, which helps explain its overall better detection performance.

4.3. Ablation Test

To assess the impact of various enhancements on our PVswin-YOLOv8s model, we conducted ablation experiments using the VisDrone2019 dataset as shown in Table 8. The integration of the Swin Transformer into the YOLOv8s model for our model led to a notable improvement in the model's ability to capture global features, which are crucial for identifying small objects within complex traffic scenes. This enhancement resulted in a 2.1% increase in mAP0.5 and a 1.7% increase in mAP0.5:0.95, indicating that the model became more effective at detecting objects with varying scales and orientations, a common challenge in UAV imagery. The addition of the CBAM module further refined the feature extraction process by introducing channel and spatial attention mechanisms. This allowed

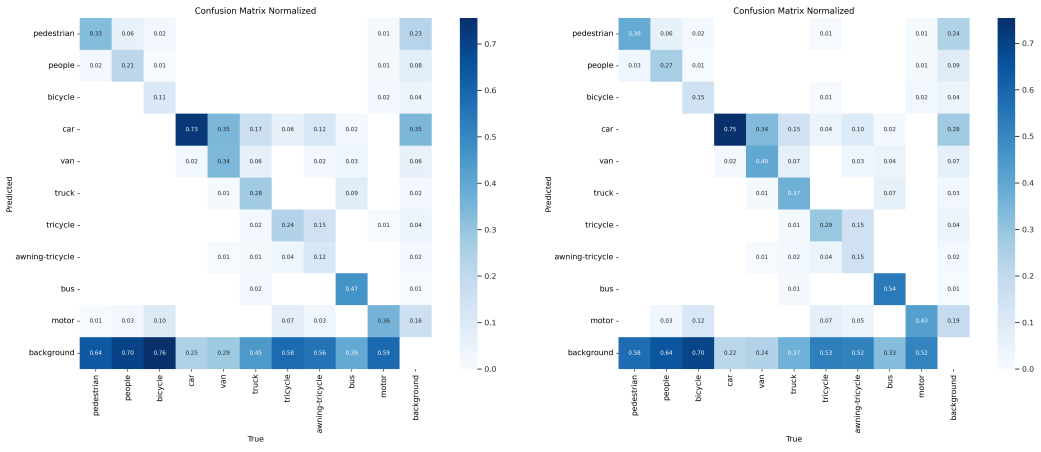
the model to focus on the most relevant features for misidentified objects, leading to a 1.3% increase in mAP0.5 and a 0.9% increase in mAP0.5:0.95. Incorporating Soft-NMS into the model addressed the issue of overlapping detection, a frequent occurrence in UAV imagery due to the proximity of objects in the detection field. Soft-NMS improved the mAP0.5 by an additional 1.1% and mAP0.5:0.95 by 0.8%, demonstrating its effectiveness in handling occlusions and ensuring that the model does not miss critical objects due to the presence of other pedestrians and vehicles. Overall, these enhancements collectively contributed to a more robust and accurate detection system for traffic management, enabling them to operate more effectively in dynamic and challenging environments.

Table 8. Ablation test results on visDrone2019-val dataset, where ↑ shows increment in results.

Models	**mAP0.5 (%)**	**mAP0.5:0.95 (%)**
Baseline YOLOv8s	38.5	23
YOLOv8s + Swin Transformer	40.6 (↑ 2.1%)	24.7 (↑ 1.7%)
YOLOv8s + Swin + CBAM	42.2 (↑ 1.3 %)	25.6 (↑ 0.9 %)
PVswin-YOLOv8s	**43**.3 (↑ 1.1%)	**26**.4 (↑ 0.8 %)

4.4. Visual Analysis

Interpretability problems with deep learning models impede the development and application of these models. We conducted evaluations and examined the model performance using three optics, confusion matrix, model inference results, and detection outcomes, to determine the visual efficacy of the proposed model for detection. This approach provides a straightforward means of understanding the model's detection capabilities. In Figure 7, we have generated confusion matrices for PVswin-YOLOv8s and YOLOv8s to illustrate the precision of our object classification method. The true and predicted categories are represented, respectively, by the rows and columns of the matrices. Values along the diagonal represent the percentage of correctly anticipated categories, whereas values elsewhere represent the percentage of wrongly predicted classes.



(**a**) (**b**)

Figure 7. (**a**) YOLOv8s confusion matrix (**b**) PVswin-YOLOv8s confusion matrix.

As can be seen in Figure 7, PVswin-YOLOv8s has a darker diagonal element in its confusion matrix than YOLOv8s, indicating its higher object categorization ability. Conversely, misidentifying extremely small autos and walkers as background is more probable, suggesting that a significant fraction of both groups are overlooked during detection. The modified model still has a rather low prediction accuracy, even though it reduces the percentage of missed detection for certain categories.

