

# Implementation and Analysis of YOLOv8-UC: An Enhanced Architecture for Underwater Object Detection

CS22B002, CS22B037, CS22B053, CS22B060

*Department of Computer Science  
Indian Institute of Technology Tirupati*

**Abstract**—Underwater object detection presents unique challenges due to light attenuation, scattering, and complex environments, which often result in reduced detection accuracy. This paper presents the implementation and analysis of YOLOv8-UC, an improved underwater object detection algorithm based on YOLOv8. The architecture incorporates several enhancements: (1) a modified Dilation-wise Residual (DWR) C2f module to expand the receptive field for improved feature extraction, (2) Large Separable Kernel Attention (LSKA) integration with SPPF to reduce information loss during feature fusion, (3) a redesigned detection head utilizing RepConv to create a shared parameter structure, and (4) an Inner-SIoU loss function that employs auxiliary bounding boxes at different scales for improved bounding box regression. Our implementation and experiments on the URPC2019 dataset demonstrate that YOLOv8-UC achieves 82.2% mAP@0.5, while maintaining computational efficiency. The model shows particular effectiveness in detecting underwater biological targets like holothurians, echinus, and scallops in complex underwater environments.

**Index Terms**—underwater object detection, YOLOv8, deep learning, Dilation-wise Residual, Large Separable Kernel Attention, Inner-SIoU

## I. INTRODUCTION

Underwater object detection technology has become increasingly important in ocean exploration and marine monitoring applications. However, the complex underwater environment presents significant challenges for computer vision systems. Issues like light attenuation and scattering in water lead to low contrast, color distortion, and blurred images, resulting in reduced detection accuracy that often fails to meet practical requirements [1].

Traditional object detection approaches that rely on manually designed features have shown limited effectiveness in underwater scenarios due to the unique challenges of this environment. With the advancement of deep learning, more sophisticated models have been developed to address these issues. The YOLO (You Only Look Once) series of architectures has shown particular promise for real-time object detection tasks, including underwater applications.

In this paper, we focus on the implementation and analysis of YOLOv8-UC, an improved YOLOv8-based underwater object detection algorithm proposed by Huang et al. [1]. This model incorporates several architectural enhancements designed to address the specific challenges of underwater object detection:

- A modified Dilation-wise Residual (DWR) C2f module to enhance feature extraction from high-level expandable receptive fields
- Integration of Large Separable Kernel Attention (LSKA) with the SPPF module to improve multi-scale feature extraction capabilities
- A redesigned detection head using shared parameter structure with RepConv to reduce computational load
- An improved Inner-SIoU loss function that uses auxiliary boundaries at different scales to accelerate bounding box regression

Our implementation and experiments on the URPC2019 dataset demonstrate that these enhancements lead to significant improvements in detection accuracy while maintaining computational efficiency, making YOLOv8-UC particularly effective for detecting underwater biological targets in complex environments.

## II. RELATED WORK

### A. YOLO Series Development

The YOLO architecture has evolved significantly since its initial introduction, with each iteration bringing improvements in accuracy, speed, and feature extraction capabilities. The most recent variant, YOLOv8, introduces a new backbone network, detection head, and loss function, providing enhanced performance over previous versions [1]. The core concept of YOLOv8 involves segmenting input images and detecting objects within individual cells, followed by bounding box prediction and non-maximum suppression to generate final results.

### B. Underwater Object Detection

Several researchers have focused on adapting object detection models for underwater environments. Pan et al. [2] proposed an improved Faster R-CNN model incorporating a Hybrid Dilated CNN to address information loss for small targets. Feng and Jin [3] developed CEH-YOLO, which uses High-order Deformable Attention for underwater target detection. Liu et al. [4] introduced MarineYOLO specifically designed for small target detection in underwater environments.

Recent approaches have increasingly focused on attention mechanisms and feature enhancement techniques. Hou et al. [5] added a self-attention layer HorBlock module to YOLOv5s

to enhance feature extraction for sea cucumber and sea urchin detection. Zhuang and Liu [6] integrated Deformable Convolutional Networks into YOLOv7's backbone and added Contextual Transformer 3 to improve feature perception.

### C. Architectural Components

Several architectural innovations have proven effective for underwater object detection:

**Dilation-wise Residual (DWR):** This module uses a two-step method (regional residualization and semantic residualization) to efficiently capture multi-scale contextual information [1].

