Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

YOLO-DC: Integrating deformable convolution and contextual fusion for high-performance object detection

Dengyong Zhang [a], Chuanzhen Xu [a], Jiaxin Chen [a], Lei Wang [b,*], Bin Deng [c]

[a] School of Computer Science and Technology, Changsha University of Science and Technology, Changsha, 410114, China
[b] School of Civil Engineering, Changsha University of Science and Technology, Changsha, 410114, China
[c] School of Hydraulic and Environmental Engineering, Changsha University of Science and Technology, Changsha, 410114, China

ARTICLE INFO

Keywords:
Object detection
YOLO
Deformable convolutions
Contextual mechanisms
Deep learning

ABSTRACT

Object detection is a fundamental task in computer vision, but existing methods often concentrate on optimizing model architectures, loss functions, and data preprocessing techniques, while frequently neglecting the potential improvements that advanced convolutional mechanisms can provide. Additionally, increasing the depth of deep learning networks can lead to the loss of essential feature information, highlighting the need for strategies that can further improve model accuracy. This paper introduces YOLO-DC, an algorithm that enhances object detection by incorporating deformable convolution and contextual mechanisms. YOLO-DC integrates a Deformable Convolutional Module (DCM) and a Contextual Information Fusion Downsampling Module (CFD). The DCM employs deformable convolution with multi-scale spatial channel attention to effectively expand the receptive field and enhance feature extraction. In parallel, the CFD module leverages both contextual and local features during downsampling and incorporates global features to enhance joint learning and reduce information loss. Compared to YOLOv8-N, YOLO-DC-N achieves a significant improvement in Average Precision (AP), increasing by 3.5% to reach 40.8% on the Microsoft COCO 2017 dataset, while maintaining a comparable inference time. The model outperforms other state-of-the-art detection algorithms across various datasets, including the RUOD underwater dataset and the PASCAL VOC dataset (VOC2007 + VOC2012). The source code is available at https://github.com/Object-Detection-01/YOLO-DC.git.

1. Introduction

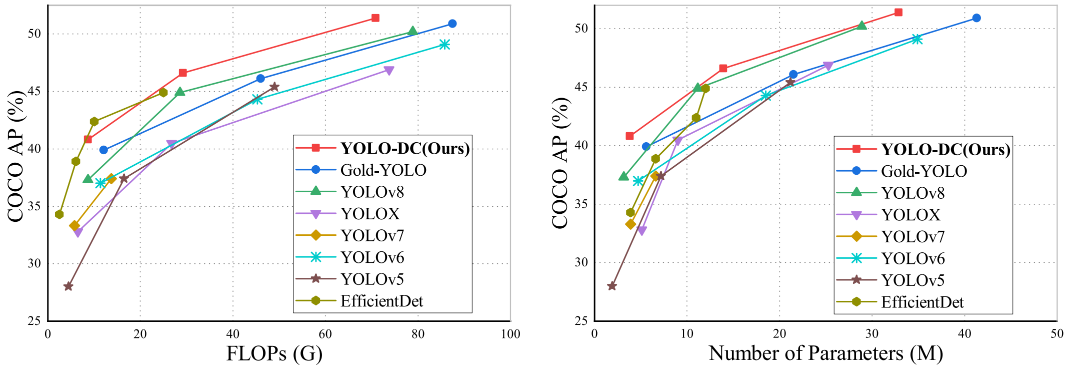
Object detection, as a fundamental task in computer vision, has witnessed transformative advancements through deep learning techniques, with applications spanning healthcare, security, and autonomous systems. The core challenge in modern detection frameworks lies in balancing detection accuracy and inference speed — requirements that often present conflicting optimization objectives across different application scenarios. While real-time systems like autonomous driving prioritize computational efficiency, domains such as medical imaging demand meticulous attention to detection precision. This inherent tension necessitates architectures capable of dynamically adapting to diverse operational requirements.

The current landscape of object detection architectures presents two distinct paradigms. Transformer-based models, exemplified by DETR [1], DINO [2], and RT-DETR [3], demonstrate superior performance in capturing long-range dependencies through self-attention mechanisms. However, their computational complexity imposes limitations on real-time applications. Conversely, CNN-based architectures like the YOLO series achieve remarkable speed–accuracy trade-offs through progressive architectural innovations. From the seminal YOLOv1 [4] to subsequent iterations, evolutionary improvements have focused on three primary axes: backbone optimization (DarkNet [5], CSPNet [6], ELAN [7]), feature fusion mechanisms (SPP [8], PANet [9], RepGFPN [10]), and loss function refinement [8,11,12]. The culmination of these efforts, YOLOv8 [13], integrates cross-domain advancements while maintaining computational efficiency.

Recent advances in hybrid architectures demonstrate the synergistic potential of cross-paradigm integration. Chen et al. [14] established that Transformer-CNN fusion architectures significantly improve small object detection through complementary attention mechanisms. Notably, the U-ATSS framework [15] achieves dual optimization of receptive field expansion and computational efficiency via learnable spatial aggregation modules. These developments find particular resonance

[*] Corresponding author.
E-mail addresses: zhdy@csust.edu.cn (D. Zhang), chuanzhen@stu.csust.edu.cn (C. Xu), jxchen@csust.edu.cn (J. Chen), LeiWang@csust.edu.cn (L. Wang), dengbin07@csust.edu.cn (B. Deng).

https://doi.org/10.1016/j.image.2025.117373
Received 19 December 2024; Received in revised form 21 April 2025; Accepted 6 June 2025
Available online 21 June 2025



(a) (b)

Fig. 1. Comparison with Other State-of-the-Art Object Detection Models on the COCO Dataset: (a) AP performance vs. computational volume (FLOPs); (b) AP performance vs. number of parameters. YOLO-DC-N achieves 40.8% AP with 8.9G FLOPs and 3.9M parameters on the MS COCO dataset. These results clearly demonstrate that the proposed YOLO-DC model strikes an optimal balance between performance and computational efficiency. Notably, the Faster R-CNN, RetinaNet, and DETR-DC models are excluded from this comparison due to their significantly higher computational and parameter requirements. However, YOLO-DC outperforms these models, as detailed in Table 1.

in medical imaging research: Özçelik et al. [16] demonstrated that chaotic swarm optimization coupled with LSTM networks effectively resolves nonlinear feature dynamics in diabetic retinopathy classification, while Çiçek et al. [17] validated anatomical modeling through fractal dimension analysis in orthodontic growth detection. Crucially, both medical studies employ multiscale wavelet decomposition to substantiate that coordinated dynamic feature selection and contextual interaction enhance model robustness in complex scenarios. These findings collectively underscore the theoretical validity of integrating deformable convolution mechanisms (adaptive spatial deformation) with contextual fusion strategies (cross-scale feature correlation) in detection tasks.

Despite notable advancements in convolutional techniques, as demonstrated by YOLO-MS [18] which employs depth-wise convolution to enhance accuracy with parameter efficiency, critical limitations persist: conventional architectures using fixed geometric structures struggle to handle deformation and occlusion due to rigid receptive fields, while existing implementations of deformable convolution underutilize dynamic alignment through simplistic nesting. Furthermore, progressive information degradation in deep residual networks induces gradient flow bias during spatial transformations [19], ultimately compromising detection accuracy and contributing to suboptimal performance observed in YOLO-based architectures.

While convolutional enhancement techniques have made significant strides, substantial opportunities remain for further refinement. For instance, YOLO-MS [18] demonstrates the feasibility of advanced convolution methods by employing depth-wise convolution to enhance accuracy while optimizing parameter efficiency.

However, conventional convolution-based architectures with fixed geometric structures face inherent limitations in handling target deformation and occlusion within complex scenes. This stems primarily from their rigid receptive fields, which struggle to capture discriminative features of morphologically variable targets. Existing implementations of deformable convolutions often adopt superficial nested designs, thereby underutilizing their dynamic feature alignment capabilities.

Furthermore, as residual networks grow deeper [19], progressive information degradation during feature extraction and spatial transformation introduces gradient flow bias during model optimization. Such bias inevitably compromises detection accuracy, ultimately contributing to the suboptimal performance observed in YOLOv8-based architectures.

To address these challenges, this paper proposes YOLO-DC, which builds upon the YOLOv8 architecture by integrating a Deformable Convolutional Module (DCM) to improve the receptive field and a Contextual Information Fusion Downsampling (CFD) Module to enhance information utilization. These innovations collectively address the limitations of traditional methods in handling complex geometric transformations and information loss. Compared to other state-of-the-art (SOTA) models, as depicted in Fig. 1, YOLO-DC demonstrates superior performance, achieving an optimal balance between computational load and parameter count. The main contributions of this paper include:

1. We introduce the Deformable Convolution Module (DCM) as a key convolutional component for feature extraction in the model. DCM employs an enhanced deformable convolution technique that surpasses conventional methods by integrating Multiscale Spatial Channel Attention (MSCA). This integration improves the generation of deformable convolutional offsets, extends the receptive field, and enhances feature extraction capabilities.
2. We present the Contextual Information Fusion Downsampling (CFD) module, designed to integrate contextual information and reduce the loss of useful information and the redundancy of irrelevant information after downsampling, thereby enhancing information utilization. The CFD module fuses contextual information, facilitating the learning of joint features from both local and surrounding contexts.
3. We employ the DCM and CFD modules to enhance the baseline model, introducing a new object detector, YOLO-DC, which outperforms existing object detection algorithms (e.g., YOLOv8 and Gold-YOLO) with comparable parameters and computational load.