As shown in Figure 8, we conducted a comprehensive visual comparison of PVswin-YOLOv8s with YOLOv5 and the baseline YOLOv8s. Figure 8a provides a visual represen-

tation of ground-truth annotations, serving as a benchmark for evaluation. Figure 8b–d display the prediction results of YOLOv5, YOLOv8s, and PVswin-YOLOv8s, respectively. Notably, Figure 8e present the charts that illustrating the detection performance. The x-axis enumerates the categories within the VisDrone 2019 dataset, while the y-axis showcases the count of ground-truth labels and prediction results of models. This detailed breakdown highlights PVswin-YOLOv8s' superior ability to identify objects, particularly pedestrians and vehicles, compared to YOLOv5 and YOLOv8s. The model excels at detecting small and occluded objects, where YOLOv8s faces challenges. The strategic adjustments made to address issues with misidentified, occluded overlapped, and small targets have significantly contributed to the enhanced performance of PVswin-YOLOv8s. In conclusion, our model demonstrates superior overall performance compared to the baseline YOLOv8s, showcasing its advanced detection capabilities in challenging scenarios.

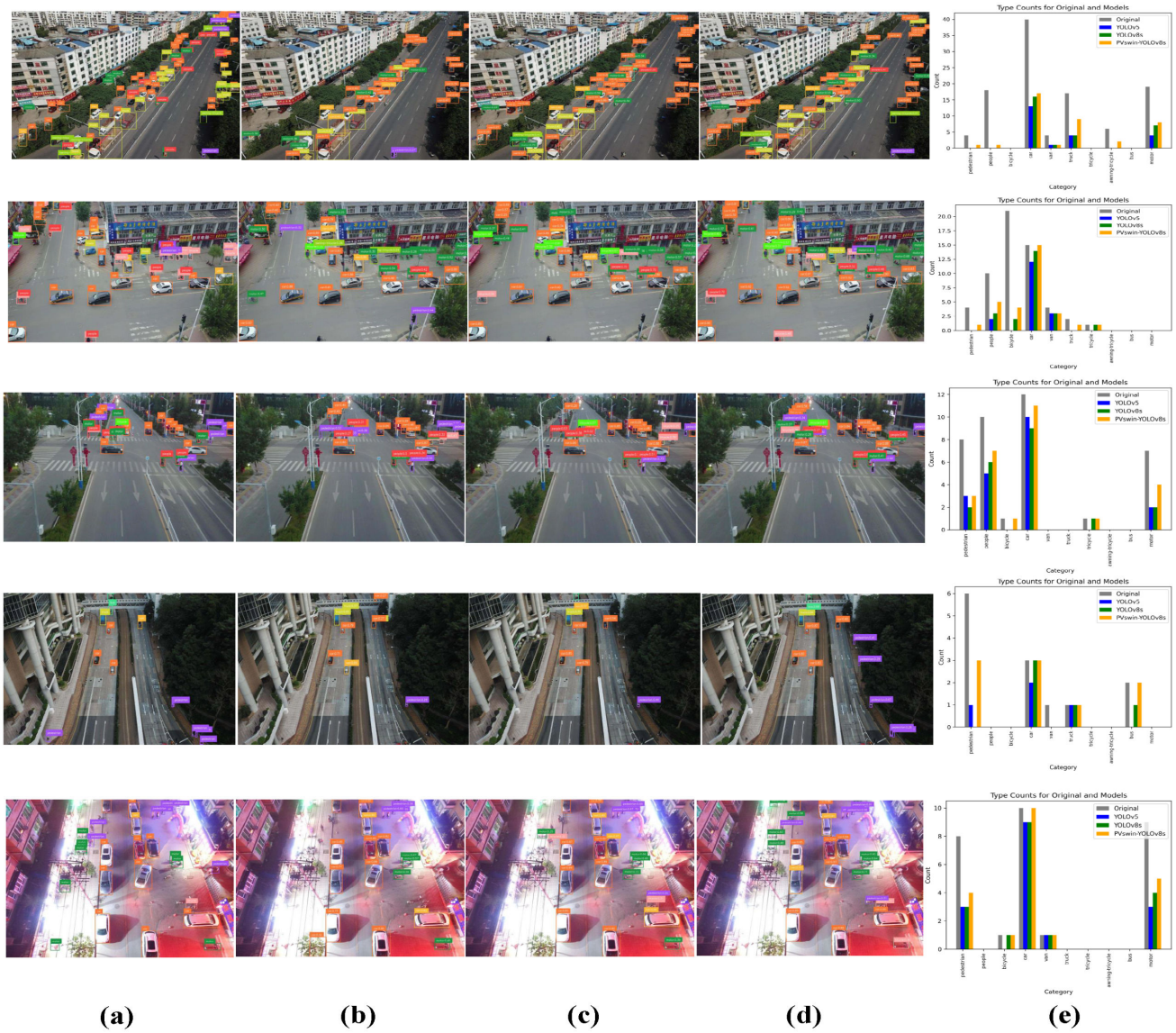


Figure 8. Comparative visualization of detection results on the Visdrone2019-test Dataset: (**a**) ground-truth annotations, (**b**) predictions by YOLOv5, (**c**) predictions by YOLOv8s, (**d**) predictions by the Enhanced PVswin-YOLOv8s Model, and (**e**) comprehensive detection performance chart. Here Count shows number of objects.

4.5. Discussion

The experimental results highlight PVswin-YOLOv8s's superior detection performance compared to YOLOv8s. Our proposed enhancements involve optimizing the backbone and neck parts of the network. PVswin-YOLOv8s integrates YOLOv8s, Swin Transformer, CBAM, and soft-NMS for robust UAV-based pedestrian and vehicle detection. The strategic placement by replacing the last C2f layer in the backbone network of YOLOv8s with a Swin Transformer block in the low-resolution feature maps of YOLOv8s addresses its limitations and then captures long-distance dependencies and diverse local information. This ensures a balance between global and local features. Secondly, CBAM in the neck network enhances feature information, leveraging its dual-channel nature, as depicted in Figure 3. Finally, soft-NMS was introduced as a replacement for NMS. Soft-NMS can be used to solve the issue of several objects overlapping in Visdrone2019 images, preventing targets from being missed. The fusion of YOLOv8s, Swin Transformer, CBAM, and soft-NMS preserves individual strengths, culminating in improved detection capabilities. In this study, we have evaluated our PVswin-YOLOv8s model in the following ways:

