electronics

MDPI

Article

YOLOv8-CGRNet: A Lightweight Object Detection Network Leveraging Context Guidance and Deep Residual Learning

Yixing Niu, Wansheng Cheng, Chunni Shi and Song Fan *

School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114051, China; niuyixing@ustl.edu.cn (Y.N.); cws@ustl.edu.cn (W.C.); shichunni@ustl.edu.cn (C.S.)
* Correspondence: fansong@ustl.edu.cn

Abstract: The growing need for effective object detection models on mobile devices makes it essential to design models that are both accurate and have fewer parameters. In this paper, we introduce a YOLOv8 Res2Net Extended Network (YOLOv8-CGRNet) approach that achieves enhanced precision under standards suitable for lightweight mobile devices. Firstly, we merge YOLOv8 with the Context GuidedNet (CGNet) and Residual Network with multiple branches (Res2Net) structures, augmenting the model's ability to learn deep Res2Net features without adding to its complexity or computational demands. CGNet effectively captures local features and contextual surroundings, utilizing spatial dependencies and context information to improve accuracy. By reducing the number of parameters and saving on memory usage, it adheres to a 'deep yet slim' principle, lessening channel numbers between stages. Secondly, we explore an improved pyramid network (FPN) combination and employ the Stage Partial Spatial Pyramid Pooling Fast (SimPPFCSPC) structure to further strengthen the network's capability in processing the FPN. Using a dynamic non-monotonic focusing mechanism (FM) gradient gain distribution strategy based on Wise-IoU (WIoU) in an anchor-free context, this method effectively manages low-quality examples. It enhances the overall performance of the detector. Thirdly, we introduce Unifying Object Detection Heads with Attention, adapting to various input scenarios and increasing the model's flexibility. Experimental datasets include the commonly used detection datasets: VOC2007, VOC2012, and VisDrone. The experimental results demonstrate a 4.3% improvement in detection performance by the proposed framework, affirming superior performance over the original YOLOv8 model in terms of accuracy and robustness and providing insights for future practical applications.

Keywords: object detection; YOLO; deep learning

Citation: Niu, Y.; Cheng, W.; Shi, C.; Fan, S. YOLOv8-CGRNet: A Lightweight Object Detection Network Leveraging Context Guidance and Deep Residual Learning. *Electronics* **2024**, *13*, 43. https://doi.org/10.3390/electronics13010043

Academic Editor: Stefanos Kollias

Received: 16 November 2023
Revised: 12 December 2023
Accepted: 18 December 2023
Published: 20 December 2023

1. Introduction

With the evolution of computer technology and the widespread application of principles of computer vision, research on target detection and tracking using computer image processing technology is gaining popularity. Object detection [1] plays a pivotal role in applications such as autonomous driving and unmanned vehicles [2], security and surveillance [3], medical imaging [4], robotics [5], and agriculture [6], with image segmentation [7] and object tracking [8]. Pedestrian re-identification [9] often relies on it. Object detection typically involves two primary steps: locating an object's position and classifying its type. Early methods employed the sliding window technique, sliding windows of various sizes across the image and running a classifier at each window position to identify targets. With technological advancements, region-based Convolutional Neural Networks (R-CNN) [10] gained popularity, initially using selective search to extract candidate regions and then classifying these regions using CNNs. The introduction of a faster r-cnn [11] containing a region suggestion network (RPN) [11], which automatically suggests areas of an image that may contain objects, can improve efficiency. However, YOLO [12] further streamlined the process by treating object detection as a regression problem, predicting bounding boxes

and class scores for all categories simultaneously, thereby circumventing multiple forward propagations. These advancements have facilitated faster and more accurate real-world applications of object detection.

Object detection methods can be combined with evaluation algorithms. Deep architectures with a region proposal network (DeepRPN-BIQA) [13] proposed a deep architecture incorporating a region proposal network (RPN) for blind image quality assessment (BIQA) of natural-scene and screen-content images. The RPN extracts important regions that affect image quality by computing visual saliency. These regions are then fed into a Convolutional Neural Network (CNN) to predict the quality score. Object detection can be enhanced by integrating it with the DeepRPN-BIQA approach proposed in this paper, thereby improving the performance and efficiency of object detection. For instance, DeepRPN-BIQA could be used as a preprocessing step to perform quality assessment on the input image. Based on the quality score, the best image can be selected, or image enhancement can be performed before feeding it into the object detection model. Alternatively, DeepRPN-BIQA could serve as a postprocessing step to assess the quality of the output results from the object detection model. The best detection results can be selected, or result optimization can be performed based on the quality score.

Object detection can be integrated with other algorithms. In the fields of pedestrian re-recognition [14] and computer vision in medical image analysis [15], object detection is used in smart video surveillance systems to provide deeper insights and automated responses. In robotics, object detection is combined with path planning [16] and obstacle avoidance algorithms [17], aiding robots in better navigation and interaction with their environment. In multimodal learning systems, object detection can be integrated with NLP [18] (natural language processing) technologies to process complex data containing both visual and textual information, such as extracting information from social media posts.

Despite significant progress in identifying and locating objects in images, challenges remain concerning computational resources, small object detection, and real-time requirements. The rapid increase in demand for object detection models on mobile devices faces the limitations of processors and GPUs, which are generally less powerful than those on desktops or servers. This limitation affects the complexity of models that can run on devices in real time. Moreover, the typically lower RAM on mobile devices implies stringent constraints on model size and runtime memory usage. Thus, designing models that are both memory-efficient and highly accurate is imperative and challenging. Although improved versions like YOLOv8 provide higher accuracy and detection outcomes, considering the model's generalization and adaptability to ever-changing application scenarios and dataset characteristics remains crucial for ensuring stability and efficacy under diverse conditions.

Inspired by these considerations, this research introduces a framework for object detection, the YOLOv8 Res2Net Extended Network (YOLOv8-CGRNet). YOLOv8 represents a popular model in the object detection domain, exemplifying one of the latest advancements in the YOLO [12] series, showcasing performance on par or better than other models in public datasets while maintaining rapid inference speed. In its latest iteration, an attempt to amalgamate the backbone network of the YOLO series with the Residual Network with multiple branches (Res2Net) [19] structure has been made, introducing the learning capability of ResNet's deeper features for object detection tasks, thereby enhancing the model's recognition of complex scenes and multi-scale objects. The integration of the downsampling module from Context GuidedNet (CGNet) [20] has fortified the learning of local contextual information, offering improved performance in detail and context understanding.

YOLOv8-CGRNet retains the downsampling module from CGNet while incorporating efficient modules like Stage Partial Spatial Pyramid Pooling Fast (SimPPFCSPC) [21] and Wise-IoU (WIoU) [22]. It merges various modules such as skip-path, Fused Convolution, and CIOU loss to optimize model performance and enhance recognition, further refining the model's feature extraction and representation capabilities. With CGblock downsampling and deep Res2block structures alternating within the network, they work collaboratively,

ensuring that while deep features are extracted, context information is effectively preserved and utilized, thereby enhancing performance on object detection tasks.

The experimental datasets comprise two widely utilized object detection sets, Pascal VOC2007 [23], VOC2012 [24], and VisDrone [25], provided by the Visual Geometry Group (VGG) of the University of Oxford, including 20 object categories encompassing everyday items from vehicles to domestic goods, animals to humans. VOC2007 contains approximately 9600 images with about 27,000 object annotations. VOC2012 includes around 11,500 images with roughly 35,000 object annotations. Pascal VOC sets the standard evaluation metrics for various tasks, employing mean Average Precision at IoU threshold 0.5 (mAP50) [26] for object detection assessment. The contributions of this research are outlined as follows: First, an exploration of the combination of the YOLOv8 series with Context GuidedNet and the Res2Net architecture has been conducted, which leverages the depth feature learning capabilities of Res2Net. This fusion facilitates enhanced object detection against complex backgrounds and a wide range of object scales, delivering superior recognition capabilities. Second, the incorporation of the SimPPFCSPC structure further augments the network's proficiency in handling the Feature Pyramid Network (FPN) [27]. This model employs FPN and integrates contextually enhanced feature extraction techniques with adaptive strategies for depth and width tailored to varying computational and performance demands. Third, this network amalgamates the advantages of various modules and, notably, introduces the novel Detect DyHead (dynamic head) [28], which possesses the aptitude to adapt to different input scenarios, significantly increasing the model's versatility. To further advance detection effectiveness, a clever mechanism utilizing dynamic non-monotonic focusing mechanism (FM) [22] gradient boosting allocation has been adopted. This mechanism effectively processes low-quality examples and elevates the overall performance of the detector.

