International Core Journal of Engineering Volume 11 Issue 2, 2025
ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0004

ResEff-YOLO: Accuracy Enhancement of YOLOv8 through Integration of ResNet, SPPF, and EfficientHead Modules

Yihe Jin[1,2], Haiyun Gan[1,2,*], Jun Li[1,2], Tao Zhang[1,2]

[1] School of Automobile and Transportation, Tianjin University of Technology and Education, Tianjin 300222, China

[2] National and Local Joint Engineering Research Center for Intelligent Vehicle Road Collaboration and Safety Technology, Tianjin 300222, China

[*]Corresponding Author

Abstract

Since computer vision has been widely applied in daily life, such as in autonomous vehicles, portable applications, augmented reality systems, and medical image analysis, the demand for architectures with lower complexity and higher accuracy has become a priority. To improve the accuracy and complexity of object detection, various methods have been developed, such as R-CNN and YOLO models. In this study, we propose an enhanced version of the YOLOv8 model to further improve detection accuracy and efficiency. Specifically, we adopted EfficientHead as the detection head, which optimizes computational resource utilization and improves inference speed while maintaining detection accuracy. For the backbone network, we incorporated the ResNet18d module along with the SPPF_LSKA module, which enhances the network's ability to learn multi-scale features, surpassing traditional convolutional layers. The deep stem structure of ResNet18d helps retain more spatial information, while SPPF_LSKA introduces Large Separable Kernel Attention (LSKA) to enhance the SPPF feature extractor, improving multi-scale feature extraction and handling of complex scenes. Experiments on the VOC dataset demonstrate that the ResEff-YOLO model outperforms the YOLOv8 series, with a mean average precision (mAP) improvement of approximately 4% and an mAP50-95 improvement of 4.2%.

Keywords

Object Detection; YOLOv8; Multi-Scale Feature Extraction.

1. Introduction

In recent years, deep learning technologies, particularly Convolutional Neural Networks (CNNs), have achieved significant advancements in the field of computer vision. Object detection, as one of the core tasks in computer vision, finds extensive applications in areas such as autonomous driving, security monitoring, and drone navigation. The YOLO (You Only Look Once) series of models has become a prominent method in object detection due to its end-to-end detection approach, efficient real-time performance, and good accuracy. YOLOv8, compared to earlier versions in this series, has demonstrated significant improvements in both accuracy and speed. However, as task complexity continues to increase, further enhancing detection precision remains a major challenge for researchers.

Existing YOLO series models have shown exceptional performance on several public datasets, particularly the COCO and PASCAL VOC datasets. YOLOv4 introduced CSPDarknet53 as its backbone and utilized techniques such as Spatial Pyramid Pooling (SPP) to improve feature extraction capabilities and inference speed. YOLOv5 made further advancements in lightweight design and

ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0004

engineering optimization, improving training processes and model deployment convenience. YOLOv8 combines the advantages of previous models, achieving higher accuracy and speed through technologies like dynamic label assignment and anchor box optimization. However, YOLOv8 still faces certain limitations in handling multi-scale targets, complex backgrounds, and objects of various shapes and poses as application scenarios become more complex.

To address this issue, this paper proposes an improved YOLOv8 model architecture. Specifically, we use ResNet18d combined with the SPPF_LSKA module as the backbone for YOLOv8 to enhance feature extraction capabilities. The deep structure of ResNet18d is better suited for extracting information in complex scenes, while SPPF_LSKA improves the model's detection performance for targets of different scales through long and short-term feature aggregation. Additionally, we adopt EfficientHead as the detection head for YOLOv8, which maintains excellent detection accuracy while reducing computational resources. These improvements lead to enhanced performance of the model on several standard datasets.

This study demonstrates that optimizing YOLOv8's backbone and detection head can further enhance object detection accuracy and performance while maintaining efficient detection speed. This provides new insights and directions for future object detection model design.

2. Related Work

In recent years, significant progress has been made in the field of object detection, especially with the advent of deep learning-based models becoming mainstream. These models have achieved impressive performance in complex scenarios by building powerful feature extraction networks and efficient detection heads. Among them, the YOLO (You Only Look Once) series models are one of the most influential representatives, with each version achieving substantial improvements in accuracy and speed. This chapter reviews the foundational work in this field, focusing on YOLO series models, the application of ResNet18d as a backbone, and other related improvements.