1. We compared our model with the baseline model YOLOv8s. Table 3 shows the mAP values for each class, as well as the mAP0.5 values for all classes. As can be seen, with the proposed PVswin-YOLOv8s model, detection accuracy improved by 4.8% compared to the baseline model on the Visdrone2019 dataset;
2. We compared our model against four versions of YOLOV8, specifically, YOLOv8n, YOLOv8s, YOLOv8m, and YOLOv8l. The results are presented in Table 4 using the VisDrone2019-val dataset and in Table 5 using the test set. PVswin-YOLOv8s outperforms YOLOv8n, YOLOv8s, and YOLOv8m, exhibiting the highest values for precision and mAP0.5. Additionally, its F1-score is comparable to YOLOv8l, with mAP0.5:0.95 only 0.1% lower than YOLOv8l based on the results in Table 4. AS can be seen in Table 5, PVswin-YOLOv8s's precision, F1 score, and mAP0.5 are superior to YOLOv8l, indicating superior detection performance despite a smaller model size. The experimental findings underscore the effective enhancement of detection accuracy for extremely small objects in our proposed structure;
3. We compared our model with previous versions of YOLO, specifically, YOLOv3-tiny, YOLOv5, YOLOv6, and YOLOv7. Here, our model also outperformed these previous version of YOLO, as shown in Table 6;
4. We carried out experiments to evaluate the performance of our proposed model against classical start-of-the-art (SOTA) models, namely, Faster-RCNN, Cascade R-CNN, RetinaNet, and CenterNet. The results are presented in Table 7. Based on these findings, our model outperforms these SOTA models in terms of detection performance;
5. We performed an ablation test that demonstrated that the integration of the Swin Transformer led to a 2.1% increase in mAP0.5 and a 1.7% increase in mAP0.5:0.95, highlighting the model's improved detection of small objects in complex scenes as shown in Table 8. The CBAM module contributed an additional 1.3% to mAP0.5 and 0.9% to mAP0.5:0.95, while Soft-NMS further improved these metrics by 1.1% and 0.8%, respectively, ensuring robust detection in occluded environments.

Following the quantitative assessments, we carried out a comprehensive visualization analysis, scrutinizing key evaluation factors such as precision, recall, and F1-score. Additionally, a thorough examination of the confusion matrix and detection results was carried out for both the baseline YOLOv8s and our enhanced PVswin-YOLOv8s model. At the same time, despite the commendable performance of our model surpassing all discussed counterparts, it is essential to acknowledge that there is room for improvement. Notably, the detection accuracy for extremely small objects, such as bicycles and tricycles, remains a challenge. This observation prompts our commitment to the next phase of major research, where the primary focus will be on further optimizing the model's accuracy for detecting diminutive objects. This

optimization endeavor is intricately balanced with the consideration of resource consumption, ensuring a holistic approach to model enhancement.