As demonstrated in Figure 1, through a series of innovations and enhancements, the proposed model manifests substantial improvements over the baseline YOLOv8, especially in intricate scenarios and multi-scale object detection tasks. These modifications not only significantly boost the model's performance but also manage to maintain a balance between the number of parameters and inference speed. Testing across multiple datasets has shown that, in comparison to the original YOLOv8, our model achieves an approximate 4% increase in accuracy.

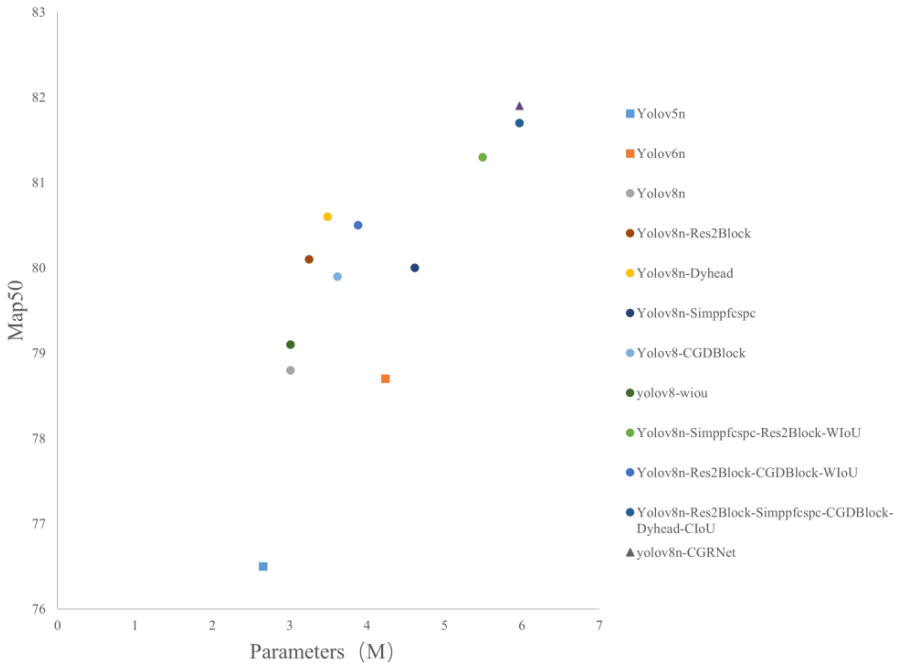

Figure 1. Comparison between the number of parameters (M) and mAP50.

The remainder of this document is organized as follows: Section 2 delineates the methodology, Section 3 discusses the experimental results, and Section 4 concludes by summarizing the findings and suggesting potential avenues for future research. Section 5 deliberates on the model's strengths, weaknesses, and application discussions.

2. Materials and Methods

The framework utilized in this study is depicted in Figure 2. A noted limitation of lightweight models is the potential compromise between model accuracy and generalization capability. Hence, it is challenging to strike an appropriate balance between performance, speed, and accuracy when designing and selecting lightweight models.

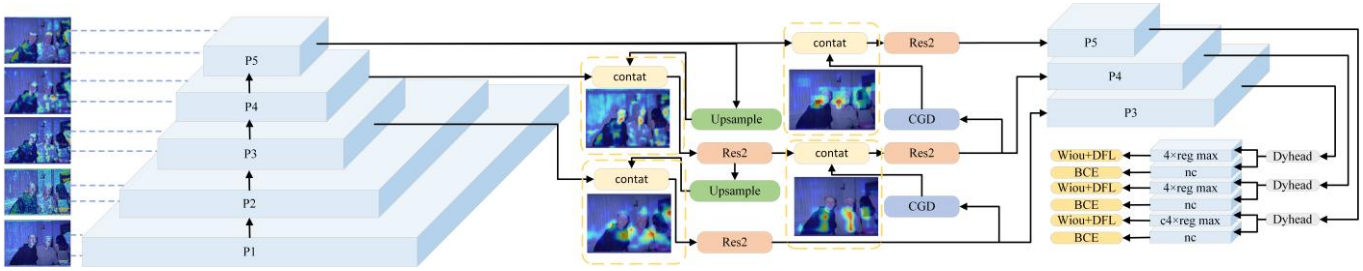

Figure 2. Overall framework.

As shown in Figure 2, the model employs a Feature Pyramid Network (FPN) [27]. Initially, the integration of a Context-Guided Block (CG) is undertaken: The Context GuidedNet (CGNet) [20] efficiently captures local features and the surrounding context, leveraging spatial dependencies and contextual information to enhance accuracy. It reduces the number of parameters and saves on memory usage, adhering to the "deep and thin" principle to diminish channel count across stages. Secondly, the fusion of a Res2Net block is implemented, constructing hierarchical-like residual connections within a single residual block. This approach augments multi-scale representation at a finer granularity and expands the receptive field for each network layer. Thirdly, the employment of a SimPPFCSPC structure further amplifies the network's capacity to process the feature pyramid effectively. Fourthly, the integration of a dynamic attention mechanism from DyHead, which encompasses scale awareness, spatial awareness, and task awareness, is reported. By dynamically combining information from multiple scales, the scale-aware attention mechanism enhances the ability to recognize things of various sizes. In addition to attending to attention at each spatial location, the spatial-awareness attention module adaptively combines various feature levels in order to acquire a representation that is more discriminative. In order to adaptably handle a variety of tasks, including classification, box regression, and center-keypoint learning, task awareness distributes attention across many channels.

2.1. Overall Structure

As depicted in Figure 3, the network's head starts with a Conv layer for initial feature extraction. The Conv part includes a 2D convolutional layer, 2D batch normalization, and a Sigmoid activation function. The CGD Block and Res2Block modules then work in concert to capture and enhance multi-scale features. The integration of SimPPFCSPC is finalized to achieve enhanced performance with fewer parameters. At the detection head, a decoupled head structure prevalent in current research is utilized, separating classification and detection tasks using an anchor-free approach. Initially, the dynamic head (DyHead) is employed for unifying scale, spatial, and task awareness within the detection head. Subsequently, the features extracted from the DyHead are refined and augmented through successive convolutional operations. After feature extraction by the Conv layer, a two-dimensional convolution operation generates an output of appropriate size. The difference between the predicted bounding boxes is quantified by the Bounding Box Loss,

and the discrepancy between the predicted class probabilities and the actual class labels is measured by the Class Loss.

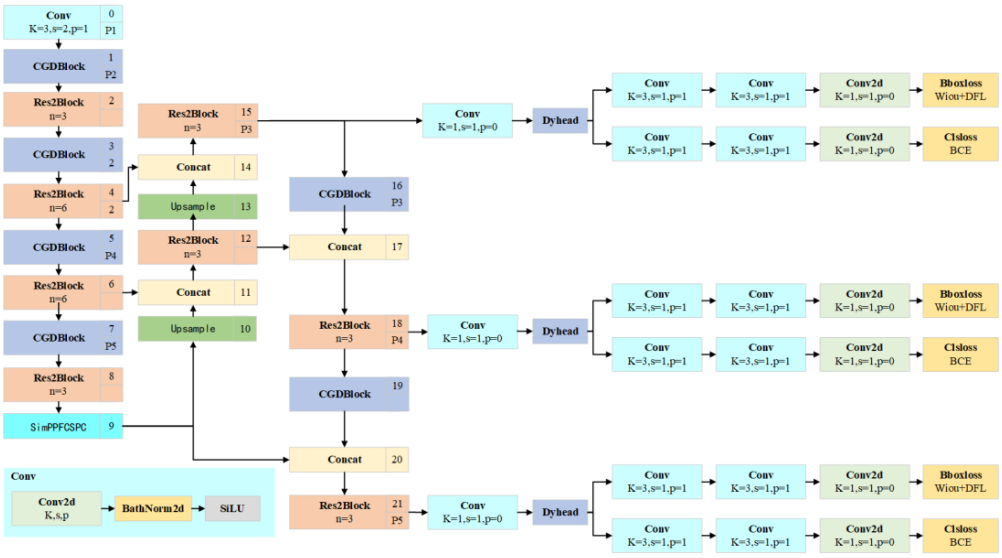

Figure 3. The specific network architecture characterized. In the image, "k" stands for Kernel size. "s" stands for Stride. "p" stands for Padding. Depth: 0.33, width: 0.25; and MaxChannels: 1024.

FPN serves as a feature fusion mechanism that enhances the detection of multi-scale targets with minimal computational cost. The main issue addressed by the FPN is the need for more handling of multi-scale variations in object detection. The downsampling ratios of the CGD blocks, numbered (1, 3, 5, and 7), are {4, 8, 16, and 32}, respectively, with each subsequent layer outputting a smaller feature map than the preceding one. These feature maps are employed within the FPN to construct a richer multi-scale feature representation. The top-down pathway begins at the last (and most profound) layer of the network, where feature maps are upsampled to increase resolution. Each upsampled feature map is laterally connected with the corresponding resolution feature map from the bottom-up pathway to merge high-level semantic information with low-level detail information. This process continues until it reaches the original image resolution. Anchor-free object detection provides an efficient alternative to tackle the complexity and limitations associated with traditional anchor-based methods by predicting key points or bounding boxes of objects directly on feature maps, simplifying the detection process while enhancing flexibility and efficiency. Center-based methods identify the center point and dimensions of each object. This framework predicts the distances to the left, top, right, and bottom edges of the target box from its center point.