2. Related work

2.1. Object detectors

Over the years, the YOLO model has evolved significantly, making considerable progress in real-time object detection. Beginning with YOLOv1, subsequent versions improved various aspects such as the backbone network, data augmentation strategies, and loss functions, which led to enhanced speed and accuracy. YOLOv2 [20] and YOLOv3 [5] improved model accuracy through the introduction of the anchor mechanism and multiscale detection, all while maintaining high speed. YOLOv4 further reduced computational overhead and increased accuracy by introducing the CSPNet structure and incorporating techniques like Mosaic data augmentation and Self-Adversarial Training (SAT) to enhance model robustness [8]. YOLOv7 introduced a series of training techniques that improved performance without increasing inference costs [7]. YOLOv8 [13] integrated many of the advancements made in the YOLO family, establishing itself as a SOTA model. Additionally, Gold-YOLO introduced a new aggregation-distribution mechanism

(GD), which facilitated efficient information exchange by fusing multi-layer features and injecting global information into higher layers [21]. YOLOX [22] and YOLOv10 [23] optimized the label assignment strategy, further enhancing inference speed.

YOLOv8, a significant update over YOLOv5, incorporated new features and enhancements that built upon the successes of its predecessors. The backbone, based on the Cross Stage Partial Network (CSPNet) concept [6], utilized Darknet-53. Unlike YOLOv5, YOLOv8 replaced the C3 module with the C2f module, aiming to maintain a lightweight structure while enhancing gradient flow richness by combining the strengths of the C3 module from YOLOv5 and the ELAN module from YOLOv7 [7]. In the neck of the YOLOv8 model, the Feature Pyramid Network with Path Aggregation Network (PAN-FPN) was employed. Unlike YOLOv5, this model adopted a decoupled head structure that segregated the classification and detection heads. The classification loss was determined by Binary Cross-Entropy Loss (BCE), while the rectangular box regression loss incorporated Generalized Focal Loss (DFL) and CIoU loss. YOLOv8 embraced an anchor-free approach in its architecture, deviating from the anchor-based approach utilized in earlier YOLO versions. It also used a task-aligned dynamic label assignment strategy [24] instead of the edge-length proportional matching method.

Despite significant advancements in YOLOv8, there is still potential for improvement, particularly in the area of convolution and in addressing the issue of information loss, which affects model accuracy. This paper adopts YOLOv8 as the baseline model, aiming to explore potential improvements in convolutional techniques and contextual information utilization to enhance model accuracy within an acceptable computational cost.

2.2. Deformable convolution

In computer vision, addressing the geometric transformations of the same object in different scenes and perspectives has been a significant challenge. Deformable Convolution (DCNv1) was developed to tackle this issue [25]. DCNv1 used additional convolutional operations to learn the offset of each sample point, controlling the deformation of the receptive field. However, this approach resulted in the inclusion of many irrelevant regions within the receptive field, introducing redundancies that affected feature extraction. DCNv2 built upon DCNv1 by generating a weight coefficient mask for each sample point to adjust the offsets, thereby refining the accuracy of the receptive field and enhancing its adaptability to geometric variations [26]. DCNv3 extended DCNv2 by adding weight sharing and multi-group mechanisms [27]. More recently, DCNv4 further enhanced the convolution operation by removing soft-max normalization and optimizing memory access [28]. However, DCNv3 and DCNv4 are more suited for large models, and both operators currently present deployment challenges. At the same time, the inherent complexity of DCNv3 and DCNv4 operator deployments limits their application.

The application of deformable convolutional in object detection has witnessed remarkable progress. Huang et al. [29] pioneered the integration of DCNv1 into the YOLO architecture, which dynamically adjusts the sampling point distribution of convolutional kernels, thereby significantly enhancing feature extraction capabilities for non-rigid targets. Building on this foundation, Dong et al. [30] embedded DCNv2 modules into the neck network of YOLOv8s, achieving a mean Average Precision (mAP) of 74.0% in urban street scene detection tasks, thereby validating its superiority in complex spatial distribution scenarios. Recently, Yuan et al. [31] further incorporated DCNv3 into the DiffusionDet framework, leveraging its enhanced nonlinear modeling capabilities to attain a 3.6% AP gain on the COCO dataset. Notably, addressing the unique challenges of underwater object detection, Yuan et al. [32] deployed DCNv2 modules to refine the feature pyramid structure of YOLO, successfully reducing the false negative rate for coral reef organisms by 2.6 percentage points in their proprietary dataset.

While these studies demonstrate the potential of DCNs across diverse detection scenarios, existing methodologies exhibit two critical limitations: First, most works merely treat deformable convolutions as plug-and-play modules without systematic analysis of dynamic variation patterns in receptive fields [26,27]. For instance, the distribution characteristics of deformation parameters across different network depths remain underexplored. Second, insufficient attention has been paid to the feature spatial shift problem induced by adaptive receptive field adjustment – specifically, the risk of deformable sampling points exceeding semantically valid regions – which becomes particularly pronounced in underwater scenarios with occluded or low-contrast targets [28].

In our proposed Deformable Convolution Module (DCM), we address these limitations by redesigning the receptive field control mechanism through a multi-scale spatial channel attention. This innovation enhances spatial-semantic consistency during deformation processes by explicitly constraining the probability of sampling point deviations beyond target bounding boxes, thereby effectively mitigating feature spatial shift issues. Detailed implementation and validation are provided in Section 4.2.

2.3. Attention and context mechanisms

Attention and context mechanisms have become crucial for improving model performance [33]. Many attention mechanisms utilize contextual information. For instance, SK-Net [34] used parallel convolutional branches with different receptive fields for multiscale feature weighting. GCNet, SCNet, GsopNet, and CCNet employed non-local mechanisms to capture diverse spatial information [35–38]. CA [39] enhanced feature representation by decomposing 2D global pooling into two 1D processes that captured positional and channel relationships. EMA [40] reshaped channels into batch dimensions for uniform spatial semantic features. LSKA [41] combined separable convolution with attention, retaining the advantages of large kernels while reducing computational overhead. GAAM used a Gaussian probability density function to dynamically adjust focus and better capture relevant information [42].

In model architecture, DPC used architectural search techniques for efficient multi-scale architectures [43]. GCNet introduced a global context module, improving image classification and object detection [35]. CGNet combined local features with environmental context, enhancing feature learning [44]. F-SSD improved small object detection by fusing multiscale features and using additional features [45].

3. Effectiveness of optimization strategies

This paper focuses on optimizing often-overlooked aspects of convolutional enhancement and information utilization in YOLOv8. This section evaluates the effectiveness of these proposed optimizations.

To improve the ability of the model to extract features, the focus is placed on enhancing convolutional techniques. In recent years, various convolution methods, such as group convolution, dilated convolution, depthwise separable convolution, and deformable convolution, have been explored to balance performance and efficiency. Advances in hardware technology enable the use of large kernel convolutions in models like YOLO-MS [18], LSKA [41], and the ConvNeXt series [46, 47], yielding notable results. However, deformable convolution is particularly effective for expanding the receptive field and improving feature extraction, making it well-suited for object detection tasks. Conventional convolutional modules in neural networks have fixed geometric structures, limiting their ability to model geometric transformations. This limitation often leads to suboptimal performance when recognizing targets with complex geometric transformations, as seen in baseline models that rely on traditional convolution. Deformable

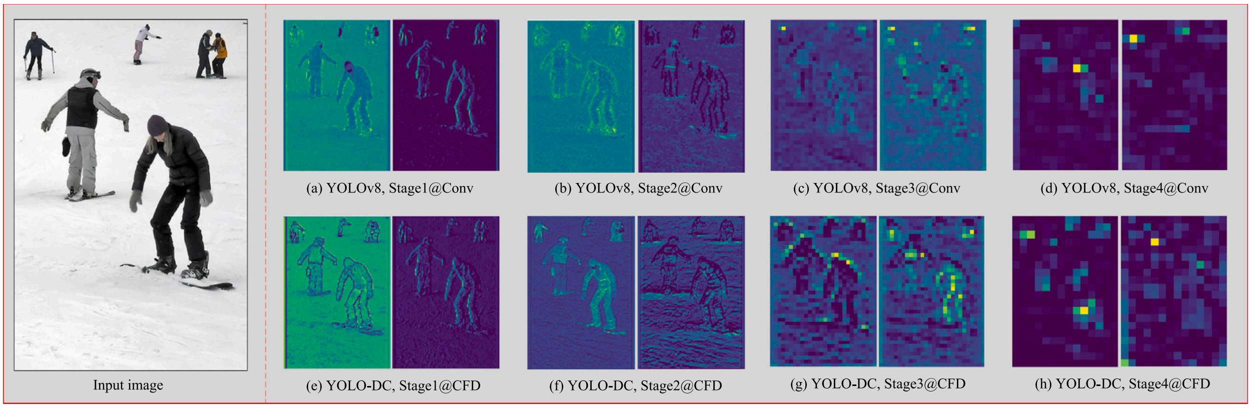


Fig. 2. Visualization of the output feature maps at each stage of the model: (a)–(d) show downsampling layer outputs from stages 1 to 4 of the YOLOv8 backbone; (e)–(h) show CFD module outputs from stages 1 to 4 of the YOLO-DC backbone.