2.1 YOLO Series Models

Since its inception, the YOLO series models have garnered widespread attention due to their single-stage detection architecture, real-time detection capability, and end-to-end design philosophy. YOLOv1 initially approached object detection as a regression problem, directly predicting bounding boxes and class probabilities through a unified network structure [1]. Subsequently, YOLOv2 and YOLOv3 significantly improved detection accuracy by introducing multi-scale detection and enhanced anchor box mechanisms [2][3]. YOLOv4 further advanced this by incorporating CSPDarknet53 as the backbone and integrating various training techniques such as CIOU loss and Mosaic data augmentation, thereby enhancing performance [4].

YOLOv5 optimized the model for lightweight design, achieving better inference speed and adaptability while maintaining detection accuracy [5]. YOLOv8 further introduced dynamic label assignment, anchor box optimization, and adaptive feature fusion modules, enhancing the model’s detection capability in complex scenarios[19]. However, despite the performance improvements in YOLOv8, there is still room for enhancement in its backbone network and detection head, especially in handling multi-scale targets, complex backgrounds, and resource-constrained environments, where further optimization of the model’s accuracy and efficiency is needed

2.2 Application of ResNet18d as a Backbone

ResNet (Residual Network) [6] has become a mainstream backbone network in various computer vision tasks due to its deep network structure and residual connections (Skip Connections) that effectively address the vanishing gradient problem. The core advantage of ResNet lies in its ability to build deeper network layers, thus extracting more rich feature representations. In object detection tasks, ResNet has been widely applied as the backbone in various detection frameworks, such as Faster R-CNN [7], Mask R-CNN [8], and RetinaNet [9].

ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0004

Researchers have also explored using ResNet as the backbone for YOLO models to enhance feature extraction capabilities. For instance, some variants of YOLOv4 have adopted ResNet as the backbone, showcasing improved detection capabilities in complex scenarios [4]. However, due to the high inference speed requirements of YOLO models, using ResNet as a backbone often incurs computational overhead. Therefore, researchers have begun exploring how to reduce computational costs through lightweight network design while maintaining detection accuracy.

In this study, we selected ResNet18d as the backbone for YOLOv8. ResNet18 is a shallower version of the ResNet series, consisting of 18 layers (convolutional layers and fully connected layers). Compared to ResNet18, ResNet18d employs a different input layer structure, where the input layer's convolutional kernels are replaced by three 3x3 convolutional layers with 2x2 strides. This design retains more spatial information and is beneficial for handling more complex or higher-resolution images.

In summary, the input layer design of ResNet18d is more sophisticated and suitable for tasks requiring finer feature extraction, while the standard ResNet18 is more appropriate for ordinary scenarios. ResNet18d is a variant of ResNet18 that retains deep feature extraction capabilities while reducing computational resource consumption. By introducing the SPPF_LSKA (Spatial Pyramid Pooling with Long Short-term Key Aggregation) module, it further enhances the model's detection capability for different scale targets, offering the potential for deploying efficient detection models on resource-constrained devices.

2.3 SPPF_LSKA Module and Multi-Scale Feature Fusion

Multi-scale feature fusion is a critical technology in object detection, especially when dealing with targets of varying sizes. Effective fusion of multi-scale information is crucial for improving detection accuracy. The Spatial Pyramid Pooling (SPP) module, which uses pooling operations at different scales to merge global and local features, has been widely used in various detection networks. However, traditional SPP modules suffer from insufficient information aggregation when handling features at different scales [10].

To address this, the SPPF_LSKA module builds on SPP by incorporating the LSKA (Large Separable Kernel Attention) mechanism. This mechanism utilizes large separable convolutional kernels to enhance the model's feature extraction capabilities and improve sensitivity to key features. In this study, the SPPF_LSKA module is integrated into the ResNet18d backbone, further enhancing the model's feature extraction and fusion capabilities.

2.4 Improvements in Detection Heads and the Application of EfficientHead

In object detection tasks, the design of the detection head directly impacts the model's detection accuracy and inference efficiency. The detection head in YOLO series models is primarily responsible for object localization and classification based on the feature maps extracted by the backbone network. As object detection tasks have become more complex, researchers have made various optimization attempts to improve the structure of detection heads. For example, YOLOv3 introduced multi-scale detection mechanisms [3], while YOLOv4 adopted PANet (Path Aggregation Network) to enhance feature fusion [4], improving detection performance.