2.2. Main Blocks

As shown in Figure 4, it is mainly divided into three blocks. First, the CGDBlock learns the joint features of local features and surrounding context and further improves the joint features with global context while maintaining low memory usage and improving accuracy. Secondly, Res2block can effectively utilize the hierarchical residual connection within a residual block to improve the multi-scale feature representation ability and thus improve the performance of various visual tasks. Finally, SimPPFCSPC is used to extract and process features at different scales to adapt to various tasks and data.

Res2block [19] processes detection in a multi-scale manner, which helps to extract global and local information. To better integrate information from different scales, we split the input features into 4 features and passed them through a 1×1 convolution. Each subset is processed by a 3×3 convolution group to obtain the output feature map. To reduce the number of parameters, the first split convolution is omitted, which can also be viewed as a form of feature reuse. A hierarchical residual connection is established between each

convolution group, $y_i = c_i(y_{i-1}) + y_i$, $y_0 = x_1$. Finally, all the output feature maps are concatenated together to form the final output feature map.

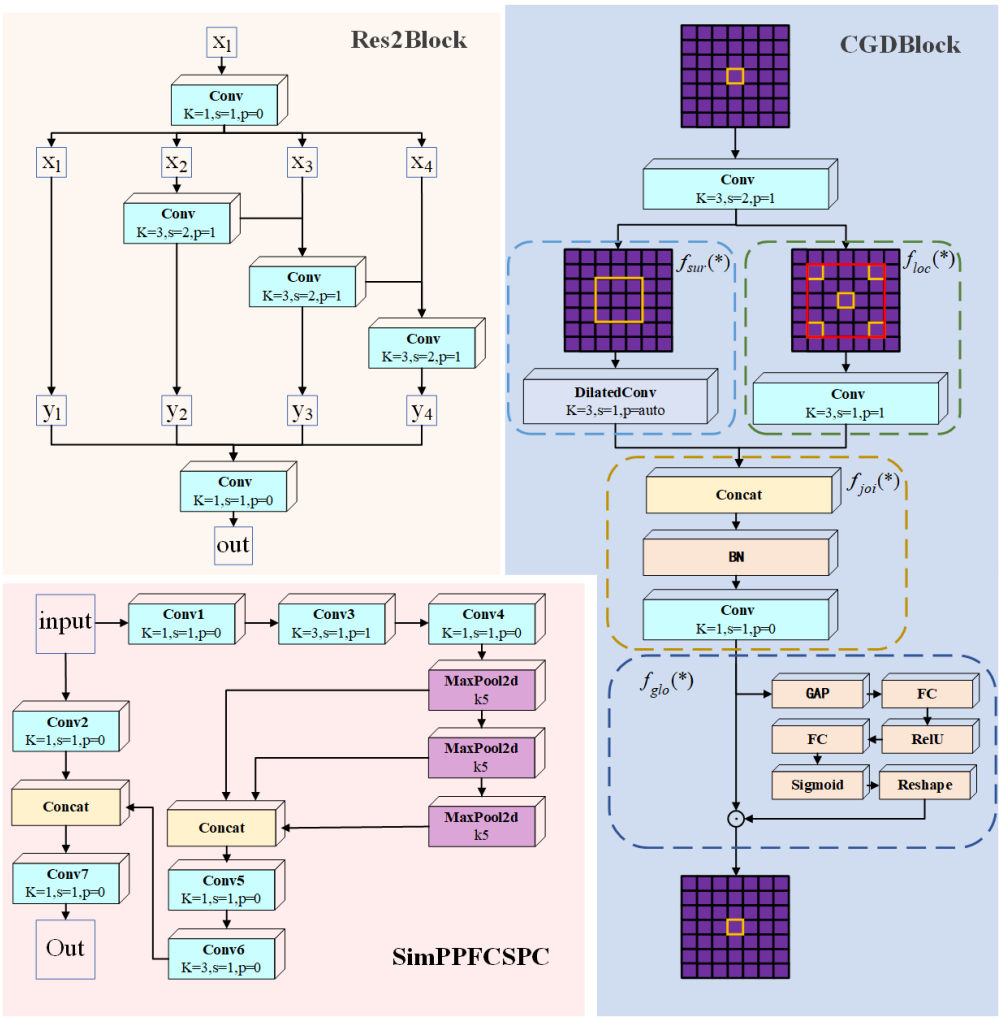

Figure 4. The main blocks of the backbone network and the neck network.

The CGDblock for downsampling and the simulation of spatial dependencies and semantic context information are improved by CGNet [20]. It consists of four sub-modules: local feature extractor $f_{loc}()$, surrounding context extractor fsur(), joint feature extractor $f_{joi}()$, and global context extractor $f_{glo}()$. It first learns the joint features of the local features and surrounding context and then uses the global context for the channel-level weighting of joint features. It also adopts residual learning to enhance information flow.

Downsampling: the conv1x1 layer initially reduces the spatial dimension of the input by half and adjusts the number of channels.

Feature integration: Local (*F_loc*) and surrounding (*F_sur*) features are connected and further processed to effectively merge these features. Then, the spatial dimension of the input is reduced by half through downsampling; the conv1x1 layer and the number of channels are adjusted. Finally, the F_glo layer is used to refine these combined features.

$$f^*_{joi} = f_{joi}(f^*_{loc}, f^*_{sur}) = BN(PReLU([f^*_{loc}, f^*_{sur}])) \quad (1)$$

Herein, f^*_{loc} and f^*_{sur}, respectively, represent local features and surrounding context features, and f^*_{joi} represents joint features. $[f^*_{loc}, f^*_{sur}]$ represents the connection operation be-

tween local features and surrounding context features. PReLU represents the parameterized linear rectification unit, and BN represents batch normalization.

$$f^*_{glo} = f_{glo}(f^*_{joi}) = FC(FC(GAP(f^*_{joi}))) \tag{2}$$

This formula describes how to obtain global context features from joint features. Here, f^*_{joi} represents joint features, and f^*_{glo} represents global context features. GAP represents global average pooling, and FC represents a fully connected layer.

$$f^*_{out} = f^*_{glo} \odot f^*_{joi} \tag{3}$$

This formula describes how to weigh the joint features at the channel level with global context features to obtain the output features. Here, f^*_{glo} represents global context features, f^*_{joi} represents joint features, and f^*_{out} represents output features. $\odot$ represents element-wise multiplication. This design reflects an intention to balance detailed local feature extraction with more global contextual information, which is usually beneficial for object detection and for enhancing spatial hierarchy structures in visual deep learning models.

In SimPPFCSPC, different input features are first extracted through conv1, conv3, and conv4. Then, dimensionality reduction is performed through the max-pooling layer. Next, the dimensionality-reduced features are further processed by conv5 and conv6 to extract features. Meanwhile, the original input is also processed by conv2 to extract input features. Finally, the outputs of these two parts are concatenated together and mapped to the target space through conv7. Speed improvement is achieved while maintaining the same receptive field.

2.3. FusionDetect–ModuleHead

The design principle of the DyHead [28] is to combine multiple attention mechanisms, each of which focuses on a different dimension: scale, space, and task. The detailed design and working principles of scale-aware attention, spatial-aware attention, and task-aware attention.

By cooperatively combining multiple self-attention mechanisms between feature levels, spatial positions, and output channels, the proposed method significantly improves the representation ability of the object detection head without increasing any computational overhead. Formula (4) represents the general form of applying the self-attention mechanism to the feature tensor, where $\pi(\cdot)$ is an attention function.

$$W(F) = \pi(F) \cdot F \tag{4}$$

Formula (5) decomposes the attention function into three sequentially applied attention functions, each acting on different dimensions of the feature tensor.

$$W(F) = \pi_C(\pi_S(\pi_L(F)) \cdot F) \cdot F \cdot F \tag{5}$$

Formula (6) represents the scale-aware attention module, which uses a 1×1 convolutional layer to learn the relative importance of different level features and normalizes the weights using a hard Sigmoid function.

$$\pi_L(F) \cdot F = \sigma(f(\sum_{S,C} F)) \cdot F \tag{6}$$

Formula (7) represents the spatial-aware attention module, which uses deformable convolution to sparsely sample and aggregate features at different levels and spatial positions and adjusts the sampling position and importance using self-learned offsets and weights.

$$\pi_S(F) \cdot F = \frac{1}{L} \sum_{l=1}^{L} \sum_{k=1}^{K} w_{l,k} \cdot F(l; p_k + \Delta p_k; c) \cdot \Delta m_k \tag{7}$$

Formula (8) represents the task-aware attention module, which uses a dynamic ReLU function to activate features on different channels and learns the activation threshold using two fully connected layers and a normalization layer.

$$\pi_C(F) \cdot F = \max(\alpha_1(F) \cdot F_c + \beta_1(F), \alpha_2(F) \cdot F_c + \beta_2(F)) \quad (8)$$

The meanings of the symbols are as follows:

F is the input feature tensor with dimensions $L \times S \times C$, where L represents the number of levels, S represents the number of spatial positions, and C represents the number of channels.