convolution (DCN), particularly DCNv2, addresses this limitation by generating offsets for each sampling point through additional learning, refining these offsets with a mask of weight coefficients to better adapt to geometric variations [26]. For example, given a convolution kernel with K sampling positions, let w_k and p_k represent the weights and pre-specified offsets of the kth position, respectively. Define a 3 × 3 convolutional kernel with a dilation factor of 1, where $x(p)$ and $y(p)$ represent features at position p from the input and output feature maps, respectively, and Δp_k and Δm_k denote the learnable offsets and weight masks. The DCNv2 operation can be expressed as shown in Eq. (1), effectively extending the receptive field to enhance feature extraction.

$$y(p) = \sum_{K=1}^{K} w_k \cdot x(p + p_k + \Delta_{p_k}) \cdot \Delta_{m_k} \tag{1}$$

Initially, we replace traditional convolution with DCNv2 convolution to optimize the baseline model, which indeed improves performance. However, the ability of DCNv2 to learn the offset Δp_k is limited by relying on a single convolutional learning process to acquire the offset. Specifically, the Δp_k is obtained by convolving the original feature layer. For instance, if the input feature layer has dimensions $w \times h \times c$, a convolution operation is first performed to obtain an offset field with dimensions $w \times h \times 2c$, where $2c$ represents the offsets in the x and y directions. This approach, which relies solely on convolution, is prone to errors and may include irrelevant regions, limiting its ability to accurately deform the receptive field and reducing its effectiveness across different scenes. Subsequent experiments confirm these challenges, indicating room for further improvement in DCNv2. To address this, the offset generation method is redesigned, introducing the Multi-scale Spatial Channel Attention (MSCA) mechanism, and the DCN-MSCA method is proposed for more accurate receptive field expansion. This leads to the development of the DCM module, with detailed improvements described in Section 4.2.

To enhance information utilization and reduce information loss, the focus is also placed on improving the downsampling stage and introducing contextual information mechanisms. The spatial transformation process in models inevitably causes information loss and introduces redundant information, leading to suboptimal performance. Specifically, during downsampling, surrounding contextual semantic information is combined with local features to generate joint features, allowing the module to learn relevant information about each object and its spatial environment. Additionally, global contextual information and channel weighting are introduced to further enhance joint feature learning, suppress redundant data, and emphasize useful information. The CFD module is developed based on this approach, with detailed improvements outlined in Section 4.3.

In the baseline model, the downsampling module operates throughout the network. We replace it with the CFD module, enabling the capture of contextual information at both spatial and semantic levels during downsampling, thereby improving information utilization compared to the direct use of convolutional downsampling in YOLOv8. Fig. 2 shows the visualized feature maps for each stage of the model. Compared to the baseline model's downsampled version, the model with the addition of the CFD module extracts more coherent texture information and produces features that are more focused on the object. These features emphasize the overall contour rather than just the local details. This indicates that the CFD module effectively integrates information, emphasizes useful data, and suppresses redundant information. Furthermore, since the downsampling module is located before the convolution module responsible for feature extraction, it integrates information before spatial transformation, reducing the generation of redundant information.

4. Method

4.1. Model architecture

This paper introduces YOLO-DC, a novel object detector built on the YOLOv8 model. YOLO-DC integrates a series of optimizations and enhancements that leverage the technological advances from previous iterations of the YOLO family, aiming to excel in object detection tasks. The success of YOLO-DC is mainly due to the DCM and the CFD modules. The DCM provides superior feature extraction capabilities, while the CFD module effectively reduces information loss during downsampling. The network architecture of YOLO-DC (as illustrated in Fig. 3) seamlessly integrates these components to enhance object detection accuracy. Drawing inspiration from the computational efficiency principles proposed by Altan et al. in plant disease monitoring [48] and industrial defect analysis [49], our architecture employs deformable convolution operators with adaptive receptive field learning to capture geometric variations, while strategically integrating contextual attention mechanisms that eliminate spatial redundancy through feature importance weighting. This co-design achieves superior detection accuracy while maintaining competitive inference speeds.

The improvement strategy involves selecting YOLOv8 as the baseline model and replacing its downsampling operation with the CFD module. Additionally, within the backbone, the C2f module is substituted with the DCM module. The DCM module significantly enhances model performance by offering robust feature extraction and a superior receptive field compared to conventional convolutional modules. Furthermore, the CFD module, typically positioned before the DCM module, leverages collaborative context information to enhance features, which are then immediately processed by the DCM module for feature extraction, thereby reducing information loss.

YOLO-DC is available in three versions to cater to different scenarios: N (lightweight), S (general), and M (high accuracy). The dimension

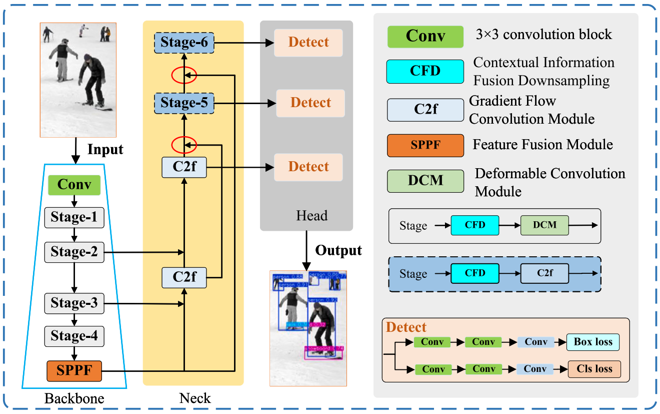


Fig. 3. Illustration of YOLO-DC network.

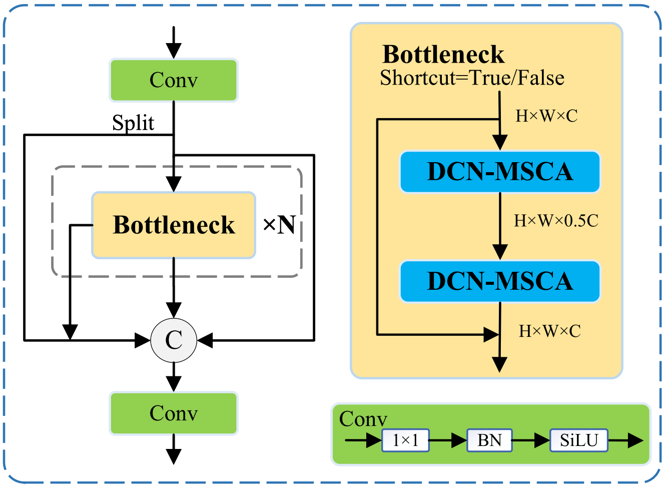


Fig. 4. Illustration of DCM Module.

scaling factor (w) for these versions is (0.25, 0.5, 0.75), the tensor scaling factor (d) is (0.33, 0.33, 0.67), and the ratio scaling factor (r) is (2, 2, 1.5), respectively. The N version is designed for lightweight scenarios with minimal hardware requirements, the S version is intended for general use, and the M version, with increased computational load and parameters, is ideal for high-precision scenarios with advanced hardware.

4.2. Deformable Convolution Module (DCM)

The Deformable Convolution Module (DCM) builds on the principles of the C2f module, which maintains a rich gradient flow, and CSPNet [6], known for enhancing convolutional network learning capabilities. The DCM module performs a convolution operation, followed by a split that divides the input into two branches. One branch passes through several Bottleneck modules with residual connections, which are part of the backbone, while the other branch remains unchanged. These branches are then merged along with the residuals from the Bottleneck module, ensuring robust feature extraction while maintaining a lightweight structure. The main convolution operator in DCM uses the proposed DCN-MSCA, providing a larger receptive field and more accurate feature extraction compared to standard convolution and DCN. This improvement is crucial for enhancing accuracy in object detection tasks, as illustrated in Fig. 4.