Recently, with the rise of lightweight neural network design, researchers have focused on maintaining the efficiency of detection heads while reducing parameters and computational load. EfficientHead is a lightweight-designed detection head that optimizes network structure to reduce redundant computations while maintaining effective feature fusion capabilities [11]. This detection head has demonstrated excellent performance in several lightweight object detection models.

In this study, we use EfficientHead as the detection head for YOLOv8, thereby maintaining detection accuracy while reducing computational load. By combining EfficientHead with the improved backbone network, the proposed model demonstrates enhanced performance on multiple public datasets, particularly achieving high accuracy and inference efficiency on the PASCAL VOC dataset.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0004

3. Method

3.1 YOLOv8 Algorithm Overview

YOLOv8 is an efficient real-time object detection model composed of an input layer, backbone network, neck architecture, and detection head. It features a deep convolutional neural network with 53 convolutional layers[12]. During the detection process, the input image passes through convolutional layers (Conv), shortcut connections (C2f), and an SPPF structure. The resulting feature maps then enter the neck architecture, where a series of convolutional operations and upsampling processes are performed. This stage achieves a fusion of shallow and deep features, which helps in more effectively detecting objects. The fused multi-scale feature maps are then fed into three different detection heads, each designed to handle objects of various sizes. For instance, the detection head responsible for large objects focuses more on deep feature information. Finally, each detection head outputs a series of prediction boxes containing the detected objects. In this paper, we employ an improved version of the YOLOv8 model to recognize and detect target objects. The specific model training and detection process is the same, as shown in Figure 1.

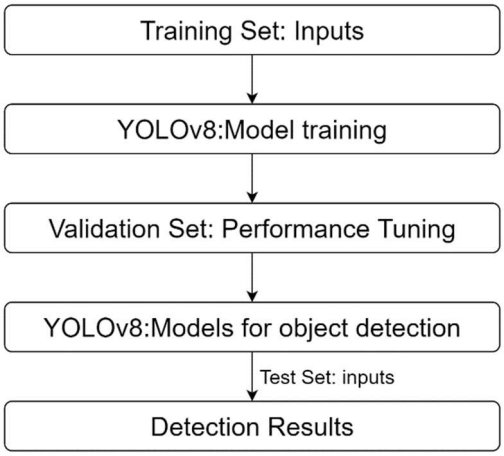

Figure 1. The flowchart of YOLOv8.

3.2 Improved YOLOv8 Network Structure

This paper proposes the ResEff-YOLOv8 framework, based on the YOLOv8 model, which optimizes the original YOLOv8 backbone and detection head to enhance performance in detecting multi-scale objects, complex backgrounds, and objects with varying shapes and poses. The improvements involve integrating three specialized mechanisms into the YOLOv8 framework:

ResNet18d: ResNet18d is an improved version of ResNet18, incorporating residual blocks to address gradient vanishing and explosion issues. It includes 18 convolutional layers, organized into multiple residual blocks, each comprising two convolutional layers and a shortcut connection. The shortcut connection enables input to bypass layers and directly connect to the output, allowing the network to learn residual information. Unlike ResNet18, ResNet18d optimizes the receptive field and feature representation through adjustments to the convolutional kernel size, stride, and downsampling strategy. Utilizing ResNet18d as the backbone for YOLOv8 leverages its enhanced deep convolutional structure and residual learning mechanism to extract richer and more diverse features from the input image[13]. The design of ResNet18d not only improves the network's training stability but also demonstrates better performance in handling multi-scale objects, which is critical for enhancing YOLOv8's accuracy and robustness in object detection.

SPPF-LSKA: The SPPF-LSKA module, combining Spatial Pyramid Pooling (SPPF) capabilities with the Large Separable Kernel Attention (LSKA) mechanism, plays a crucial role in extracting and enhancing features as part of the backbone. The SPPF component maps input feature maps across multiple spatial scales by constructing different scale pooling layers and then aggregates this multi-

ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0004

scale feature information to capture rich contextual details. Meanwhile, the LSKA component uses large separable convolutional kernels to build an attention mechanism, dynamically adjusting the importance of different positions and channels in the feature map, thus enhancing the model's sensitivity to key features. The SPPF-LSKA module organically integrates these two components to create an efficient module capable of both extracting multi-scale features and strengthening critical feature representations[14]. By introducing SPPF-LSKA, YOLOv8 can more effectively capture multi-scale information within images, while the LSKA attention mechanism highlights features essential for object detection. This enhanced feature representation not only improves the model's robustness to variations in object shape, size, and pose but also boosts detection accuracy in complex backgrounds.