**Large Separable Kernel Attention (LSKA):** Proposed by Lau et al. [7], LSKA decomposes large 2D convolution kernels into more efficient 1D kernels to reduce computational complexity while maintaining effectiveness.

**Re-parameterized Convolution (RepConv):** This technique uses multi-branch convolutions during training and re-parameterizes them into a single branch during inference, reducing computational load while maintaining performance [1].

**Inner-SIoU Loss:** Zhang et al. [8] introduced this improved loss function that uses auxiliary bounding boxes of different scales to calculate loss, accelerating bounding box regression.

## III. METHODOLOGY

### A. Overall Architecture

Our implementation of YOLOv8-UC follows the architecture proposed by Huang et al. [1], as illustrated in Fig. 1. The model builds upon the YOLOv8n architecture with several key modifications:

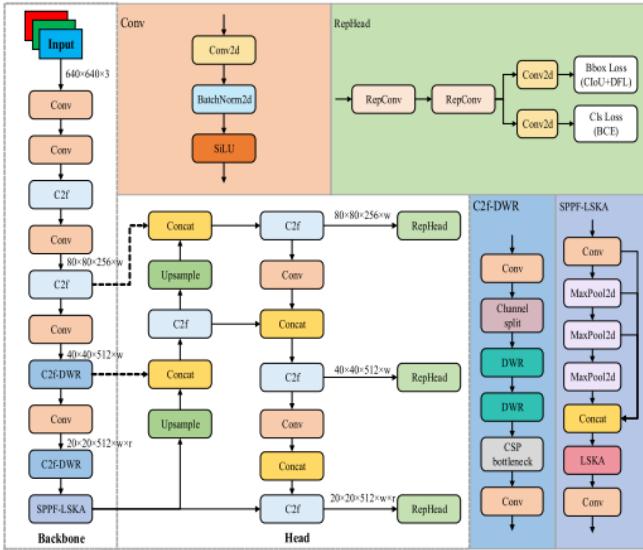


Fig. 1. The structure of YOLOv8-UC model

- The last two C2f modules in the backbone network are replaced with C2f-DWR modules
- The SPPF module is enhanced with LSKA to reduce information loss during feature fusion

- The detection head is redesigned with a shared parameter structure using RepConv
- The loss function is replaced with Inner-SIoU to improve convergence speed and generalization

### B. C2f-DWR Module

The Dilation-wise Residual (DWR) module is integrated into the C2f structure of YOLOv8, replacing the original Bottleneck. This modification enables the network to have a larger receptive field during high-level feature extraction, improving the efficiency of multi-scale information capture.

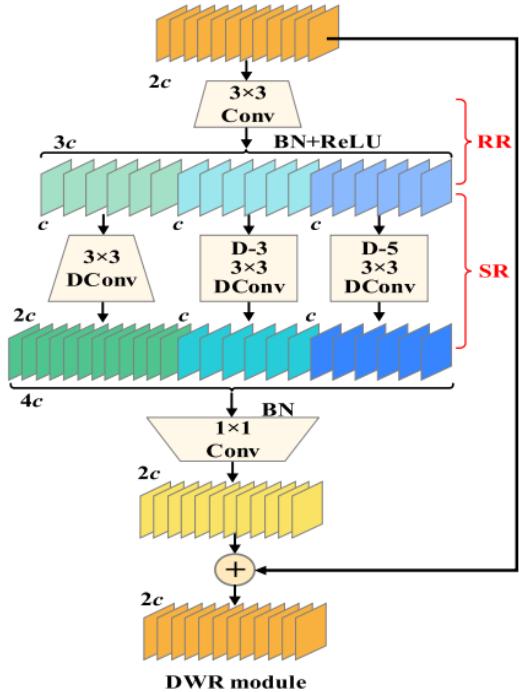


Fig. 2. The structure of DWR model

The DWR module employs a two-step approach for acquiring multi-scale contextual information:

- Regional Residualization:** This step generates concise feature maps with different region sizes, using  $3 \times 3$  convolutions combined with batch normalization and ReLU activation.
- Semantic Residualization:** Multi-rate dilated depth convolutions perform morphological filtering on the regional features, applying specific receptive fields to each channel.

This approach simplifies the learning process and preserves multi-scale contextual information more effectively. The final features are obtained by concatenating all feature maps, applying batch normalization, and using pointwise convolution to merge features.