To accurately extend the convolutional receptive field and improve feature extraction capabilities, we redesigned the offset generation process of the DCNv2 convolution by introducing the MSCA mechanism. This mechanism improves offset attention in both the X and Y directions, allowing for more accurate receptive field deformation and reducing the inclusion of irrelevant regions. The enhanced DCN-MSCA method integrates the MSCA mechanism into the offset generation process, resulting in improved feature extraction capabilities, as shown in part a on the left side of Fig. 5.

The MSCA mechanism extends the Coordinate Attention (CA) approach to address information loss through multiscale fusion. While CA decomposes global pooling into one-dimensional feature encodings in two spatial directions to form orientation-aware feature maps, its direct decomposition using two global pooling operations can be somewhat limited. The subsequent single splicing-then-convolution operation often fails to facilitate adequate information exchange, leading to suboptimal performance in certain models. Our MSCA mechanism addresses these issues by integrating multiscale information to compensate for losses during transformation, enhancing both spatial and channel information.

The proposed MSCA mechanism operates through three main branches, each handling multiscale information at different gradients. The first branch transforms global pooling into one-dimensional codes for the X and Y directions. The second branch convolves these codes, extracting X_{weight} and Y_{weight} after Sigmoid activation to refine fusion accuracy. The third branch performs global pooling on the initial inputs, with its output guiding the encoded information from the second branch. The final output is obtained by combining information from all three branches, ensuring accurate and robust feature extraction. The schematic is shown in part b on the left side of Fig. 5.

In MSCA attention, decomposing global pooling into one-dimensional coding operations can be represented by Eq. (2), where Z_c denotes the output in the height or width direction, H and W represent the height and width, i and j are specific channels, and x_c denotes the input feature map.

$$Z_c = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W} x_c(i,j) \tag{2}$$

For a given input x_c, each channel is initially encoded along the horizontal and vertical coordinates using pooling operation dimensions of (H,1) or (1,W). Therefore, the outputs in the height and width directions can be represented by Eqs. (3) and (4), where $Z_c^h(h)$ and $Z_c^w(w)$ denote the outputs when the height of the given channel is h and the width is w, respectively.

$$Z_c^h(h) = \frac{1}{W}\sum_{0\leq i\leq W} x_c(h,j) \tag{3}$$

$$Z_c^W(w) = \frac{1}{H}\sum_{0\leq j\leq H} x_c(j,w) \tag{4}$$

The obtained outputs in the width and height directions are combined to create a tensor with the same number of channels as the input. This tensor then undergoes convolution, passes through the activation function, and is separated into outputs g_c^h and g_c^w in the X and Y directions, respectively. Simultaneously, the tensor is split into weighting coefficients H_c^h and W_c^w, which guide the outputs in the X and Y directions. This can be expressed in Eq. (5) and Eq. (6), where $g_c'^h$ and $g_c'^w$ denote the outputs in the height and width directions, respectively, after the fusion of the weight coefficients.

$$g_c'^h(i) = g_c^h(i) \times W_c^h(i) \tag{5}$$

$$g_c'^w(j) = g_c^w(j) \times W_c^w(j) \tag{6}$$

Therefore, the output of the MSCA attention can be expressed as Eq. (7), where x_c denotes the feature map of the original input, and a_c represents the globally complemented information from global pooling.

$$y_c(i,j) = x_c(i,j) \times g_c'^h(i) \times g_c'^w(j) \times a_c(i,j) \tag{7}$$

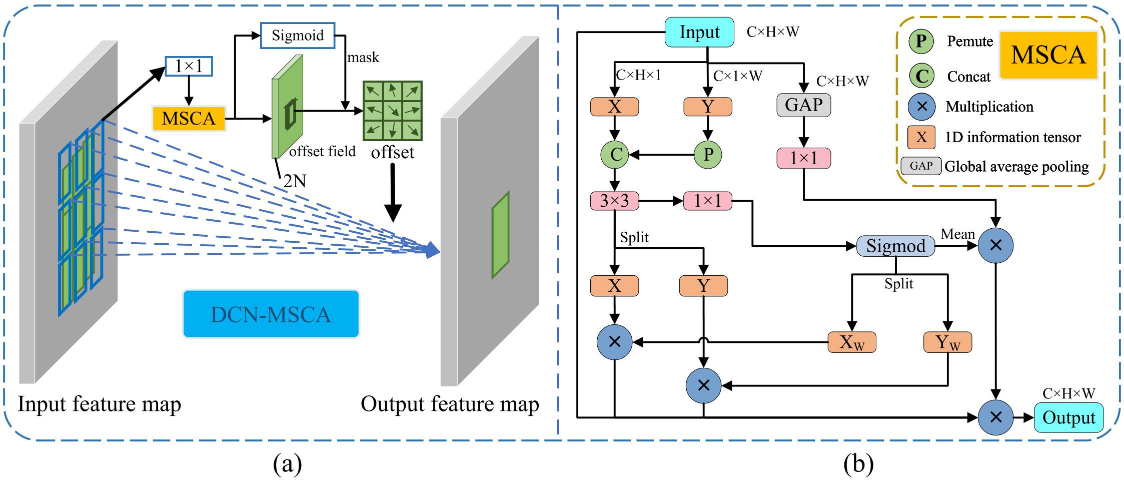


Fig. 5. Illustration of DCN-MSCA convolution. (a) Schematic of the overall flow of DCN-MSCA convolution; (b) Schematic of the MSCA mechanism.

4.3. Contextual Information Fusion Downsampling module (CFD)

The CFD module is essential for downsampling and integrating information. It merges local features with contextual semantic information and enhances them using global context, improving information utilization. Positioned before the DCM module in the backbone and the C2f module in the neck, the CFD spans the entire network from spatial to semantic levels, facilitating rapid and accurate feature transfer and preventing information loss.

The CFD module first performs a 3 × 3 convolution with a stride of 2 for downsampling, followed by a regular convolution to collect local features and a dilated convolution to capture contextual features. It then passes through a joint feature extractor, including a concatenation layer, batch normalization, and SiLU activation to fuse local and contextual features. Finally, a global feature extractor, consisting of a global pooling layer and two fully connected layers, extracts features and generates a weight vector to guide joint feature fusion and produce the final output (See Fig. 6).

CGNet [44] is a lightweight model for semantic segmentation. Inspired by the Context-Guided Block (CG block) in CGNet that combines the surrounding contextual information with local features, the CFD module borrows its design idea to combine it with the downsampling module and use it for Object detection modeling. To maximize the use of contextual information, the CFD module incorporates a downsampling operation before the main convolution module, facilitating comprehensive integration of contextual information. This fusion enhances feature extraction and improves overall information extraction without significantly increasing computational load. Additionally, the CFD module is integrated throughout the entire model architecture, from the spatial to the semantic level, spanning the backbone network to the PAN structure. As the feature map undergoes downsampling, the CFD module compensates for information loss caused by the transformation process.

5. Experiment evaluation

5.1. Experiment setups

Datasets: The following three widely used datasets were used for the performance evaluation:

- **Microsoft COCO 2017 dataset** [50]: The dataset stands as one of the most extensive public datasets for object detection, featuring 80 categories, ranging from pedestrians and transportation to animals, household goods, and public facilities. The training set comprises 118,287 images, with an additional test set of 5,000 images.
- **PASCAL VOC dataset (07+12):** The dataset combines the training and validation sets from PASCAL VOC 2007 and PASCAL VOC 2012, utilizing the test set from PASCAL VOC 2007. The training set encompasses 16,551 images, and the test set includes 4,952 images.
- **RUOD dataset** [51]: Addressing various challenges in underwater detection, the RUOD dataset includes 9,800 images in the training set and 4,200 images in the test set. The dataset tagged target categories including: fish, divers, starfish, coral, sea turtles, sea urchins, sea cucumbers, scallops, squids, and jellyfish in 10 categories. Furthermore, RUOD incorporates three additional test sets specifically designed for environmental challenges: fog effect, color bias, and light interference. This setup enables a comprehensive evaluation of detector performance.

Implementation details: The experiments were conducted using YOLOv8 as the baseline model. The batch size for training YOLO-DC was set to 64, and the stochastic gradient descent (SGD) optimizer with a momentum factor of 0.937 was employed. The weight decay of the optimizer is 5e−4. The initial learning rate was set to 1e−2 and gradually reduced to a final learning rate of 1e−4. A total of 500 epochs were conducted for training YOLO-DC, with mosaic augmentation disabled during the last 10 epochs to improve accuracy. All other hyperparameters were kept consistent with the baseline model. The model utilized binary cross entropy (BCE) loss for classification, along with distribution focal loss (DFL) and complete IoU (CIoU) loss for rectangular box regression.

To ensure the accuracy of the experimental data, no pre-trained weights were used in any of the experiments, allowing the model to be trained from scratch. YOLO-DC used single-scale images (640 × 640) as input, and the evaluation metrics followed the COCO criteria. The reported results include the normalized average precision (AP) at various intersection over union (IoU) thresholds. Additionally, the average precision for different target sizes was reported, denoted as AP-Small, AP-Medium, and AP-Large, representing small, medium, and large targets under the COCO evaluation metrics.