EfficientHead: The original detection head used a decoupled head structure[19] with a parallel branch approach, where features first pass through two 3x3 convolutional layers and then a standard convolution to separately calculate bounding box loss and classification loss. To improve the performance of the enhanced model in high-precision object detection tasks, it was necessary to redesign the detection head to obtain better feature detail extraction. Therefore, this paper proposes an efficient module to replace the 3x3 convolutional layers in the original YOLOv8s model detection head.

Our research provides a highly accurate YOLO model that demonstrates superior performance across various datasets, excelling in handling multi-scale objects, complex backgrounds, and objects with different shapes and poses. This model offers a viable solution for tasks requiring high-precision object detection, significantly contributing to the overall effectiveness of object recognition and detection technologies.

3.2.1 ResNet18d Module

ResNet, introduced in 2015, is a specialized neural network designed to address the limitations of traditional deep neural networks, where stacking additional layers is intended to improve accuracy and performance, particularly for complex problems. However, in conventional convolutional neural network models, there is a maximum depth threshold, beyond which adding more layers results in increased errors. ResNet (Residual Network) solves the issue of vanishing gradients in deep neural networks by using skip connections that allow gradients to flow through alternative shortcuts.

Inspired by VGG-19[15], the ResNet architecture uses a standard network structure with added skip connections. The architecture primarily consists of 3×3 convolutional layers with fixed feature map dimensions of 64, 128, 256, and 512, and every two convolutional layers are followed by a shortcut. The differences between various types of ResNet (ResNet-18, ResNet-50, ResNet-152) lie mainly in the convolutional setup of the 2.x layers. At each stage, an additional 1×1 convolutional layer is added, but with different feature map dimensions (256, 512, 1024, 2048). Moreover, in ResNet-152, the number of convolutions in the 3.x and 4.x layers is doubled, resulting in 8 and 36 layers, respectively.

As shown in Table 1, a previous study compared the performance of several ResNet family models[16]. The study used a benchmark with a batch size of 16, an image size of 224×224, and was run on a GTX 1080 with 8GB of memory. The speed displayed in Table II represents the total time required for one forward pass and one backward pass. Due to the fewer layers in ResNet-18 compared to other models, it has a relatively faster speed. Therefore, ResNet-18 is used as the backbone for the solution model.

Building upon ResNet-18, ResNet-18d further enhances feature extraction capabilities and adaptability by optimizing the network structure. ResNet-18d retains the core architecture of ResNet-18, which is based on residual block design, but introduces adjustments at a more detailed level. Through the use of larger convolutional kernels, optimized stride designs, and improved downsampling strategies, ResNet-18d expands the receptive field, improving its performance in handling multi-scale objects. While the overall architectures are similar, these refinements in ResNet-18d enable it to provide better feature representation and detection performance than ResNet-18 in

ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0004

specific tasks. Therefore, using ResNet-18d as the backbone for YOLOv8 allows for improved object detection accuracy and robustness while maintaining the model's efficiency.

Table 1. ResNet performance on mobile devices by previous research.

Network	Layers	Speed(ms)
ResNet18	18	31.54
ResNet34	34	51.59
ResNet50	50	103.58
ResNet101	101	156.44
ResNet152	152	217.91

3.2.2 SPPF-LSKA Module

The multi-scale feature extraction module is a crucial component of the YOLO series algorithms, typically located at the end of the backbone network with a fixed output size. The SPPF (Spatial Pyramid Pooling Fast) module is a redesigned version of the SPP (Spatial Pyramid Pooling) module[10]. While the output remains the same, each time features pass through the pooling layers, the results are preserved. After the features go through three rounds of max-pooling, all the resulting outputs are concatenated. The advantage of this approach is that it significantly reduces the computational load and the number of parameters brought by the multi-scale feature extraction module, greatly enhancing the model's speed and efficiency. However, the downside is that its ability to extract spatial information at different scales is not as effective as the SPP module. Therefore, this paper integrates the Lightweight Spatial Kernel Attention (LSKA) module into the SPPF module to improve its capability of extracting spatial information at different scales, without adding significant computational overhead or increasing the number of parameters.