### C. SPPF-LSKA Module

To address the potential loss of critical information in the standard SPPF layer of YOLOv8, we integrated the

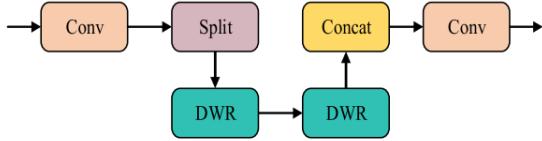


Fig. 3. C2f-DWR model

Large Separable Kernel Attention (LSKA) module. LSKA offers advantages in terms of long-range dependencies and spatial adaptability while reducing computational complexity. The key innovation of LSKA is the decomposition of large

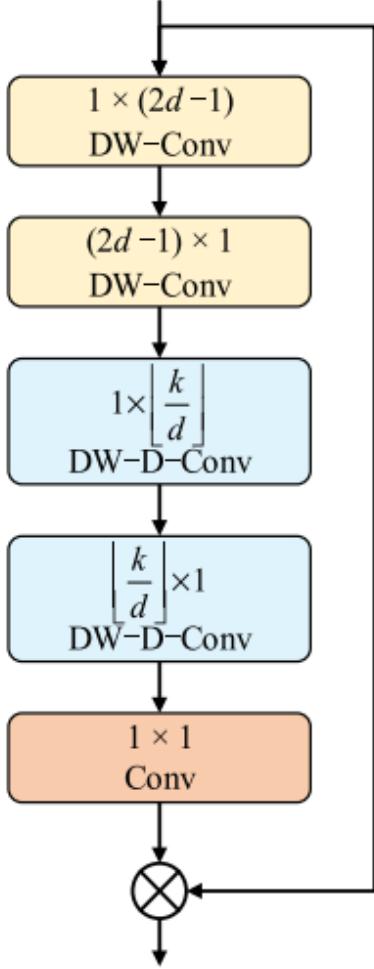


Fig. 4. The structure of LSKA

2D convolution kernels into two 1D kernels (horizontal and vertical), significantly reducing parameters and computational demands. Despite this simplification, LSKA maintains performance comparable to traditional approaches while effectively capturing key image features.

In our implementation, LSKA is integrated with SPPF to enhance multi-scale feature extraction, focusing more effec-

tively on important features during fusion and reducing the likelihood of losing critical information.

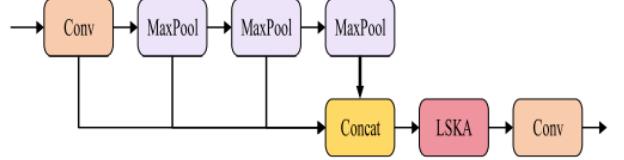


Fig. 5. SPPF-LSKA model

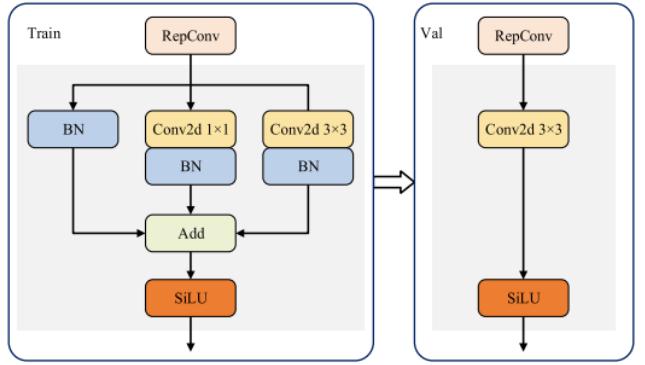


Fig. 6. The structure of RepConv.

#### D. RepHead with RepConv

To address parameter redundancy and computational load in the detection head, we implemented a shared parameter structure using re-parameterized convolution (RepConv).

The RepConv module employs a parallel structure during training, consisting of BatchNorm, 1x1 convolution, and 3x3 convolution to enrich multi-scale information. During inference, this multi-branch structure is fused into a single-branch structure through parameter re-parameterization, resulting in a single efficient 3x3 convolution operation.

Our implementation replaces the standard convolution in the original detection head with RepConv, creating what we call RepHead. This modification provides stronger multi-scale feature fusion capability and feature extraction ability while reducing the model's parameter count.