All model experiments were conducted using 2 NVIDIA RTX 3090 GPUs. The deep learning framework used was PyTorch 2.0.

5.2. Comparisons with state-of-the-art methods

Comparisons on the Microsoft COCO 2017 dataset: In order to fully evaluate the performance of the models, numerous SOTA models such as EfficientDet [52], RetinaNet [53], DETR-DC [1], Faster R-CNN [54], YOLOv5 [11], YOLOv6 [12], YOLOv7 [7], YOLOX [22], YOLO-MS [18], Gold-YOLO [21], YOLOv9 [19], YOLOv10 [23] and the

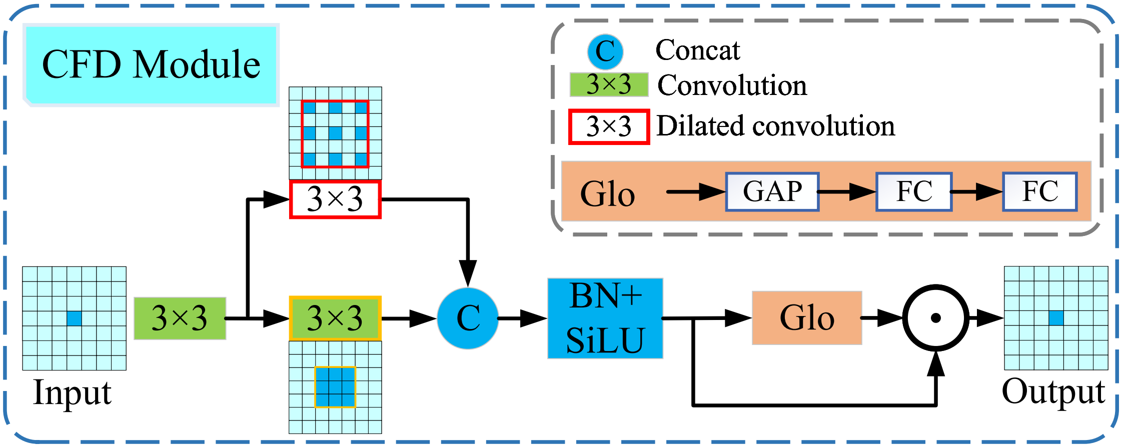


Fig. 6. Illustration of CFD module.

Table 1
Comparison of YOLO-DC with other SOTA models on the COCO 2017 dataset. Several SOTA models were chosen for comparison with different versions of YOLO-DC under the same experimental conditions, where FPS and inference time represent performance metrics for a batch size of 32. Model names in the table with the suffix “50”, “101”, “R50”, or “R101” indicate that the model uses ResNet-50 or ResNet-101 as the backbone network.

Method	Size	AP^{val}	AP^{val}_{50}	AP_s	AP_m	AP_l	FPS	Latency	Params	FLOPs
EfficientDet-D0	512	34.3	54.2	12.0	37.3	50.2	95	10.5 ms	3.9 M	2.5 G
EfficientDet-D1	640	38.9	57.6	16.9	43.3	55.0	69	14.5 ms	6.6 M	6.1 G
EfficientDet-D2	768	42.4	61.3	20.5	46.0	57.4	51	19.6 ms	8.1 M	11 G
EfficientDet-D3	896	44.9	64.0	25.6	48.4	58.8	32	31.5 ms	12 M	25 G
RetinaNet-50	800	35.7	55.0	18.9	38.9	46.3	14	72.6 ms	29 M	165 G
RetinaNet-101	800	37.8	57.3	20.2	41.1	49.2	10	105.5 ms	38 M	205 G
Faster R-CNN	800	27.2	48.4	6.6	28.6	45.0	8	120 ms	42 M	180 G
Faster R-CNN +++	800	34.9	55.7	15.6	38.7	50.9	2	460 ms	60 M	246 G
DETR-DC5-R50	800	43.3	63.1	22.5	47.3	61.1	12	82.4 ms	41 M	187 G
DETR-DC5-R101	800	44.9	64.7	23.7	49.5	62.3	10	96.7 ms	60 M	253 G
YOLOv5-N	640	28.0	45.7	14.0	31.8	36.6	735	1.4 ms	1.9 M	4.5 G
YOLOv5-S	640	37.4	56.8	21.7	42.5	48.8	444	2.3 ms	7.2 M	16.5 G
YOLOv5-M	640	45.4	64.1	28.4	50.8	57.7	209	4.8 ms	21.2 M	49.0 G
YOLOv6-N	640	37.0	52.4	16.8	40.2	52.6	1187	0.8 ms	4.7 M	11.4 G
YOLOv6-S	640	44.3	61.2	23.6	48.7	59.8	484	2.0 ms	18.5 M	45.3 G
YOLOv6-M	640	49.1	66.1	30.0	54.1	64.5	226	4.4 ms	34.9 M	85.8 G
YOLOv7-Tiny	416	33.3	49.9	–	–	–	1196	0.8 ms	6.2 M	5.8 G
YOLOv7-Tiny	640	37.4	55.2	19.9	41.1	50.8	519	1.9 ms	6.2 M	13.7 G
YOLOvX-N	416	32.8	50.3	14.0	35.5	48.3	1143	0.9 ms	5.1 M	6.5 G
YOLOvX-S	640	40.5	59.3	23.9	45.2	53.8	396	2.5 ms	9.0 M	26.8 G
YOLOvX-M	640	46.9	65.6	29.0	51.2	60.9	179	5.6 ms	25.3 M	73.8 G
Gold-YOLO-N	640	39.9	55.9	19.7	44.1	57.0	1030	1.0 ms	5.6 M	12.1 G
Gold-YOLO-S	640	46.1	63.3	25.3	50.2	62.6	446	2.2 ms	21.5 M	46.0 G
Gold-YOLO-M	640	50.9	68.2	32.3	55.3	66.3	220	4.5 ms	41.3 M	87.5 G
YOLO-MS-XS	640	43.4	60.4	23.7	48.3	60.3	131	7.6 ms	4.5 M	8.7 G
YOLO-MS-S	640	46.2	63.7	26.9	50.5	63.0	110	9.0 ms	8.1 M	15.6 G
YOLO-MS	640	51.0	68.6	33.1	56.1	66.5	80	12.3 ms	22.2 M	40.1 G
YOLOv8-N	640	37.3	52.6	15.3	35.6	54.7	734	1.4 ms	3.2 M	8.7 G
YOLOv8-S	640	44.9	60.8	23.6	47.1	65.7	387	2.6 ms	11.2 M	28.6 G
YOLOv8-M	640	50.2	67.2	28.9	53.6	69.6	176	5.7 ms	28.9 M	78.9 G
YOLOv9-S	640	46.8	63.4	26.6	56.0	64.5	–	–	7.7 M	26.4 G
YOLOv9-M	640	51.4	68.1	33.6	57.0	68.0	–	–	20.0 M	76.3 G
YOLOv10-N	640	38.5	–	–	–	–	542	1.8 ms	2.3 M	6.7 G
YOLOv10-S	640	46.3	–	–	–	–	401	2.5 ms	7.2 M	21.6 G
YOLOv10-M	640	51.1	–	–	–	–	210	4.7 ms	15.4 M	59.1 G
YOLO-DC-N	640	40.8	56.9	16.6	39.6	60.0	676	1.5 ms	3.9 M	8.9 G
YOLO-DC-S	640	46.6	63.5	24.6	48.5	65.8	334	3.0 ms	13.9 M	29.2 G
YOLO-DC-M	640	50.4	67.3	27.9	53.9	69.7	147	6.8 ms	32.9 M	70.9 G

Table 2
Comparative evaluation of YOLO-DC with other SOTA models on the PASCAL VOC (07+12) and RUOD datasets.

Method	Params	FLOPs	PASCAL VOC(07+12)		RUOD	
			$AP^{val}(\%)$	$AP_{50}^{val}(\%)$	$AP^{val}(\%)$	$AP_{50}^{val}(\%)$
YOLOv5-N	1.9 M	4.5 G	45.1	72.5	53.6	72.1
YOLOv5-S	7.2 M	16.5 G	53.0	76.2	58.8	79.4
YOLOv6-N	4.7 M	11.4 G	60.5	82.0	59.7	84.2
YOLOv7-Tiny	6.2 M	13.7 G	53.8	79.3	57.2	85.3
YOLOv8-N	3.2 M	8.7 G	59.1	79.9	61.9	85.3
YOLO-DC-N	3.9 M	8.9 G	**62.6**	**82.4**	**62.8**	**85.6**

Table 3
Ablation study on different modules using YOLOv8-N as the baseline model: evaluating performance on the microsoft COCO 2017 dataset.