The LSKA module is based on the Large Kernel Attention (LSA) concept[17] and has demonstrated its performance in visual attention networks, as shown in Figure 2. The innovation of LSKA lies in decomposing the 2D convolutional kernels of depthwise separable convolutions into cascaded horizontal and vertical 1D convolutional kernels (1xk, kx1), allowing it to adaptively capture long-range dependencies. This approach retains comparable performance while consuming less computation and memory.

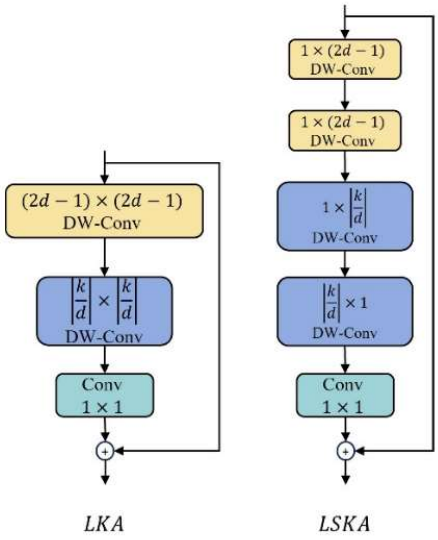

Figure 2. LKA and LSKA structure diagrams.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0004

The improved SPPF module is illustrated in Figure 3, where the LSKA attention module is embedded at the end of the module. After all feature maps are concatenated, they are reintroduced into the LSKA attention module. Based on contextual dependencies, feature weights are adaptively recalibrated, helping the SPPF module enhance its ability to extract multi-scale features.

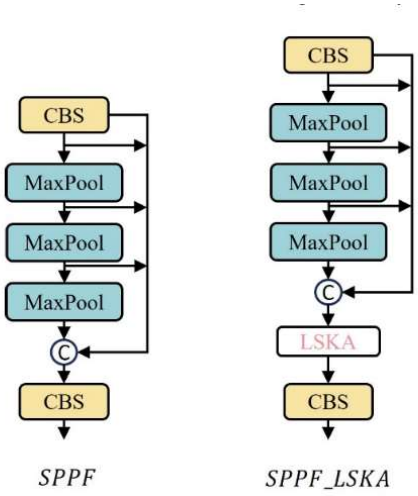

Figure 3. Comparison chart of SPPF before and after improvement.

3.2.3 EfficientHead Module

In object detection, the head structure is typically divided into two types. The first type is the Fully Connected Head (FC head), where features extracted by each node are interconnected in a fully connected layer network. This gives the FC head stronger spatial sensitivity but also results in a higher number of parameters compared to the convolutional head. The second type is the Convolutional Head (Conv head), which has a simpler network structure and requires less computation compared to the FC head. FC heads generally perform better in classification tasks, while Conv heads are more suitable for localization tasks[18]. Although FC heads can offer a slight advantage in detection accuracy, they are not conducive to lightweight models. The detection head in YOLOv8s adopts a convolutional head[19], outputting both classification and regression information. As shown in Figure 4, the original detection head uses a decoupled head structure[20] with a parallel branch approach, where features first pass through two 3x3 convolutional layers followed by a standard convolution to compute the bounding box loss and classification loss separately. To enhance the performance of the improved model in high-precision object detection tasks, it is necessary to redesign the detection head to improve the ability to extract detailed features. Therefore, this paper proposes an efficient module to replace the 3x3 convolutional layers in the original YOLOv8s model's detection head.