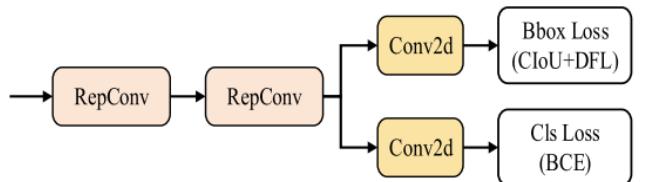


Fig. 7. RepHead model

### E. Inner-SIoU Loss Function

The original IoU-based loss functions have inherent limitations in adapting to diverse detection tasks. To address this, we adopt the Inner-SIoU loss function [8], which introduces scale-controlled auxiliary bounding boxes to differentiate regression samples.

Inner-SIoU dynamically adjusts the size of these auxiliary boxes:

- Smaller boxes for high IoU samples to accelerate convergence.
- Larger boxes for low IoU samples to aid learning.

#### Auxiliary Ground Truth Box:

$$b_{gt}^l = x_{gt}^c - \frac{w_{gt} \cdot r}{2}, \quad b_{gt}^r = x_{gt}^c + \frac{w_{gt} \cdot r}{2} \quad (1)$$

$$b_{gt}^t = y_{gt}^c - \frac{h_{gt} \cdot r}{2}, \quad b_{gt}^b = y_{gt}^c + \frac{h_{gt} \cdot r}{2} \quad (2)$$

#### Auxiliary Predicted Box:

$$b^l = x^c - \frac{w \cdot r}{2}, \quad b^r = x^c + \frac{w \cdot r}{2} \quad (3)$$

$$b^t = y^c - \frac{h \cdot r}{2}, \quad b^b = y^c + \frac{h \cdot r}{2} \quad (4)$$

#### Intersection:

$$\text{inter}_x = \min(b_{gt}^r, b^r) - \max(b_{gt}^l, b^l) \quad (5)$$

$$\text{inter}_y = \min(b_{gt}^b, b^b) - \max(b_{gt}^t, b^t) \quad (6)$$

$$\text{inter} = \text{inter}_x \cdot \text{inter}_y \quad (7)$$

#### Union and Inner-IoU:

$$\text{union} = A_{gt}^{aux} + A^{aux} - \text{inter} \quad (8)$$

$$\text{IoU}_{inner} = \frac{\text{inter}}{\text{union}} \quad (9)$$

#### Loss:

$$\mathcal{L}_{\text{Inner-SIoU}} = \mathcal{L}_{\text{SIoU}} + \text{IoU} - \text{IoU}_{inner} \quad (10)$$

Here,  $r$  = ratio, and  $A_{gt}^{aux} = w_{gt} \cdot h_{gt} \cdot r^2$ ,  $A^{aux} = w \cdot h \cdot r^2$  denote the areas of auxiliary ground truth and prediction boxes respectively. This formulation helps the model distinguish between easy and hard samples and enhances convergence.

## IV. EXPERIMENTS

### A. Dataset and Implementation Details

We used the URPC2019 dataset for model training and evaluation, which contains 5,786 underwater images across five categories: echinus, holothurian, starfish, scallop, and water weeds.

Our implementation was conducted on a DGX system with the following specifications:

- Operating System: Ubuntu
- GPU: 8x NVIDIA H100 Tensor Core GPUs
- CPU: Dual Intel Xeon Platinum 8480+
- Deep Learning Framework: PyTorch 1.13.1
- Python: 3.7

Training hyperparameters included a batch size of 16, initial learning rate of 0.01, and 200 training epochs.



Fig. 8. Partial images of the URPC dataset

### B. Evaluation Metrics

To evaluate model performance comprehensively, we used the following metrics:

- Precision (P): True Positives / (True Positives + False Positives)
- Recall (R): True Positives / (True Positives + False Negatives)
- Mean Average Precision (mAP): The average area under the precision-recall curves across all classes
- Parameters: Total number of model parameters

We report mAP@0.5 (IoU threshold of 0.5) and mAP@0.5:0.95 (IoU from 0.5 to 0.95 in increments of 0.05) to provide a comprehensive evaluation.