Method	AP^{val}	AP_{50}^{val}	AP_s	AP_m	AP_l	FPS	Latency	Params	FLOPs
YOLOv8-N(Baseline)	37.30	52.61	15.28	35.59	54.66	734	1.4 ms	3.2 M	8.7 G
+DCNv2	38.63	54.44	15.62	37.63	55.91	756	1.3 ms	3.2 M	7.4 G
+DCN-MSCA(DCM)	39.03	54.97	15.72	37.93	56.32	724	1.4 ms	3.2 M	7.9 G
+CFD	38.44	53.83	15.24	36.81	55.68	789	1.3 ms	3.8 M	9.8 G
YOLO-DC(DCM+CFD)	**40.80**	**56.89**	**16.62**	**39.61**	**60.04**	676	1.5 ms	3.9 M	8.9 G

baseline model YOLOv8 [13] were selected for the experiments. They were compared with YOLO-DC on the Microsoft COCO 2017 dataset under the same experimental conditions.

The experimental results presented in Table 1 indicate that each version of YOLO-DC surpasses YOLOv8, with significant improvements in overall performance. Enhanced information utilization by the CFD module and robust feature extraction through the DCM module contribute to these advancements. For instance, YOLO-DC-N achieves an AP of 40.8%, a 3.5% improvement over YOLOv8-N, while maintaining similar computational and parameter requirements with minimal increase in inference time. YOLO-DC also outperforms YOLOv8-S and YOLOv8-M, with AP gains of 1.7% and 0.2%, respectively.

Compared with other SOTA models, YOLO-DC shows significant improvements in performance. YOLO-DC-N achieves a 12.8% AP increase over YOLOv5-N, 3.8% over YOLOv6-N, 3.4% over YOLOv7-Tiny, 8% over YOLOX-N, and 0.9% over Gold-YOLO-N. Additionally, the S and M versions of YOLO-DC show notable AP improvements compared to the corresponding S and M versions. Across all versions, YOLO-DC consistently outperforms EfficientDet, RetinaNet, and Faster R-CNN, with significantly lower computational overhead and faster inference times.

Although YOLO-DC models show a slight increase in parameters compared to YOLOv8, they achieve an AP boost of approximately 1% to 3%, which is considered a favorable trade-off. Despite Gold-YOLO employing self-distillation during training, both the N and S versions of YOLO-DC surpass it, with only a marginally lower AP for YOLO-DC-M. Importantly, YOLO-DC models require fewer computational resources and parameters compared to Gold-YOLO. Among the models compared, only YOLO-MS and DETR-DC5 achieve slightly higher APs than YOLO-DC; however, YOLO-DC is faster and has lower computational overhead. For example, YOLO-DC-N requires only 20% of the inference time of YOLO-MS-XS, and YOLO-DC models operate with less than 10% of the computational and parametric overhead of DETR-DC5. However, it is worth noting that the AP of YOLO-DC-M is 5.6% higher than that of DETR-DC5-R101, reaching 50.9%, while YOLO-DC requires fewer computations and parameters, and has a faster inference time. Comparing with the newly released YOLOv9 and YOLOv10, the N and S versions of the YOLO-DC are still superior to the YOLOv10, while the YOLOv9 does not have a corresponding N version and the YOLO-DC-S is only slightly less powerful than the YOLOv9.

Comparisons on PASCAL VOC and RUOD datasets: For a comprehensive evaluation of model performances, we conducted comparative experiments on the PASCAL VOC dataset (07+12) and the underwater dataset RUOD. The selected models included YOLOv5 (N, S) [11], YOLOv6-N [12], YOLOv7-Tiny [7], and the baseline model YOLOv8-N [13].

Table 2 presents the results of the comparison tests between YOLO-DC and other SOTA models on the PASCAL VOC (07+12) dataset and the RUOD dataset. The results show that on the PASCAL VOC (07+12) dataset, the AP of YOLO-DC-N is significantly higher compared to YOLOv8-N, reaching 62.6%, which is an improvement of 3.5%. This outperforms all other selected SOTA models. Notably, YOLOv6-N, one of the top performers, lags behind YOLO-DC-N by 0.39% in AP, while YOLO-DC-N maintains lower computational and parametric quantities than YOLOv6-N. Benefiting from the exceptional performance of deformable convolution in handling geometric deformations, YOLO-DC consistently outperforms other SOTA models on the RUOD dataset. Compared to YOLOv8-N, YOLO-DC-N achieves a 1.1% increase in AP, reaching 62.8%. Despite having only half the parameters of YOLOv5-S, YOLO-DC-N achieves a 4% higher AP.

Experiments demonstrate that the YOLO-DC model outperforms several state-of-the-art algorithms across multiple datasets in terms of parameters, computation, inference time, and detection accuracy.

5.3. Ablation study

To systematically validate the efficacy of the proposed improvements, this study employs YOLOv8-N as the baseline model and conducts ablation analysis on the COCO 2017 dataset through a phased experimental approach: First, combinatorial efficacy validation is performed on DCNv2, DCM, CFD module, and the MSCA (results presented in Table 3). Subsequently, a cross-layer deployment contribution analysis is conducted specifically for the DCM and CFD modules to reveal their differentiated optimization mechanisms across the backbone network, neck network, and detection head (results detailed in Table 4).

5.3.1. Combinatorial efficacy validation

This section investigates the combined efficacy of DCNv2, DCM, CFD, and MSCA through a phased ablation study. The experimental results are systematically documented in Table 3. In this study, we used YOLOv8-N as the baseline model, which initially achieved an AP of 37.3%. Our first improvement involved replacing the normal convolution in the C2f module of the baseline model with DCNv2, resulting in a 1.33% improvement in AP, reaching 38.63%. Additionally, under the COCO evaluation metrics, there were improvements in AP for targets of different scales, with increases of 0.34%, 2.04%, and 1.25% in AP_s, AP_m, and AP_l, respectively. This highlights the superior performance of DCNv2 convolution in expanding the receptive field and enhancing feature extraction capabilities compared to normal convolution.

After further enhancing DCNv2 by incorporating the MSCA mechanism to improve its feature extraction capabilities, there was a 0.4%

Table 4
Ablation study of the DCM and CFD modules between different layers of the model using YOLOv8-N as a baseline model: performance was evaluated on the Microsoft COCO 2017 dataset. Annotated with “*” for our proposed YOLO-DC configuration.

Method	AP^{val}	AP_{50}^{val}	AP_s	AP_m	AP_l	FPS	Latency	Params	FLOPs
YOLOv8-N(Baseline)	37.30	52.61	15.28	35.59	54.66	734	1.4 ms	3.2 M	8.7 G
DCM-Backbone	39.03	54.97	15.72	37.93	56.32	724	1.4 ms	3.2 M	7.9 G
DCM-Neck	37.33	52.67	15.22	35.83	54.62	704	1.4 ms	3.2 M	8.0 G
DCM-Head	37.45	53.23	15.34	35.81	55.02	698	1.4 ms	3.2 M	8.0 G
DCM-All	39.08	54.99	15.73	37.94	56.42	653	1.6 ms	3.3 M	8.1 G
DCM-Backbone-CFD*	40.80	56.89	16.62	**39.61**	60.04	676	1.5 ms	3.9 M	8.9 G
DCM-Neck-CFD	38.57	54.18	17.19	37.27	55.93	685	1.4 ms	3.9 M	8.9 G
DCM-Head-CFD	38.84	54.69	16.58	36.92	55.89	678	1.4 ms	3.8 M	9.0 G
DCM-All-CFD	**40.87**	**56.96**	**16.68**	39.49	**60.29**	617	1.6 ms	4.1 M	9.1 G

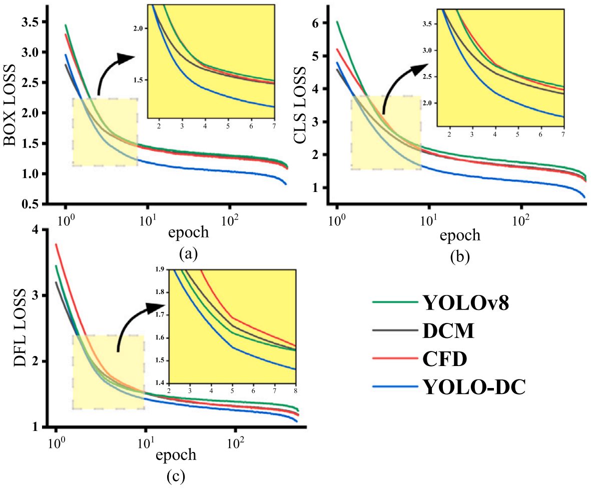


Fig. 7. Model Loss Curves. (a) CIoU Loss; (b) Classification BCE Loss; (c) DFL Loss.

increase in AP compared to DCNv2 alone and a 1.73% increase compared to the baseline model that utilizes normal convolution, resulting in an AP of 39.03%. The APs for small, medium, and large targets (AP_s, AP_m, AP_l) saw improvements of 0.44%, 2.34%, and 1.66%, respectively. This improvement highlights the effectiveness of the MSCA mechanism, demonstrating that integrating MSCA provides a more accurate receptive field for DCN-MSCA convolution compared to previous approaches with DCNv2. The feature extraction capability of the DCM module using DCN-MSCA convolution is significantly enhanced compared to the baseline model, leading to a substantial performance improvement.