W is the output feature tensor with the same dimensions as F.

π is the attention function, which can have different forms.

σ is the hard Sigmoid function, defined as σ(x) = max(0, min(1, (x + 1)/2)).

f is a linear function implemented by a 1×1 convolutional layer.

$w_{l,k}$ are the weight parameters of the deformable convolution learned from the input features.

p_k is the original sampling position, and Δp_k is the self-learned offset.

F (l; $p_k + \Delta p_k$; c) represents the feature value at the position $p_k + \Delta p_k$ on the l-th level and the c-th channel.

Δm_k is the self-learned importance scalar, learned from the input features.

α_1, α_2, β_1, and β_2 are the parameters of the dynamic ReLU function, learned from the super function θ(·). θ(·) consists of a global average pooling layer, two fully connected layers, and a normalization layer, and it applies a shifted Sigmoid function to normalize the output to the interval $[-1, 1]$.

Fc represents the feature slice on the c-th channel.

The proposed method solves the problem of improving the performance of various object detection heads without providing a unified perspective, which was attempted by previous works. By cooperatively combining multiple self-attention mechanisms between feature levels, spatial positions, and output channels, the representation ability of the object detection head is significantly improved. Although this method enhances the representation ability of the object detection head, it does not bring a large computational overhead.

3. Experiments

In this section, we demonstrate the superiority of YOLOv8-CGRNet by evaluating its effectiveness in recognizing voc2012 and voc2007 and show the improvement of each step in recognition performance through ablation experiments.

3.1. Introduction to the Dataset

As shown in Table 1, the research utilized the VOC2007 and VOC2012 datasets, released by the PASCAL (Pattern Analysis, Statistical Modeling, and Computational Learning) Network Organisation, funded by the European Union. The VOC2007 dataset comprises 9963 images taken in diverse environments, including both indoor and outdoor scenes. Annotations for the 24,640 objects depicted within these images are provided, with each object delineated by a bounding box. The dataset encompasses 20 categories, such as persons, animals, vehicles, and household furniture. The VOC2012 dataset includes 17,125 images, with a total of 27,450 annotated objects, and features similar environmental conditions and object annotations as VOC2007.

VisDrone was collected by the AISKYEYE team at the Lab of Machine Learning and Data Mining, Tianjin University, China. It includes 288 video clips formed by 261,908 frames and 10,209 static images. The data were captured using various drone-mounted cameras. Each image has a corresponding annotation file that contains the location, category, and occlusion degree of each object.

Table 1. Dataset attributes.

Dataset	Images	Number of Class	Objects
Voc2012 [24]	11,540	20	27,450
Voc2007 [23]	9963	20	24,640
VisDrone [25]	10,209	10	2,645,719

3.2. Detailed Implementation

The full VOC2012 dataset, alongside the VOC2007 training and validation subsets, was employed for training, with evaluations conducted on the VOC2007 test set. The model was configured for mobile application compatibility, with depth and width multipliers set at 0.33 and 0.25, respectively, and a channel limit of 1024. A consistent training regime of 200 epochs was maintained across all experiments. Ablation studies indicated that, while certain frameworks may enhance efficiency individually, their performance can diminish when integrated. Various fusion methods were trialed before establishing a synergy of YOLOv8 with Context GuidedNet, Res2Net, SimPPFCSPC, DyHead, and WIoU structures that yielded improved compatibility. An NVIDIA 3080 GPU with 10 GB of memory facilitated the computational process.

The first step entailed configuring the baseline model, YOLOv8n, optimized for mobile devices, and training it on the combined VOC datasets for 200 epochs. Evaluating this model on the VOC2007 test set provided a benchmark for subsequent refinements. In the second step, Res2Net structure integration augmented the baseline model's expressive capacity for feature representation. This modified model underwent identical training processes, validating Res2Net's contributions through performance comparisons pre- and post-integration. Upon confirming Res2Net's effectiveness, the third step incorporated the SimPPFCSPC structure to enhance complex feature learning capabilities. The fourth step involved integrating Context GuidedNet and DyHead structures to bolster contextual comprehension and detail resolution in target detection, respectively. The inclusion of the WIoU structure aimed to refine the model's locational accuracy, culminating in a comprehensive model fusion. Each integration step's impact was meticulously documented through key performance metrics such as detection precision, model size, and operational speed, ensuring a thorough multidimensional performance evaluation. The experimental methodology prioritized repeatability and the stability of structural combinations. This progression of experimental steps aimed to provide researchers with a definitive framework for assessing the specific impact of various architectures on model performance. Comparisons were also drawn with other models, including YOLOv5 and YOLOv6 variants scaled to equivalent depth and width, with a channel limit of 1024.

3.3. Experimental Results

In the PASCAL VOC dataset, object categories exhibit distinct shapes, sizes, and contextual variances. For instance, the "cat" category tends to occupy larger image areas with rich texture information, facilitating effective learning and recognition by the model. Conversely, objects such as "bottles" and "plants" are typically smaller and more challenging to discern against complex backgrounds, highlighting the significance of structural enhancements for such categories.

As demonstrated by Table 2, Fast R-CNN reported an overall mean Average Precision at the IoU threshold of 0.5 (mAP50) [26] of 68.4%, with exceptional performance in the "aeroplane" category at 82.3%. Faster R-CNN achieved a higher overall mAP50 of 70.4%, with an "aeroplane" category precision of 84.9%. YOLO models registered lower performance in detecting "bottle" category objects, with YOLOv3-tiny achieving a notable mAP of 72.3% and displaying superior performance in the "motorbike" and "train" categories. The YOLOv5n and YOLOv6n iterations exhibited consistent improvements across all categories, with mAP50s of 76.5% and 78.7%, respectively.

Table 2. The mAP50 for the PASCAL VOC2007 test dataset, utilizing a union of VOC2007 and VOC2012 trainval data. Herein, "R" stands for Res2Net, "C" for CGNet, "S" for SimPPFCSPC, "D" for DyHead, and "W" for WIoU. In the table, the bolded values represent the highest mAP among these methods.

Method	All	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	Tv
Fast [11]	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73	55	87.5	80.5	80.8	72	35.1	68.3	65.7	80.4	64.2
Faster [29]	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
YOLO [12]	57.9	77	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
SSD300 [30]	72.4	85.6	80.1	70.5	57.6	46.2	79.4	76.1	89.2	53	77	60.8	87	83.1	82.3	79.4	45.9	75.9	69.5	81.9	67.5
SSD512 [30]	74.9	87.4	82.3	75.8	59	52.6	81.7	81.5	**90**	55.4	79	59.8	88.4	84.3	84.7	83.3	50.2	78	66.3	86.3	72
YOLOv3-tiny	72.3	72.2	84.7	65.6	65.6	57.4	81.4	85.1	76.7	53.1	75.4	66.1	73.4	84.3	84.1	83.9	45.8	73.8	65	77.3	75
YOLOv5n	76.5	84.3	87.1	72.5	69.2	62.3	84.8	89.1	81.1	58.2	76.7	75.8	79.6	86.7	83.8	85.8	45.3	74.3	72.4	84	76.3
YOLOv6n	78.7	83.6	89.1	74.3	68.9	66.3	86.4	90.2	85.5	63.1	77.7	77.9	83.7	89.4	86.9	87	47.7	74.9	77.2	86.8	76.5
YOLOv8n	78.8	85.8	88.7	73.8	69.4	65.2	85.7	90.3	84.3	61.4	81.2	76.8	81	88.3	85.8	87.1	50.8	78.5	76.1	89.3	77.1
YOLOv8n-W	79.1	86.2	88.8	74.6	69.2	66.1	85.3	90.8	86	61.8	80.9	78.5	82.4	89.9	86.1	87.1	53.1	78	74.5	86.4	75.5
YOLOv8n-C	79.9	87	89.8	77.2	70.7	67.8	85.9	90.8	85.3	62.4	83.8	79.1	81.3	89.9	85.9	87.9	49.3	79.5	76	88.7	78.9
YOLOv8n-S	80	87.6	89.6	76.7	71.6	66.1	86.1	90.6	86.5	62.7	80.7	76.7	84.9	90.1	87.3	87.6	50.6	79	77.9	88.4	79.2
YOLOv8n-R	80.1	86.3	88.1	76.9	70.4	66.7	86.8	91.6	87.8	62.7	81.8	78.8	83.5	90.5	87.3	88	53.8	76.6	77.7	89.9	77.2
YOLOv8n-R-C-W	80.5	88.3	90	76.1	73.7	68	86.7	91.1	86.3	63.9	80.2	78.8	84.6	89.5	88.4	88.4	52.7	77.5	79	88	78.3
YOLOv8n-D	80.6	88.4	88.9	76.4	71.9	68.3	88.6	91.7	87.1	62.7	80.9	79.3	81.9	90.5	87	88.4	52.1	81.1	78.2	89.8	78.3
YOLOv8n-R-S-W	81.3	88.5	89.8	**78.6**	74	67.9	88.2	**92**	88.2	64.5	80.8	**81.6**	86	91.3	**89**	88.7	52.2	79.3	78.2	88.5	78.7
YOLOv8n-R-S-C-D	81.7	89	89.9	77.5	75.7	**69.8**	**89.7**	91.9	88.6	65.8	80.7	77.9	85.3	90.5	87.8	89.3	52.4	**79.7**	**81.5**	**90.9**	80.3
YOLOv8- CGRNet	**81.9**	**89.3**	**89.9**	77.3	**76.2**	68.7	89	91.7	89.1	**66.2**	**81.4**	79	**86.5**	**91.5**	88	**89.1**	**53.8**	78.7	80.8	90.2	**80.9**