### C. Results and Analysis

Our study, shown in Fig 9, demonstrates the effectiveness of YOLOv8-UC. Compared to the YOLOv8n baseline, the full model achieves:

- +1.0% higher recall (78.2% vs. 77.2%)
- +1.7% higher mAP@0.5:0.95 (47.3% vs. 45.6%)

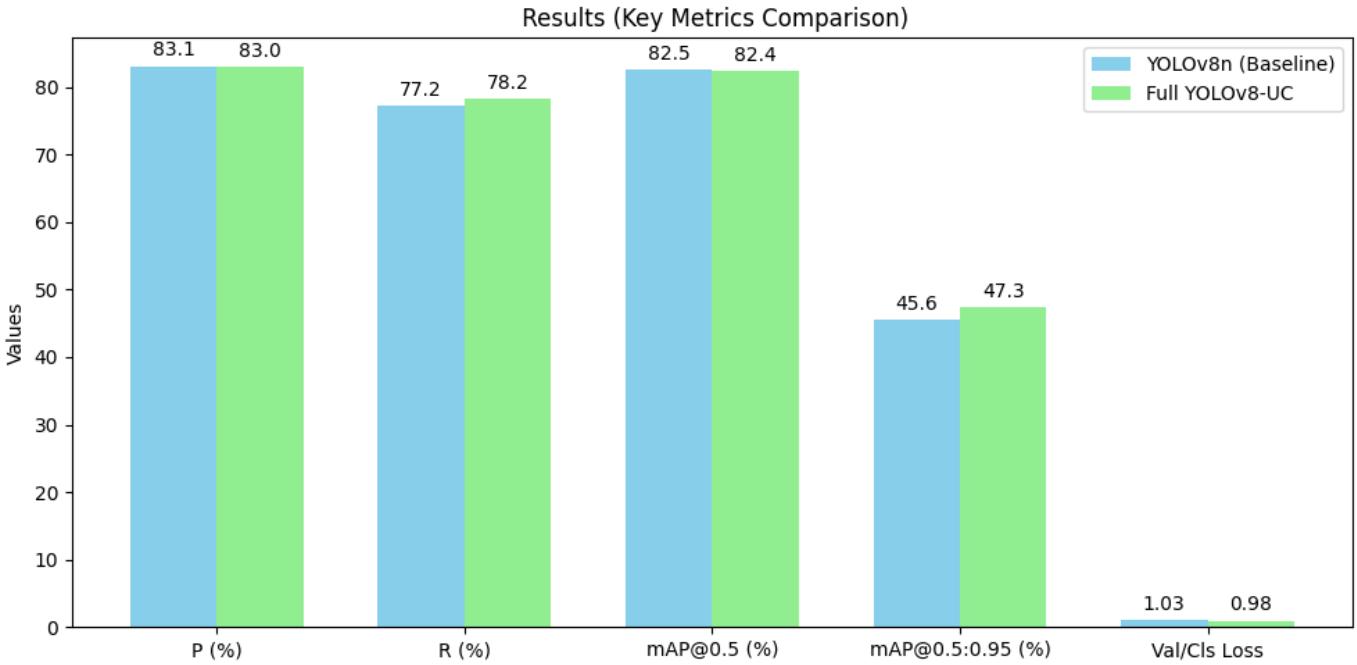


Fig. 9. Comparison of YOLOv8n (Baseline) and Full YOLOv8-UC on key performance metrics.

- Lower validation classification loss (0.98 vs. 1.03)

while maintaining nearly identical precision (83.0% vs. 83.1%) and mAP@0.5 (82.4% vs. 82.5%). This highlights improved robustness and class prediction stability, critical for real-world deployment.

Confusion matrix analysis shown in Fig 11 showed particularly strong performance for echinus (89% accuracy), starfish (83%), with slightly lower but still impressive results for scallop (71%).

Heatmap visualization demonstrated that YOLOv8-UC was more focused on the features of target objects themselves, ignoring unrelated feature details that often distracted the baseline model.

## V. CONCLUSION

In this paper, we presented the implementation and analysis of YOLOv8-UC, an enhanced underwater object detection architecture based on YOLOv8. Through the integration of the Dilation-wise Residual (DWR) module, Large Separable Kernel Attention (LSKA), RepConv, and Inner-SIoU loss function, our model effectively addresses the challenges of underwater object detection.

Our experimental results demonstrate that YOLOv8-UC achieves significant improvements in detection accuracy compared to the baseline YOLOv8n model. These improvements are achieved with only a modest increase in model parameters and computational requirements, maintaining the model's suitability for practical deployment.