Using the CFD module alone increased the AP of the model by 1.14% over the baseline, reaching 38.44%. Additionally, AP_m and AP_l showed increases of 1.22% and 1.02%, respectively. This demonstrates the effectiveness of the module in enhancing information utilization and reducing losses by integrating surrounding and global feature information into the main convolution module after downsampling.

Employing both the DCM and CFD modules in the YOLO-DC model leads to a 3.5% improvement in AP, reaching 40.8% compared to the baseline. The CFD module enhances information utilization by integrating contextual details, while the DCM module improves feature extraction with a more precise receptive field. Placing the CFD module before the DCM module allows for efficient transfer of critical information, resulting in superior feature extraction. This synergy explains why the combined improvement exceeds the sum of the individual modules. Although the parameter count increases by 0.7 M and the computational load by 0.2 G, the inference time remains nearly unchanged, indicating minimal computational overhead for a significant performance gain.

To comprehensively evaluate the effectiveness of the model improvements, we conducted a systematic analysis of loss function characteristics. Although retaining the baseline model’s loss function configuration (Binary Cross-Entropy (BCE) loss for classification tasks and a combination of Distribution Focal Loss (DFL) and Complete IoU (CIoU) loss for bounding box regression), we quantified the impact of the DCM and CFD modules on loss evolution through three controlled experimental groups under strictly controlled experimental conditions.

As demonstrated by the convergence curve comparisons in Fig. 7, the results reveal that regardless of the loss type, the convergence rate improves with the incorporation of DCM and CFD modules. Specifically, YOLO-DC achieves faster convergence speed and lower final loss compared to the baseline configurations.

5.3.2. Cross-layer deployment contribution analysis

Building on the validated synergistic effects of module combinations, we further investigate the performance impacts of deploying the DCM and CFD module across distinct network hierarchies. As shown in Table 4, deploying DCM solely in the backbone network (DCM-Backbone) elevates the AP to 39.03%, marking a 1.73% AP improvement over the baseline (37.30%→ 39.03%). This configuration achieves a medium-scale target detection accuracy (AP_m) of 37.93%, significantly surpassing deployments in the neck (DCM-Neck: 35.83%) and head (DCM-Head: 35.81%). These results underscore the backbone network’s decisive role in multi-scale detection through dynamic receptive field adaptation. While full-layer DCM deployment (DCM-All) marginally improves AP to 39.08%, it incurs a 11% inference speed penalty (734→ 653 FPS) due to increased computational complexity (8.1 GFLOPs vs. baseline 8.7 GFLOPs), highlighting the efficiency trade-offs of global deformable operations.

Integrating the CFD module further demonstrates its unique optimization potential. Co-implementation of CFD with DCM in the backbone (DCM-Backbone-CFD*) increases AP to 40.80% - a 1.77% gain over the standalone DCM-Backbone - with AP_m reaching 39. 61%. This enhancement originates from CFD’s context-aware feature fusion during downsampling, which mitigates local detail loss inherent in deformable convolutions. In contrast, co-deploying CFD with DCM in the neck (DCM-Neck-CFD) or head (DCM-Head-CFD) yields limited improvements (38.57% and 38.84% AP, respectively), corresponding to gains of only 1.25% and 1.39% over their individual deployments. Notably, full-layer co-deployment (DCM-All-CFD) achieves peak AP (40.87%) but at the cost of a 28% parameter increase (4.1M vs. baseline 3.2M) and it incurs a 16% inference speed penalty (734→ 617 FPS). These findings solidify the backbone-centric co-deployment strategy (DCM-Backbone-CFD*) as the optimal balance between accuracy and efficiency for real-world deployment.

The cross-layer ablation study conclusively demonstrates: (1) The backbone network is paramount for leveraging the dynamic feature extraction capabilities inherent to the DCM architecture; (2) CFD systematically enhances information utilization through multi-scale context aggregation. Our optimal synergy strategy (YOLO-DC-N, i.e., DCM-Backbone-CFD*) delivers a 3.5% AP gain (37.30%→40.80%) while maintaining computational overhead at 8.9 GFLOPs, thereby achieving state-of-the-art efficiency-accuracy trade-offs in object detection.

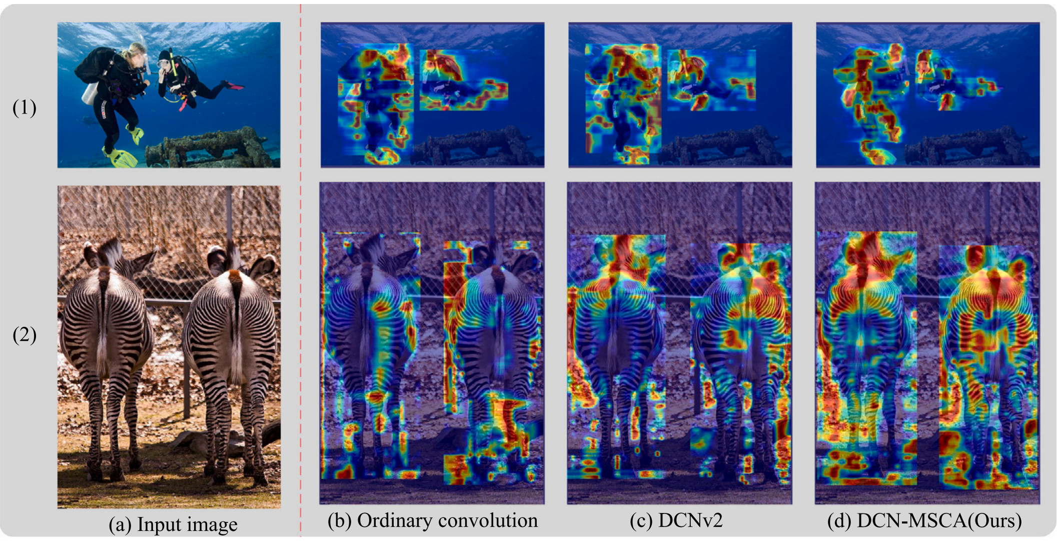


Fig. 8. Comparison plots of feature extraction results with different convolutions using YOLOv8 as the base model. (a) shows the input image, while (b) to (d) display the visualized heatmaps (GradCAM [55]) extracted by the model at the convolution stage of the backbone network after applying normal convolution, DCNv2, and DCN-MSCA (Ours), respectively.

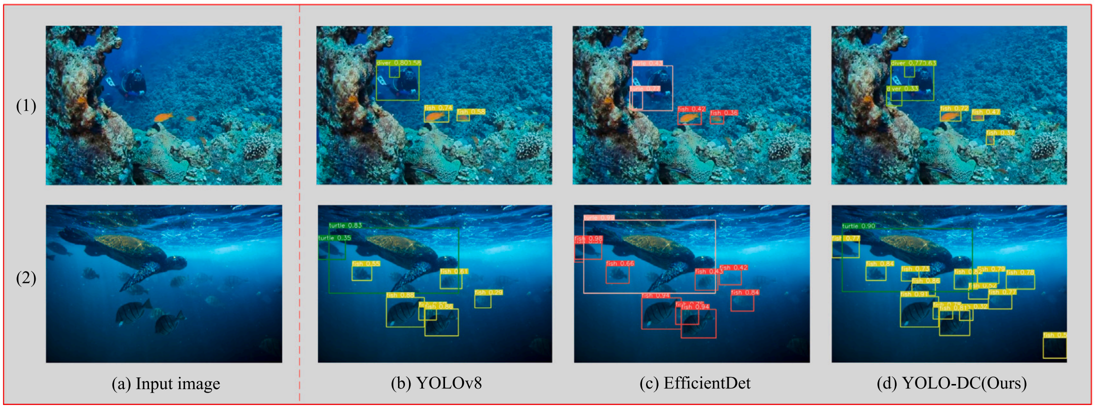


Fig. 9. Comparison of image detection results from the RUOD dataset. (a) Original input image; (b) Detection results of YOLOv8-N; (c) Detection results of EfficientDet-D1; (d) Detection results of YOLO-DC-N. Prediction categories and confidence scores are displayed above the prediction box in the figure.

5.4. Visualizing qualitative research

In this subsection, we compare the feature visualization of the improved module with the target detection results of SOTA methods.