The YOLOv8n's enhanced understanding of small objects and large scene contexts translated to over 70% mAP50 across multiple categories, attesting to its robustness as a general-purpose object detection model. The performance in categories such as "aeroplane", "bicycle", "boat", and "bottle" was particularly notable, likely due to the optimized deep learning architecture and effective training methodologies.

Through multi-scale feature extraction, the Res2Net structure provided the model with refined feature representations, as evidenced by improved performance in the "vehicle" and "aeroplane" categories. This scalability and the hierarchical connection approach of Res2Net may also facilitate better local detail capture in object detection, benefiting categories with intricate details such as "cats" and "dogs".

The introduction of SimPPFCSPC, with its enhanced feature pyramid and pooling strategy, supplied the model with richer scale information.

Figure 5 illustrates that the numerals along the main diagonal (extending from the upper left to the lower right corner) represent accurate predictions, signifying instances where the model has correctly identified each category as such. Conversely, figures situated off the main diagonal denote incorrect predictions where the model has erroneously assigned a category to another.

The diagonal values within the confusion matrices of each model, as depicted in Figure 5, correspond to the count of instances accurately classified. Higher values on this diagonal indicate superior performance. By comparing these figures, one may determine which model performs optimally for specific categories. Values off the diagonal reflect the number of predictive errors a model makes within a particular category. A model exhibiting lower numbers on the diagonal is deemed to yield more precise predictions for that specific category. Within these confusion matrices, several observations can be made as follows: Firstly, the yolo-tiny has an overall lower prediction count. Secondly, in distinguishing between 'chair' and 'sofa', the yolov8n-Simppfcspc-Res2 emerges as the most accurate. Thirdly, YOLOv8-CGRNet demonstrates a more balanced performance distribution across various categories, rendering it robust for overall effectiveness. Furthermore, YOLOv8-CGRNet registers minimal misclassifications in the 'dog' and 'cat' categories and exhibits no misclassifications for 'train' and 'aeroplane'.

As depicted in Figure 6, the model's detection accuracy is evident through its overall and category-specific F1 score performance. Within each subplot, lines of different colors represent various categories, such as "person", "car", "dog", and others. These curves illustrate the balance between the probability of correct predictions and the probability of false predictions at various confidence thresholds, which serves to assess model performance.

Upon analysis of the F1 confidence curves, it is observed that the majority of the models reach their peak between the confidence thresholds of 0.3 and 0.6. This peak may represent the optimal balance point between accuracy in detection and the reduction of false positives. Each model demonstrates significant variation in detection accuracy across different categories. Lightweight models, specifically YOLOv3-tiny and YOLOv5n, score relatively lower on the overall F1 score, highlighting the trade-off between speed and accuracy inherent in lightweight models. YOLOv8-CGRNet, in particular, showcases that each ablation study exhibits higher F1 scores, particularly in the medium confidence threshold range. It suggests that structural improvements can significantly enhance model detection performance.

Table 3 illustrates the progression of mean Average Precision (mAP50) for different models with the increase in training epochs. mAP50, which employs an Intersection over Union (IoU) threshold of 0.5 to calculate precision, shows a rapid ascent in the initial epochs for all model variants, indicating an expedited learning process and improvement in target detection accuracy. Variance is observed in the rate at which different model variants converge. During the early training stages, the performance enhancements for these variants are closely matched. Over time, however, models with more complex structures may exhibit superior performance. As the number of epochs rises, the rate of performance improvement for all models begins to plateau, suggesting a stabilization in learning efficacy.

The YOLOv8-CGRNet model's performance in later epochs exceeds that of other models, which substantiates the assertion that YOLOv8-CGRNet's strategic enhancements indeed bolster the model's generalization abilities in complex scenarios.

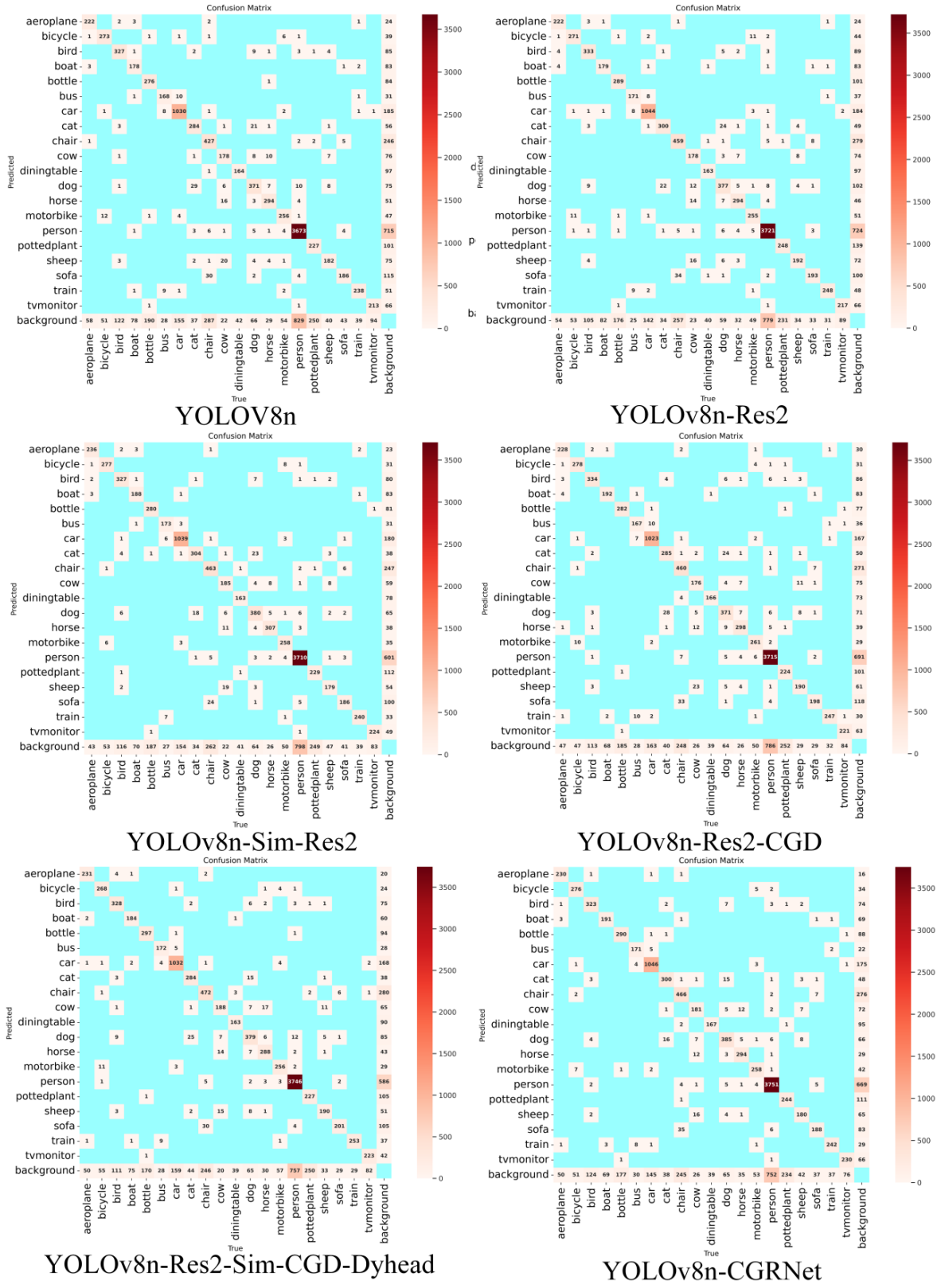

Figure 5. Confusion matrix.