The model shows particular effectiveness in detecting underwater biological targets like holothurians, echinus, starfish, and scallops in complex underwater environments. This makes

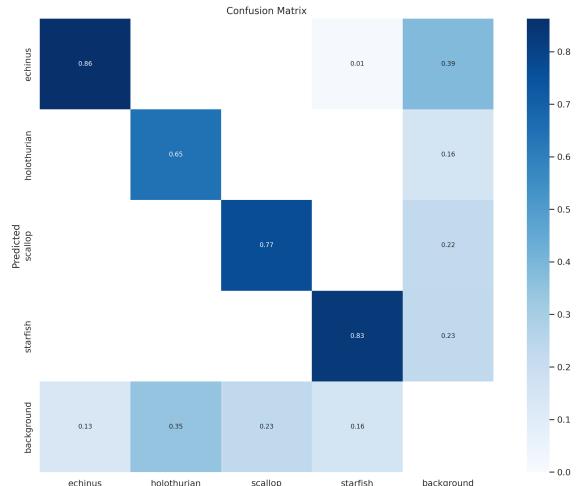


Fig. 10. Confusion Matrix for YOLOv8n

YOLOv8-UC particularly valuable for applications in ocean exploration, marine biology research, and underwater environmental monitoring.

Future work could explore further refinements to the architecture, adaptation to additional underwater scenarios, and potential integration with underwater image enhancement techniques to further improve detection performance in extremely challenging conditions.

## REFERENCES

- [1] J. Huang, C. Fang, X. Zheng, and J. Liu, "YOLOv8-UC: An Improved YOLOv8-Based Underwater Object Detection Algorithm," *IEEE Access*,

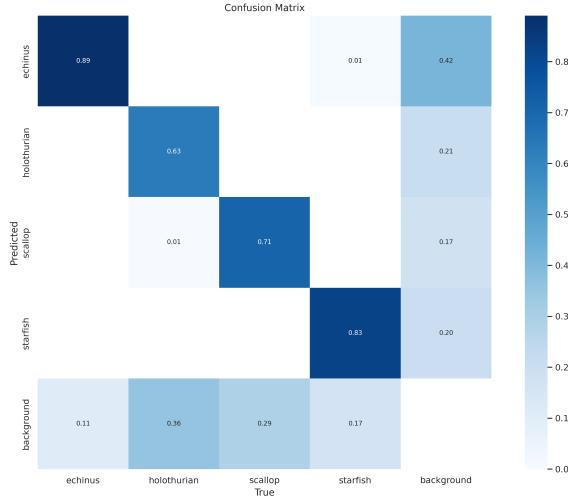


Fig. 11. Confusion Matrix for YOLOv8-UC

- vol. 12, pp. 172186-172195, 2024.
- [2] H. Pan, H. Zhang, X. Lei, F. Xin, and Z. Wang, “Hybrid dilated faster RCNN for object detection,” *J. Intell. Fuzzy Syst.*, vol. 43, no. 1, pp. 1229-1239, Jan. 2022.
  - [3] J. Feng and T. Jin, “CEH-YOLO: A composite enhanced YOLO-based model for underwater object detection,” *Ecological Informat.*, vol. 82, Sep. 2024, Art. no. 102758.
  - [4] L. Liu, C. Chu, C. Chen, and S. Huang, “MarineYOLO: Innovative deep learning method for small target detection in underwater environments,” *Alexandria Eng. J.*, vol. 104, pp. 423-433, Oct. 2024.
  - [5] C. Hou, Z. Guan, Z. Guo, S. Zhou, and M. M. Lin, “An improved YOLOv5s-based scheme for target detection in a complex underwater environment,” *J. Mar. Sci. Eng.*, vol. 11, no. 5, p. 1041, May 2023.
  - [6] H. Zhuang and S. Liu, “Underwater biological target detection algorithm and research based on YOLOv7 algorithm,” *IAENG Int. J. Comput. Sci.*, vol. 51, no. 6, pp. 594-601, Jun. 2024.
  - [7] K. W. Lau, L.-M. Po, and Y. A. U. J. E. S. W. A. Rehman, “Large separable kernel attention: Rethinking the large kernel attention design in CNN,” *Expert Syst. With Appl.*, vol. 236, Feb. 2024, Art. no. 121352.
  - [8] H. Zhang, C. Xu, and S. Zhang, “Inner-IoU: More effective intersection over union loss with auxiliary bounding box,” arXiv:2311.02877, 2023.