5. Conclusions

In smart cities, challenges related to pedestrian and vehicle detection, including extremely small items, occluded objects, and complicated backgrounds, remain for UAVs used in traffic congestion detection. Most of the existing algorithms suffer from low detection accuracy. To optimize the detection performance of the model, this research proposed an optimized pedestrian and vehicle detection model by integrating YOLOv8s, Swin Transformer, CBAM model, and soft-NMS, called PVswin-YOLOv8s. Firstly, the backbone network of YOLOv8s incorporates the Swin Transformer model by replacing the last C2f layer for global feature extraction for extremely small object detection. Secondly, to address the challenge of missed detections, we opt to integrate the CBAM into the neck of the YOLOv8s network. This inclusion involves leveraging both the channel attention module and spatial attention module, which prove effective in extracting feature information flow within the network. Finally, to avoid missing targets, Soft-NMS is used to resolve the problem of several objects overlapping in detection images. When applied to the VisDrone2019-val dataset, the proposed network decreased the miss identification of occluded overlapped and extremely small items, and showed an improved detection accuracy of 4.8% over the baseline YOLOv8s model. The proposed PVswin-YOLOv8s model also outperformed some previous versions of YOLO, four sizes of YOLOv8, and some classical models in terms of detection accuracy. However, there is room for improvement in terms of identifying extremely small vehicles and pedestrians. When it comes to detecting extremely small objects such as bicycles, the improved model is still not as accurate as it could be. Improving model detection accuracy with consideration for resource consumption will be the main focus of future studies.

Author Contributions: Conceptualization, N.U.A.T., Z.L., Z.Z., and M.A.; Methodology, N.U.A.T., Z.L., Z.Z., and M.A.; Software, N.U.A.T. and Z.L.; Validation, Z.Z. and M.A.E.; Formal analysis, N.U.A.T., Z.L., M.A., and M.A.E.; Investigation, M.A. and M.A.E.; Resources, M.A.E.; Writing—original draft, N.U.A.T.; Writing—review and editing, Z.L., Z.Z., M.A., and M.A.E.; Supervision, Z.Z.; Funding acquisition, M.A.E. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by EIAS Data Science Lab, CCIS, Prince Sultan University.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to thank Prince Sultan University for paying the APC of this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Nie, X.; Peng, J.; Wu, Y.; Gupta, B.B.; Abd El-Latif, A.A. Real-time traffic speed estimation for smart cities with spatial temporal data: A gated graph attention network approach. *Big Data Res.* **2022**, *28*, 100313. [CrossRef]
2. Iftikhar, S.; Asim, M.; Zhang, Z.; Muthanna, A.; Chen, J.; El-Affendi, M.; Sedik, A.; Abd El-Latif, A.A. Target Detection and Recognition for Traffic Congestion in Smart Cities Using Deep Learning-Enabled UAVs: A Review and Analysis. *Appl. Sci.* **2023**, *13*, 3995. [CrossRef]
3. Hazarika, A.; Poddar, S.; Nasralla, M.M.; Rahaman, H. Area and energy efficient shift and accumulator unit for object detection in IoT applications. *Alex. Eng. J.* **2022**, *61*, 795–809. [CrossRef]
4. Waheed, S.R.; Suaib, N.M.; Rahim, M.S.M.; Khan, A.R.; Bahaj, S.A.; Saba, T. Synergistic Integration of Transfer Learning and Deep Learning for Enhanced Object Detection in Digital Images. *IEEE Access* **2024**, *12*, 13525–13536. [CrossRef]
5. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9627–9636.
6. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229.

7. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
8. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
9. Terven, J.; Cordova-Esparza, D. A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. *arXiv* **2023**, arXiv:2304.00501.
10. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
11. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6569–6578.
12. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
13. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
14. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. . [CrossRef] [PubMed]
15. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
16. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
17. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
18. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
19. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in vision: A survey. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 1–41. [CrossRef]
20. Li, Y.; Mao, H.; Girshick, R.; He, K. Exploring plain vision transformer backbones for object detection. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2022; pp. 280–296.
21. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
22. Hatamizadeh, A.; Nath, V.; Tang, Y.; Yang, D.; Roth, H.R.; Xu, D. Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In Proceedings of the International MICCAI Brainlesion Workshop, Singapore, 18 September 2021; pp. 272–284.
23. Liu, Z.; Tan, Y.; He, Q.; Xiao, Y. SwinNet: Swin transformer drives edge-aware RGB-D and RGB-T salient object detection. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 4486–4497. [CrossRef]
24. Jannat, F.E.; Willis, A.R. Improving classification of remotely sensed images with the swin transformer. In Proceedings of the SoutheastCon 2022, Mobile, AL, USA, 26 March–3 April 2022; pp. 611–618.
25. Srinivas, A.; Lin, T.Y.; Parmar, N.; Shlens, J.; Abbeel, P.; Vaswani, A. Bottleneck transformers for visual recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 16519–16529.
26. Zhao, H.; Jia, J.; Koltun, V. Exploring self-attention for image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10076–10085.
27. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
28. Naseer, M.M.; Ranasinghe, K.; Khan, S.H.; Hayat, M.; Shahbaz Khan, F.; Yang, M.H. Intriguing properties of vision transformers. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 23296–23308.
29. Mittal, P.; Sharma, A.; Singh, R.; Dhull, V. Dilated convolution based RCNN using feature fusion for Low-Altitude aerial objects. *Expert Syst. Appl.* **2022**, *199*, 117106. [CrossRef]
30. Liu, Z.; Qiu, S.; Chen, M.; Han, D.; Qi, T.; Li, Q.; Lu, Y. CCH-YOLOX: Improved YOLOX for Challenging Vehicle Detection from UAV Images. In Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 18–23 June 2023; pp. 1–9.
31. Deng, L.; Bi, L.; Li, H.; Chen, H.; Duan, X.; Lou, H.; Zhang, H.; Bi, J.; Liu, H. Lightweight aerial image object detection algorithm based on improved YOLOv5s. *Sci. Rep.* **2023**, *13*, 7817. [CrossRef] [PubMed]
32. Lou, H.; Duan, X.; Guo, J.; Liu, H.; Gu, J.; Bi, L.; Chen, H. DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor. *Electronics* **2023**, *12*, 2323. [CrossRef]

33. Hui, Y.; Wang, J.; Li, B. STF-YOLO: A small target detection algorithm for UAV remote sensing images based on improved SwinTransformer and class weighted classification decoupling head. *Measurement* **2024**, *224*, 113936. [CrossRef]
34. Tang, S.; Fang, Y.; Zhang, S. HIC-YOLOv5: Improved YOLOv5 for Small Object Detection. *arXiv* **2023**, arXiv:2309.16393.
35. Sirisha, M.; Sudha, S.V. An Advanced Object Detection Framework for UAV Imagery Utilizing Transformer-Based Architecture and Split Attention Module: PvSAMNet. *Trait. Signal* **2023**, *40*, 1661. [CrossRef]
36. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 20 December 2023).
37. Wang, C.Y.; Liao, H.Y.M.; Yeh, I.H. Designing network design strategies through gradient path analysis. *arXiv* **2022**, arXiv:2211.04800.
38. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
39. Cao, Y.; Chen, K.; Loy, C.C.; Lin, D. Prime sample attention in object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11583–11591.
40. Li, X.; Lv, C.; Wang, W.; Li, G.; Yang, L.; Yang, J. Generalized focal loss: Towards efficient representation learning for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 3139–3153. [CrossRef]
41. Ju, R.Y.; Cai, W. Fracture Detection in Pediatric Wrist Trauma X-ray Images Using YOLOv8 Algorithm. *arXiv* **2023**, arXiv:2304.05071.
42. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. Tood: Task-aligned one-stage object detection. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 3490–3499.
43. Oksuz, K.; Cam, B.C.; Kalkan, S.; Akbas, E. Imbalance problems in object detection: A review. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3388–3415. [CrossRef] [PubMed]
44. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017, pp. 6000–6010.
45. Bodla, N.; Singh, B.; Chellappa, R.; Davis, L.S. Soft-NMS–improving object detection with one line of code. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5561–5569.
46. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. Detection and tracking meet drones challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 7380–7399. [CrossRef] [PubMed]
47. Powers, D.M. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv* **2020**, arXiv:2010.16061.
48. Boyd, K.; Eng, K.H.; Page, C.D. Area under the precision-recall curve: Point estimates and confidence intervals. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, 23–27 September 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 451–466.
49. Adarsh, P.; Rathi, P.; Kumar, M. YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; pp. 687–694.
50. Ultralytics. YOLOv5: A State-of-the-Art Real-Time Object Detection System. 2021. Available online: https://docs.ultralytics.com (accessed on 19 December 2023).
51. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
52. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
53. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
54. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.