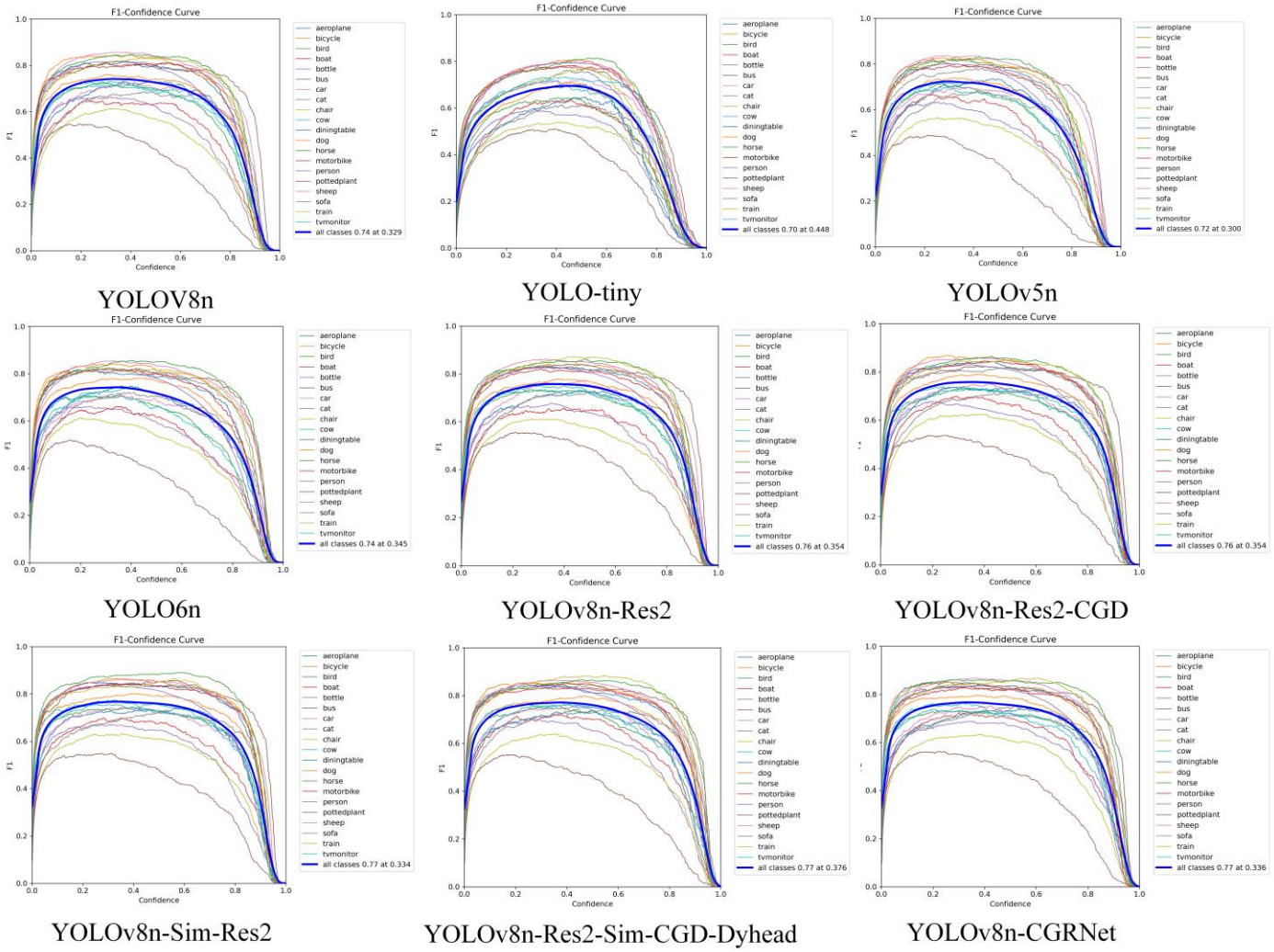

Figure 6. The F1 score performance of the model across multiple categories at varying confidence thresholds.

Table 3. The progress of mAP50 per 10 epochs.

Epochs	Yolov5n	Yolov6n	Yolov8n	Yolov8n-W	yolov8n-C	Yolov8n-S	Yolov8n-R	YOLOv8n-R-C-W	YOLOv8n-R-S-W	YOLOv8n-R-S-C-D	Yolov8n-CGRNet
1	0.258	0.099	0.592	0.592	0.127	0.338	0.385	0.116	0.089	0.381	0.333
11	39.104	34.854	47.921	47.921	44.188	47.449	46.333	45.993	45.215	41.167	41.449
21	54.421	55.094	61.498	61.498	59.795	62.95	62.202	61.874	62.111	60.095	62.101
31	62.427	63.847	67.648	67.648	67.672	69.514	69.123	69.131	70.09	68.609	70.142
41	66.522	68.487	71.573	71.573	72.259	73.328	73.171	73.351	74.205	73.211	74.006
51	68.908	71.127	73.732	73.732	74.34	75.611	75.036	75.859	76.012	75.576	76.195
61	70.365	72.662	74.848	74.848	75.645	77.034	76.173	77.001	77.37	76.968	77.253
71	71.33	73.599	75.57	75.57	76.363	77.675	76.974	77.588	78.125	77.948	78.136
81	72.019	74.118	76.073	76.073	76.904	78.027	77.471	78.032	78.645	78.578	78.617
91	72.571	74.7	76.434	76.434	77.314	78.445	77.874	78.412	79.108	79.009	79.044
101	73.026	75.251	76.831	76.831	77.711	78.829	78.218	78.743	79.531	79.453	79.484
111	73.483	75.844	77.155	77.155	78.022	79.144	78.507	79.144	79.834	79.739	79.875
121	73.905	76.355	77.498	77.498	78.256	79.418	78.851	79.466	80.16	80.101	80.193
131	74.28	76.745	77.775	77.775	78.531	79.597	79.096	79.657	80.364	80.389	80.553
141	74.704	77.215	78.048	78.048	78.797	79.722	79.362	79.893	80.59	80.688	80.865
151	75.044	77.501	78.29	78.29	78.944	79.812	79.596	80.113	80.804	80.983	81.139
161	75.391	77.768	78.469	78.469	79.154	79.966	79.796	80.206	81.067	81.214	81.308
171	75.718	78.05	78.686	78.686	79.319	80.019	79.962	80.292	81.14	81.344	81.527
181	76.002	78.208	78.893	78.893	79.411	80.016	80.119	80.383	81.186	81.492	81.68
191	76.291	78.478	78.952	78.952	79.63	79.979	80.125	80.482	81.213	81.637	81.824
200	76.46	78.645	78.986	78.986	79.81	80.001	80.139	80.476	81.258	81.701	81.91

Table 4 showcases the performance of various YOLOv6n and YOLOv8n model versions and variants in object detection on the VisDrone2019-DET dataset across different

categories. These models incorporate several technological enhancements like Res2Net, CGNet, SimPPFCSPC, DyHead, and WIoU, contributing to their performance. A general trend of improvement is observed from YOLOv6n to YOLOv8-CGRNet in overall Average Precision across all categories, with YOLOv8-CGRNet performing the best at 29.9%. In specific categories, all models tend to perform relatively well in detecting cars and trucks while showing lower effectiveness in categories like bicycles and people. The introduction of technologies like Res2Net and CGNet generally correlates with performance enhancement, especially in complex or dynamic scenes. The combination of multiple technologies, as seen in models like YOLOv8n-R-S-C-D, often leads to further improvements. Compared to YOLOv6n, the YOLOv8n series demonstrates superior performance in most categories, likely due to differences in architecture and training methodologies. This data illustrates that by employing various architectural enhancements and combined techniques, the YOLO models significantly enhance their ability to detect objects in complex drone imagery, particularly in common categories such as vehicles and pedestrians.

Table 4. The mAP50 for the VisDrone2019-DET. Herein, "R" stands for Res2Net, "C" for CGNet, "S" for SimPPFCSPC, "D" for DyHead, and "W" for WIoU. In the table, the bolded values represent the highest mAP among these methods.