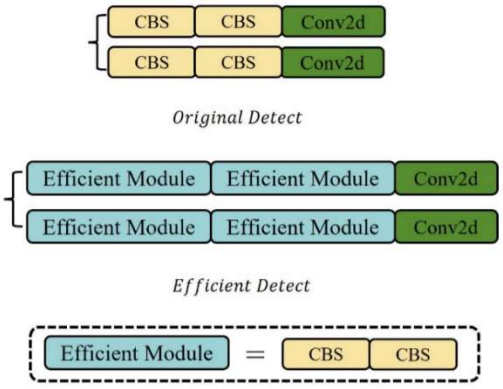

Figure 4. Original and efficient detection heads.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0004

The improved efficient detection head retains the decoupled head structure and parallel branch feature processing approach. However, in each branch, two efficient modules are stacked to replace the two 3x3 convolutional layers in the original detection head, followed by a standard convolutional layer to compute the output. The efficient module consists of two 3x1 convolutional layers. Since 3x1 convolution requires less computation, it enhances non-linearity and improves the model's ability to express complex functions. Thus, with only a slight increase in computational load, deeper and richer image features can be extracted, preserving more spatial information and significantly improving the model's detection performance.

4. Experiment

4.1 Experimental Environment Configuration

The model proposed in this paper, along with the models used for comparison, were all trained on a local GPU. Table 2 provides the specific configuration details. In the experiments conducted in this chapter, all object detection algorithms employed the SGD optimizer to adjust the learning rate during training, with the momentum parameter set at 0.937. The initial learning rate was set to 0.001, and weight decay was set to 0.0005. At the start of training, a Warmup strategy was implemented, with the warmup period lasting for three epochs. Additionally, an EarlyStopping strategy was applied to automatically halt training when the test set loss no longer decreased, thereby preventing overfitting. Throughout the training process, the batch size was set to 16, with a total of 200 training epochs.

Table 2. Experimental hardware and software configuration information.

Project	Detail
CPU	13th Gen Inter(R) Core(TM) i9-13980HX
GPU	NVIDIA GeForce RTX 4060 Laptop GPU
RAM	16GB
Operating system	64-bit Windows 11
PyTorch	2.1.0
CUDA	12.1
Python	3.8.16

4.2 Dataset

The Pascal Visual Object Classes (VOC) dataset[21] is one of the most widely recognized benchmark datasets in the field of computer vision. It is primarily used for tasks such as image classification, object detection, and image segmentation. Initially launched in 2005 under the leadership of the University of Oxford, the VOC dataset has undergone several expansions in subsequent years, including in 2007, 2010, and beyond. The dataset encompasses a rich variety of object categories and diverse image scenes, making it a popular choice for evaluating models' ability to detect objects of different classes in real-world scenarios. Key versions of the dataset include VOC2007, VOC2010, and VOC2012, which are commonly used for training and evaluating deep learning-based object detection algorithms.

The VOC dataset contains 20 object categories, such as people, animals, vehicles, and household items. These categories are presented in various complex backgrounds and lighting conditions, with high-quality annotations. Each image is annotated with bounding boxes for the objects and labels for their corresponding categories. For object detection tasks, the VOC dataset uses mean Average Precision (mAP) as the primary evaluation metric, which measures the overall performance of detection models across different categories.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0004

In object detection tasks, the VOC dataset is frequently employed for training and validating mainstream detection algorithms such as Faster R-CNN, YOLO, SSD, and others[22]. By training on the VOC dataset, these algorithms can effectively learn the features of various object categories and achieve high detection accuracy on the test sets. Due to the dataset's diversity and widespread use, it has played a crucial role in advancing object detection technology.

As deep learning technology continues to evolve, the VOC dataset has become an experimental benchmark for developing new object detection models, which is why this study uses the VOC dataset to implement object detection tasks.

4.3 Evaluation Metrics

To evaluate the performance of the target detection model, several key assessment metrics are used. These metrics provide insight into the model's effectiveness across different aspects and guide the selection of the most suitable model for specific scenarios. For instance, if a region exhibits various types of pavement damage, the mean Average Precision (mAP) metric becomes crucial. On the other hand, if the roads in the area are particularly hazardous, where any pavement damage could result in severe consequences, recall becomes more significant. The specific metrics used for evaluation are detailed in Table 3 below.

Table 3. Description of evaluation metrics for the YOLOv8-based model.

Name	Calculation Method
P (Precision)	$\frac{TP}{TP+FN}$
R (Recall)	$\frac{TP}{TP+FP}$
F1	$\frac{P \times R}{P+R} \times 2$
mAP (mean Average Precision)50	$\frac{1}{N}\sum_{i=1}^{N} AP_i$

In this context, TP (True Positive) refers to the number of instances that the model correctly identified as positive. FN (False Negative) denotes the number of positive instances that the model mistakenly classified as negative. FP (False Positive) indicates the number of negative instances that the model incorrectly labeled as positive. Here, N represents the total number of classes, and APi stands for the average precision across all classes.