Method	All	Pedestrian	People	Bicycle	Car	Van	Truck	Tricycle	Awning Tricycle	Bus	Motor
yolov6n	25.3	20	10.3	4.12	66	31	29.1	9.78	12.4	50.5	20.2
yolov5n	26.5	21.3	11.3	6.04	66.5	30.8	31.3	11.5	13.1	50.2	23.2
yolov8n	26.9	22.1	11.6	5.89	66.7	30.5	30.9	12.7	14.7	50.8	23.1
Yolov8n-S	27.1	22.9	11.4	7.36	66.8	30.8	29.9	12.4	15.1	50.9	24
yolov8n-W	27.5	23	12.7	6.49	67.2	31.3	31.6	11.6	14.9	52.6	23.8
yolov8n-C	27.8	22.7	12.2	5.56	67.5	32.5	32.2	13.3	16.6	52	23
yolov8n-R-S	28.2	23.2	12.8	7.27	68	32.1	33.2	13.1	15.7	52.5	24.3
yolov8n-R	28.2	23.4	12.4	6.31	68.1	32.5	32.9	13	16.3	52.7	24.2
yolov8n-R-C	28.5	23.8	13.3	7.5	68.1	32.3	31.9	13	16.6	53.3	25
yolov8n-D	28.7	23.5	12.5	7.44	68.3	32.9	33.8	14.4	15.7	53.2	24.7
YOLOv8n-R-S-C-D	28.8	22.7	12.4	6.98	68.6	**33.9**	35.6	12.7	17.6	53.1	24.4
yolov8- CGRNet	29.9	**24.2**	**13.6**	**7.72**	**69.3**	33.3	36.8	**14.9**	**18**	**55.2**	**26.1**

The combined preprocess and postprocess time on the RTX 3080 is only 2 ms. In contrast, the combined preprocess and postprocess time on the ARM Cortex-A57 is significantly higher at 24 ms.

The data in Tables 5 and 6 display tests conducted on the NVIDIA GeForce RTX 3080, showing that CGR-Net experiences negligible differences in terms of time, influenced by preprocess and postprocess times.

Table 5. The test results of various object detection models when evaluated on the VisDrone dataset, utilizing a NVIDIA GeForce RTX 3080 graphics card.

	Yolov6n	Yolov5n	Yolov8n	Yolov8n-S	Yolov8n-W	Yolov8n-C	Yolov8n-R-C	Yolov8n-D	YOLOv8n-R-S	Yolov8n-R	YOLOv8n-R-S-C-D	Yolov8-CGRNet
FPS	175.44	153.85	161.29	151.52	153.85	119.05	112.36	99.01	109.89	131.57	85.47	84.03
Inference (ms)	3.7	4.5	4.2	4.6	4.5	6.4	6.9	8.1	7.1	5.6	9.7	9.9
GFLOPs	11.8	7.1	8.1	9.4	8.1	9.1	9.9	9.7	11.2	8.8	12.8	12.8

Table 6. The VOC2007 test results of various object detection models when evaluated on the VisDrone dataset, utilizing a NVIDIA GeForce RTX 3080 graphics card.

	Yolov6n	Yolov5n	Yolov8n	Yolov8n-S	Yolov8n-W	Yolov8n-C	Yolov8n-R-C	Yolov8n-D	YOLOv8n-R-S	Yolov8n-R	YOLOv8n-R-S-C-D	Yolov8-CGRNet
FPS	72.46	109.89	99.01	87.72	99.01	90.09	84.03	85.47	75.76	92.59	67.57	67.57
Inference (ms)	1.8	2.1	2.3	2.3	1.9	3.0	3.5	4.6	3.5	2.6	6.4	6.5
GFLOPs	11.8	7.1	8.1	9.4	8.1	9.1	9.9	9.7	11.2	8.8	12.8	12.8

The data from Tables 7 and 8, showcasing tests on the ARM Cortex-A57 CPU, a proxy for mobile device performance, illustrate CGR-Net's operational feasibility in a mobile

environment. Although there is a noticeable drop in FPS and an increase in inference time compared to high-end GPUs, CGR-Net still manages to function, suggesting its adaptability to less powerful hardware.

Table 7. The test results of various object detection models when evaluated on the VisDrone dataset, utilizing an ARM Cortex-A57cpu.

	Yolov6n	Yolov5n	Yolov8n	Yolov8n-S	Yolov8n-W	Yolov8n-C	Yolov8n-R-C	Yolov8n-D	YOLOv8n-R-S	Yolov8n-R	YOLOv8n-R-S-C-D	Yolov8-CGRNet
FPS	1.07	1.02	1.10	1.00	1.01	0.76	0.62	0.52	0.61	0.99	0.42	0.41
Inference (ms)	929	974	907	999	984	1310	1624	1931	1631	1011	2404	2458
GFLOPs	11.8	7.1	8.1	9.4	8.1	9.1	9.9	9.7	11.2	8.8	12.8	12.8

Table 8. The VOC2007 test results of various object detection models when evaluated on the VisDrone dataset, utilizing an ARM Cortex-A57 graphics card.

	Yolov6n	Yolov5n	Yolov8n	Yolov8n-S	Yolov8n-W	Yolov8n-C	Yolov8n-R-C	Yolov8n-D	YOLOv8n-R-S	Yolov8n-R	YOLOv8n-R-S-C-D	Yolov8-CGRNet
FPS	0.44	0.40	0.47	0.43	0.45	0.32	0.29	0.22	0.27	0.39	0.19	0.19
Inference (ms)	2225	2440	2086	2254	2163	3083	3457	4506	3705	2510	5314	5321
GFLOPs	11.8	7.1	8.1	9.4	8.1	9.1	9.9	9.7	9.4	8.8	12.8	12.8

4. Discussion

According to the data in Table 2, the YOLOv8 series of models surpasses other models in overall mAP50 performance on the PASCAL VOC2007 test set, particularly after the integration of Res2Net (R), Context GuidedNet (C), SimPPFCSPC (S), DyHead (D), and WIoU (W). YOLOv8-CGRNet, with its ensemble of sophisticated network modules and mechanisms, exhibits increased precision and robustness.

The framework, however, is not without drawbacks. First, higher accuracy often comes with an increase in computational costs, as indicated in Tables 5–8. Balancing accuracy with computational expense remains a challenge in resource-constrained application scenarios. Second, certain categories, like 'bottle' or 'plant', show lower detection accuracy, which suggests that deficiencies remain in detecting small or irregularly shaped objects. Third, the incorporation of new technologies significantly heightens model complexity, which leads to prolonged training durations and heightened computational resource demands.

Based on the experimental comparisons shown in Tables 1–4, this study explores various application scenarios, demonstrating how the adapted models perform across different datasets. It is observed that while some models may excel in one dataset, their effectiveness diminishes when applied to another. Our integrated model, although increasing computational demands, enhances adaptability and accuracy across diverse datasets. This feature is particularly advantageous for mobile deployment, where processing capabilities are limited.

The augmented model maintains efficacy on mobile platforms, as the camera capture rate limits render the performance of the improved model nearly equivalent to the original on standard computers or compute-capable boards. However, on lower-powered ARM chips, while the speed may decrease, the accuracy significantly improves, making it suitable for scenarios where precision is paramount.

This flexibility is crucial in a range of applications. For instance, in autonomous vehicle systems where hardware limitations constrain processing power and speed, the model's high-precision object detection capabilities are vital for safety and reliability. It accelerates the response to road conditions, obstacles, and traffic signs, promoting safer navigation and decision making. Additionally, the model's efficiency makes it ideal for real-time detection applications such as Augmented Reality (AR), autonomous drone navigation, surveillance, access control, and interactive marketing.

In summary, our fusion model balances increased computational demands with improved adaptability and accuracy, making it a versatile solution for both power-constrained

mobile devices and higher-capability computing platforms, catering to a wide range of applications that require varying levels of accuracy and processing speed.

5. Conclusions

The advent of deep learning and computer vision has ushered in a new era of analytical capabilities, with model optimization and improvement taking center stage. This progression necessitates a deeper exploration of algorithmic architectures and parameter optimization, particularly in the context of object detection models on mobile devices. The demand for models that balance a small memory footprint with high accuracy is rapidly increasing, presenting a unique set of challenges given the parameter-heavy nature of current leading-edge networks, which are ill-suited for mobile environments.

In this paper, we have proposed the YOLOv8-CGRNet method, which represents a significant step forward in this domain. Our approach synergizes YOLOv8 with the CGNet and the Res2Net structure, enhancing the model's ability to learn deep features from Res2Net. This integration provides a multi-scale representation at a more granular level without adding to the model's complexity or computational demands. The CGNet is particularly adept at capturing local features and contextual information, leveraging spatial dependencies to bolster accuracy.