4.4 Ablation Study

The purpose of the ablation study is to verify the positive impact of the improvement strategies on model training results and to test the synergistic interactions between different modules. This study includes three modules: ResNet18d, SPPF_LSKA, and EfficientHead. The results are shown in Table 4, which presents the experimental outcomes using different modules.

The experimental results show that when the ResNet18d module was integrated into the network alone, precision increased by 3.5%, recall improved by 3.3%, mAP@50 rose by 2.4%, and mAP@0.5:0.95 increased by 3.2%, indicating a significant improvement. When the SPPF_LSKA module was introduced on its own, precision improved by 0.5%, recall increased by 0.4%, mAP@50 rose by 0.5%, and mAP@0.5:0.95 improved by 0.7%, showing a modest enhancement. When both the ResNet18d and SPPF_LSKA modules were incorporated, precision improved by 4.1%, recall increased by 3.8%, mAP@50 rose by 2.8%, and mAP@0.5:0.95 increased by 3.8%. When the

ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0004

EfficientHead module was integrated into the network alone, precision slightly decreased, while recall improved by 0.5%, mAP@50 increased by 2.4%, and mAP@0.5:0.95 improved by 3.2%, showing a modest improvement.

Table 4. Results of the ablation study.

ResNet18d	SPPF_LSKA	EfficientHead	P	R	mAP@0.5	mAP@0.5-0.95
×	×	×	0.781	0.713	0.788	0.581
√	×	×	0.816	0.746	0.812	0.613
×	√	×	0.786	0.717	0.793	0.588
×	×	√	0.779	0.719	0.794	0.586
√	√	×	0.822	0.751	0.816	0.619
√	√	√	0.819	0.755	0.828	0.623

When all three improvements were introduced, the model's precision slightly decreased compared to the integration of just ResNet18d and SPPF_LSKA. However, compared to the original model, precision increased by 3.8%, recall improved by 4.2%, mAP@50 rose by 4%, and mAP@0.5:0.95 increased by 4.2%. This indicates that the combination of the ResNet18d, SPPF_LSKA, and EfficientHead modules allowed the model to better adapt to multi-scale object detection without suppressing each other's performance. The comparison of training mAP for all models is shown in Figure 5. It can be observed that the models with two modules (ResNet18d and SPPF_LSKA) and the model with all three improvements significantly outperform the other models. Furthermore, the model with all three improvements achieved the highest mAP@0.5:0.95 and demonstrated strong stability in the later stages of training.

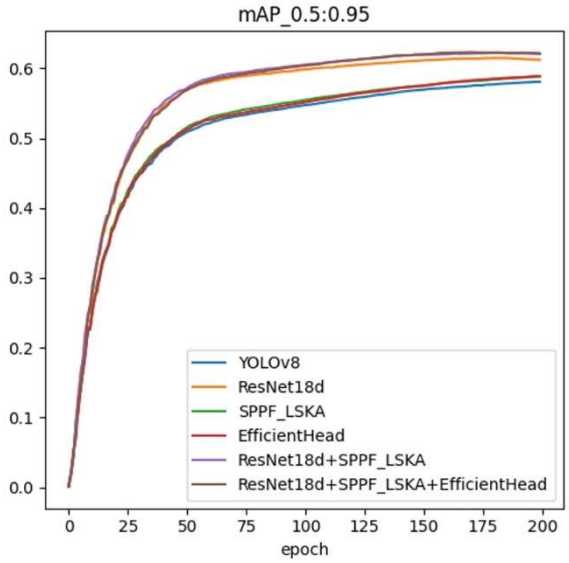

Figure 5. Variation curves of mAP@50:95 during training for different models.

5. Results

The experimental results show that our targeted improvements to the YOLOv8 model successfully addressed the challenges of poor performance in detecting small objects, multi-scale objects, and

ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0004

objects with backgrounds that are difficult to distinguish. The optimized ResEff-YOLO model performs exceptionally well in object detection tasks.

By integrating ResNet18d into the backbone architecture of YOLOv8, the receptive field and feature representation capabilities were enhanced. Additionally, we replaced the SPPF structure with the SPPF-LSKA structure by introducing deformable large convolution kernels and constructing 2D deep convolution layers using a series of consecutive 1D convolution kernels. This design improves the model's feature extraction ability without significantly increasing the model's parameters and computational complexity. Finally, EfficientHead was adopted as the detection head of YOLOv8, where two efficient modules were stacked to replace the original detection head's two 3x3 convolution layers, and the output was calculated through a standard convolution layer. This design allows the model to comprehend images at different scales without a substantial increase in complexity.

The final ResEff-YOLO model demonstrates significant improvements in detection performance compared to the original YOLOv8 mode. In future work, the research will focus on making the model more lightweight while maintaining detection speed, in order to better adapt to high-precision object detection tasks.

6. Conclusion

This paper proposes a novel object detection framework based on an improved YOLOv8 model. The backbone network and detection head of YOLOv8 have been optimized to expand the model's receptive field, enabling the network to capture more global context information and reduce the impact of complex backgrounds on detection results. Specifically, the original backbone has been replaced with ResNet18d, and the SPPF_LSKA module has been introduced to enhance feature extraction, particularly for multi-scale object detection. In the detection head, we designed and implemented EfficientHead, which reduces computational cost while further improving detection accuracy.

Experimental results on the PASCAL VOC dataset validate that the proposed model demonstrates significant performance improvements across various evaluation metrics. Specifically, compared to the original YOLOv8 model, our improved model shows a 3.8% increase in overall precision, a 4.2% increase in overall recall, a 4% improvement in mAP@50, and a 4.2% improvement in mAP@50:95, proving the effectiveness and superiority of our approach.This indicates that by optimizing the design of the backbone and detection head, we can effectively enhance overall performance in object detection tasks while maintaining model efficiency.

Future work will focus on further optimizing the model's inference speed and computational resource usage, as well as exploring the use of more advanced models as the foundation for training. Our goal is to apply the model in more real-world scenarios. In addition to improving detection accuracy, we will also focus on making the model even more lightweight-such as by reducing model size, minimizing floating-point operations, and enhancing computational efficiency to achieve end-to-end efficient and accurate detection. Moreover, we will explore ways to improve the model's robustness and generalization capabilities on larger datasets and in more complex environments, addressing more challenges in the field of object detection.

References

[1] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[2] Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).

[3] Redmon, J. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[4] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0004

[5] Thuan, D. (2021). Evolution of Yolo algorithm and Yolov5: The State-of-the-Art object detention algorithm.

[6] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[7] Ren, S. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497.

[8] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

[9] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (pp. 2980-2988).

[10] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9), 1904-1916.

[11] Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10781-10790)..

[12] Wang, H., Han, X., Song, X., Su, J., Li, Y., Zheng, W., & Wu, X. (2024). Research on automatic pavement crack identification Based on improved YOLOv8. International Journal on Interactive Design and Manufacturing (IJIDeM), 1-11.

[13] Amin, J., Shazadi, I., Sharif, M., Yasmin, M., Almujally, N. A., & Nam, Y. (2024). Localization and grading of NPDR lesions using ResNet-18-YOLOv8 model and informative features selection for DR classification based on transfer learning. Heliyon, 10(10).

[14] Zhao, L., Liang, G., Hu, Y., Xi, Y., Ning, F., & He, Z. (2024). YOLO-RLDW: An Algorithm for Object Detection in Aerial Images Under Complex Backgrounds. IEEE Access.

[15] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[16] J. Johnson, "CNN benchmarks," 2017.[Online]. Available: https://github.com/jcjohnson/cnn-benchmarks#readme

[17] Lau, K. W., Po, L. M., & Rehman, Y. A. U. (2024). Large separable kernel attention: Rethinking the large kernel attention design in cnn. Expert Systems with Applications, 236, 121352.

[18] Song, G., Liu, Y., & Wang, X. (2020). Revisiting the sibling head in object detector. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11563-11572).

[19] Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLO (Version 8.0.0) [Computer Software]. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 12 June 2024).

[20] Ge, Z. (2021). Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430.

[21] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. International journal of computer vision, 88, 303-338.

[22] Ahmad, T., Ma, Y., Yahya, M., Ahmad, B., Nazir, S., & Haq, A. U. (2020). Object detection through modified YOLO neural network. Scientific Programming, 2020(1), 8403262.