Furthermore, we have delved into an improved pyramid network combination utilizing the SimPPFCSPC structure, which augments the network's proficiency in managing FPN. The innovative application of a dynamic, non-monotonic FM gradient gain distribution strategy, which operates on an anchor-free basis, effectively addresses the challenge of low-quality samples, thereby enhancing the detector's overall efficacy.

This head network amalgamates the strengths of various modules, with a particular emphasis on our Unifying Object Detection Heads with Attentions. This module is designed to be versatile across a range of input scenarios, improving the model's adaptability. By integrating multiple self-attention mechanisms in a coordinated fashion across feature levels, spatial positions, and output channels, we have succeeded in elevating the representational power of the object detection heads without incurring a substantial computational burden.

Our experimental evaluation of the VOC2007 and VOC2012 datasets has provided a robust model for demonstrating the efficacy of YOLOv8-CGRNet. The results have been promising, showcasing the model's capability to achieve top-tier performance on these well-established benchmarks. However, it is important to acknowledge the potential limitations and avenues for future research. While our model excels in memory efficiency and accuracy, the quest for an even smaller model footprint without compromising performance continues. Additionally, the real-time application of YOLOv8-CGRNet in diverse environments remains an area ripe for exploration.

In conclusion, YOLOv8-CGRNet stands as a testament to the potential of innovative model design in the realm of mobile object detection. It paves the way for future research aimed at refining and deploying lightweight, high-performance models across a spectrum of real-world applications.

Future optimizations will focus on the following aspects: First, the exploration of new lightweight model designs that do not compromise applicability in edge computing and mobile devices. Second, the development of models capable of operating across multiple domains and the enhancement of their adaptability to different data distributions. Third, the employment of integrated learning approaches to enhance model robustness and more research on dynamic adaptive networks that allow the model to adjust its structure dynamically based on different inputs.

Performance can also be improved by expanding the training dataset. Recognizing that a more diverse dataset, such as Pascal 2011, which includes the "Pascal Visual Object Classes (VOC) Challenge", can significantly improve the model's robustness. Moreover, integrating various databases, such as ImageNet and COCO (Common Objects in Context), will further refine our model's detection capabilities. This expansion not only provides

young researchers with the opportunity to delve deeper into a broader range of data processing and analysis techniques but also ensures the adaptability of the model in different real-world scenarios. Simultaneously, we can explore integration with assessment-type algorithms like DeepRPN-BIQA [13] or apply the structure to segmentation algorithms similar to SegR-Net [31]. On the other hand, future iterations of YOLOv8-CGRNet can benefit from exploring advanced machine learning techniques and cross-disciplinary applications, particularly in medical imaging, where increased accuracy and precision are crucial. Therefore, future work on our model is not only aimed at enhancing its performance but also at expanding its application spectrum in the fields of computer vision and medical image analysis.

Author Contributions: Methodology, Y.N. and W.C.; Software, Y.N.; Validation, C.S.; Writing—original draft, Y.N.; Writing—review & editing, S.F.; Supervision, S.F. All authors have read and agreed to the published version of the manuscript.

Funding: This project is funded by Liaoning University of Science and Technology, under the 2023 Graduate Education Reform, Technological Innovation, and Entrepreneurship Project of Liaoning University of Science and Technology, with the funding number LKDYC202208.

Data Availability Statement: The data presented in this study are available in the article.

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

References

1. Cheng, G.; Yuan, X.; Yao, X.; Yan, K.; Zeng, Q.; Xie, X.; Han, J. Towards large-scale small object detection: Survey and benchmarks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 23821153. [CrossRef] [PubMed]
2. Wang, H.; Xu, Y.; Wang, Z.; Cai, Y.; Chen, L.; Li, Y. Centernet-auto: A multi-object visual detection algorithm for autonomous driving scenes based on improved centernet. *IEEE Trans. Emerg. Top. Comput. Intell.* **2023**, *7*, 742–752. [CrossRef]
3. Jahangir, H.; Lakshminarayana, S.; Maple, C.; Epiphaniou, G. A Deep Learning-Based Solution for Securing the Power Grid against Load Altering Threats by IoT-Enabled Devices. *IEEE Internet Things J.* **2023**, *10*, 23205575. [CrossRef]
4. Shamshad, F.; Khan, S.; Zamir, S.W.; Khan, M.H.; Hayat, M.; Khan, F.S.; Fu, H. Transformers in medical imaging: A survey. *Med. Image Anal.* **2023**, *88*, 102802. [CrossRef] [PubMed]
5. You, K.; Zhou, C.; Ding, L. Deep learning technology for construction machinery and robotics. *Autom. Constr.* **2023**, *150*, 104852. [CrossRef]
6. Ragu, N.; Teo, J. Object detection and classification using few-shot learning in smart agriculture: A scoping mini review. *Front. Sustain. Food Syst.* **2023**, *6*, 1039299. [CrossRef]
7. Zhang, S.; Zhang, C. Modified U-Net for plant diseased leaf image segmentation. *Comput. Electron. Agric.* **2023**, *204*, 107511. [CrossRef]
8. Fu, C.; Lu, K.; Zheng, G.; Ye, J.; Cao, Z.; Li, B.; Lu, G. Siamese object tracking for unmanned aerial vehicle: A review and comprehensive analysis. *Artif. Intell. Rev.* **2023**, *56*, 1417–1477. [CrossRef]
9. Zeng, Z.; Li, Z.; Cheng, D.; Zhang, H.; Zhan, K.; Yang, Y. Two-stream multirate recurrent neural network for video-based pedestrian reidentification. *IEEE Trans. Ind. Inform.* **2017**, *14*, 3179–3186. [CrossRef]
10. 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.
11. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
12. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
13. ur Rehman, M.; Nizami, I.F.; Majid, M. DeepRPN-BIQA: Deep architectures with region proposal network for natural-scene and screen-content blind image quality assessment. *Displays* **2022**, *71*, 102101. [CrossRef]
14. Liu, H.; Zheng, T.; Sun, F.; Wang, C.; Deng, L. ER-DeepSORT: Pedestrian Multiobject Tracking with Enhanced Reidentification. *IEEJ Trans. Electr. Electron. Eng.* **2023**, *18*, 427–435. [CrossRef]
15. Li, X.; Jiang, Y.; Liu, Y.; Zhang, J.; Yin, S.; Luo, H. RAGCN: Region aggregation graph convolutional network for bone age assessment from X-ray images. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–12. [CrossRef]
16. Ji, Y.; Ni, L.; Zhao, C.; Lei, C.; Du, Y.; Wang, W. TriPField: A 3D potential field model and its applications to local path planning of autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 3541–3554. [CrossRef]
17. Müller, H.; Niculescu, V.; Polonelli, T.; Magno, M.; Benini, L. Robust and efficient depth-based obstacle avoidance for autonomous miniaturized uavs. *IEEE Trans. Robot.* **2023**, *39*, 4935–4951. [CrossRef]

18. Bayer, M.; Kaufhold, M.-A.; Buchhold, B.; Keller, M.; Dallmeyer, J.; Reuter, C. Data augmentation in natural language processing: A novel text generation approach for long and short text classifiers. *Int. J. Mach. Learn. Cybern.* **2023**, *14*, 135–150. [CrossRef]
19. Gao, S.H.; Cheng, M.M.; Zhao, K.; Zhang, X.Y.; Yang, M.H.; Torr, P. Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 652–662. [CrossRef]
20. Wu, T.; Tang, S.; Zhang, R.; Cao, J.; Zhang, Y. Cgnet: A light-weight context guided network for semantic segmentation. *IEEE Trans. Image Process.* **2020**, *30*, 1169–1179. [CrossRef]
21. Li, C.; Li, L.; Geng, Y.; Jiang, H.; Cheng, M.; Zhang, B.; Ke, Z.; Xu, X.; Chu, X. Yolov6 v3. 0: A full-scale reloading. *arXiv* **2023**, arXiv:2301.05586.
22. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. *arXiv* **2023**, arXiv:2301.10051.
23. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
24. Everingham, M.; Eslami, S.A.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
25. 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]
26. Martin, D.R.; Fowlkes, C.C.; Malik, J. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 530–549. [CrossRef] [PubMed]
27. Lin, T.-Y.; Dollar, 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.
28. Dai, X.; Chen, Y.; Xiao, B.; Chen, D.; Liu, M.; Yuan, L.; Zhang, L. Dynamic head: Unifying object detection heads with attentions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7373–7382.
29. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 28.
30. 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; Proceedings, Part I 14. Springer International Publishing: Berlin/Heidelberg, Germany, 2016.
31. Ryu, J.; Rehman, M.U.; Nizami, I.F.; Chong, K.T. SegR-Net: A deep learning framework with multi-scale feature fusion for robust retinal vessel segmentation. *Comput. Biol. Med.* **2023**, *163*, 107132. [CrossRef]