Given the optimization of the model convolutional structure by introducing DCN-MSCA convolution, which more accurately extends the receptive field and enhances feature extraction, we conduct comparative visualization experiments using YOLOv8 as the baseline model in its backbone network. Normal convolution, DCNv2, and our DCN-MSCA convolution are compared. As shown in Fig. 8, GradCAM generates heatmaps for feature extraction. It is clear that features extracted by ordinary convolution are scattered and limited, focusing only on localized areas of the object, such as the lower limbs of the diver and the head and back of the zebra being overlooked. DCNv2 shows broader coverage but still misses many details and includes irrelevant regions. In contrast, DCN-MSCA convolution extracts more comprehensive and focused features. As depicted in the figure, the limbs and head of both the diver and zebra are clearly captured, with fewer irrelevant regions. This improvement results from the MSCA mechanism, which allows for more precise deformation of the receptive field, leading to targeted feature extraction and reduced irrelevant information.

We visualize and compare the detection results of YOLO-DC against YOLOv8 and EfficientDet models. As shown in Fig. 9, YOLOv8 and EfficientDet exhibit limitations, including missed detections and false positives, especially with occluded or blurred objects. In contrast, YOLO-DC, with enhanced feature extraction provided by DCN-MSCA convolution, achieves superior detection of geometrically transformed objects. Furthermore, the integration of the CFD module improves information utilization, delivering the best performance among the evaluated models.

6. Conclusion

This paper introduces YOLO-DC, an object detection model that incorporates a novel Deformable Convolution Module (DCM) and Contextual Information Fusion Downsampling (CFD) module. The DCM enhances feature extraction by integrating deformable convolution with multi-scale spatial channel attention, while the CFD module optimizes the use of local and surrounding context to reduce information loss during downsampling. These innovations position YOLO-DC ahead of other state-of-the-art models. Our experiments demonstrate that YOLO-DC performs exceptionally well in both general object detection scenarios and challenging environments such as underwater detection, highlighting its robustness and adaptability. YOLO-DC was deployed in an ecological fish passage monitoring system at a major water conservancy hub, delivering real-time, precise fish detection in complex underwater environments. This demonstrates its effectiveness as a research tool with practical applications in related fields.

Looking forward, we aim to further develop YOLO-DC for practical deployments by systematically investigating its generalizability across diverse complex scenarios. While our current work demonstrates significant advances in dynamic feature modeling, it is imperative to note that challenges persist in computational efficiency optimization. Specifically, the computational complexity requires innovative solutions such as the chaotic optimization algorithm proposed by Özçelik et al. [16] and the lightweight CNN architecture developed by Yağ et al. [48], both of which show promising potential for model compression. Furthermore, building upon Çiçek et al.'s chaotic functional connectivity matrix [49], we envision novel opportunities for multimodal contextual fusion through nonlinear correlation modeling. These interconnected research directions collectively form a roadmap for enhancing both theoretical foundations and practical applicability in subsequent studies.

CRediT authorship contribution statement

Dengyong Zhang: Writing – review & editing, Visualization, Supervision, Resources, Project administration, Funding acquisition. **Chuanzhen Xu:** Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Jiaxin Chen:** Writing – review & editing, Resources, Project administration, Data curation. **Lei Wang:** Investigation. **Bin Deng:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant (No. 62172059, 62402062), Hunan Provincial Key Research and Development Program, China under Grant (No. 2024AQ2027), Natural Science Foundation of Changsha City, China under Grant (No. kq2402031), Science and Technology Project of Hunan Provincial Department of Water Resources, China under Grant (No. XSKJ2024064-36).

Data availability

Data will be made available on request.

References

[1] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European Conference on Computer Vision, ECCV, 2020, pp. 213–229, http://dx.doi.org/10.1007/978-3-030-58452-8_13.

[2] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. Ni, H.-Y. Shum, DINO: DETR with improved DeNoising anchor boxes for end-to-end object detection, in: The Eleventh International Conference on Learning Representations, ICLR, 2023, http://dx.doi.org/10.48550/arXiv.2203.03605.

[3] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, J. Chen, Detrs beat yolos on real-time object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2024, pp. 16965–16974.

[4] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 779–788, http://dx.doi.org/10.1109/CVPR.2016.91.

[5] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, 2018, http://dx.doi.org/10.48550/arXiv.1804.02767, arXiv preprint arXiv:1804.02767.

[6] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, I.-H. Yeh, CSPNet: A new backbone that can enhance learning capability of CNN, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, 2020, pp. 1571–1580, http://dx.doi.org/10.1109/CVPRW50498.2020.00203.

[7] C.-Y. Wang, A. Bochkovskiy, H.-Y.M. Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2023, pp. 7464–7475, http://dx.doi.org/10.1109/CVPR52729.2023.00721.

[8] C.-Y. Wang, A. Bochkovskiy, H.-Y.M. Liao, Scaled-YOLOv4: Scaling cross stage partial network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 13024–13033, http://dx.doi.org/10.1109/CVPR46437.2021.01283.

[9] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2018, pp. 8759–8768, http://dx.doi.org/10.1109/CVPR.2018.00913.

[10] X. Xu, Y. Jiang, W. Chen, Y. Huang, Y. Zhang, X. Sun, Damo-yolo: A report on real-time object detection design, 2022, arXiv preprint arXiv:2211.15444.

[11] G. Jocher, YOLOv5 by ultralytics, 2020, http://dx.doi.org/10.5281/zenodo.3908559, https://github.com/ultralytics/yolov5.

[12] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, et al., YOLOv6: A single-stage object detection framework for industrial applications, 2022, http://dx.doi.org/10.48550/arXiv.2209.02976, arXiv preprint arXiv:2209.02976.

[13] G. Jocher, A. Chaurasia, J. Qiu, Ultralytics YOLOv8, 2023, https://github.com/ultralytics/ultralytics.

[14] Y.-L. Chen, C.-L. Lin, Y.-C. Lin, T.-C. Chen, Transformer-CNN for small image object detection, Signal Process.: Image Commun. (SPIC) 129 (2024) 117194, http://dx.doi.org/10.1016/j.image.2024.117194.

[15] J. Wu, J. Chen, Q. Lu, J. Li, N. Qin, K. Chen, X. Liu, U-ATSS: A lightweight and accurate one-stage underwater object detection network, Signal Process.: Image Commun. (SPIC) 126 (2024) 117137, http://dx.doi.org/10.1016/j.image.2024.117137.

[16] Y.B. Özçelik, A. Altan, Overcoming nonlinear dynamics in diabetic retinopathy classification: A robust AI-based model with chaotic swarm intelligence optimization and recurrent long short-term memory, Fractal Fract. 7 (8) (2023) http://dx.doi.org/10.3390/fractalfract7080598, URL: https://www.mdpi.com/2504-3110/7/8/598.

[17] O. Çiçek, Y.B. Özçelik, A. Altan, A new approach based on metaheuristic optimization using chaotic functional connectivity matrices and fractal dimension analysis for AI-driven detection of orthodontic growth and development stage, Fractal Fract. (2025) http://dx.doi.org/10.3390/fractalfract9030148, URL: https://api.semanticscholar.org/CorpusID:276665513.

[18] Y. Chen, X. Yuan, R. Wu, J. Wang, Q. Hou, M.-M. Cheng, YOLO-MS: rethinking multi-scale representation learning for real-time object detection, IEEE Trans. Cogn. Dev. Syst. (TCDS) (2023) http://dx.doi.org/10.1109/TCDS.2023.3238181.

[19] C.-Y. Wang, I.-H. Yeh, H.-Y. Mark Liao, YOLOv9: Learning what you want to learn using programmable gradient information, in: Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XXXI, Springer-Verlag, Berlin, Heidelberg, 2024, pp. 1–21, http://dx.doi.org/10.1007/978-3-031-72751-1_1.

[20] J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 6517–6525, http://dx.doi.org/10.1109/CVPR.2017.690.

[21] C. Wang, W. He, Y. Nie, J. Guo, C. Liu, Y. Wang, K. Han, Gold-YOLO: Efficient object detector via gather-and-distribute mechanism, NIPS, in: Neural Information Processing Systems, vol. 36, 2023, pp. 51094–51112, http://dx.doi.org/10.48550/arXiv.2309.11331.

[22] Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, Yolox: Exceeding yolo series in 2021, 2021, http://dx.doi.org/10.48550/arXiv.2107.08430, arXiv preprint arXiv:2107.08430.

[23] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, G. Ding, YOLOv10: Real-time end-to-end object detection, 2024, arXiv:2405.14458, URL: https://arxiv.org/abs/2405.14458.

[24] C. Feng, Y. Zhong, Y. Gao, M.R. Scott, W. Huang, Tood: Task-aligned one-stage object detection, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 3490–3499, http://dx.doi.org/10.1109/ICCV48922.2021.00349.

[25] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, ICCV, 2017, pp. 764–773, http://dx.doi.org/10.1109/ICCV.2017.89.

[26] X. Zhu, H. Hu, S. Lin, J. Dai, Deformable convnets v2: More deformable, better results, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 9300–9308, http://dx.doi.org/10.1109/CVPR.2019.00953.

[27] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, et al., Internimage: Exploring large-scale vision foundation models with deformable convolutions, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2023, pp. 14408–14419, http://dx.doi.org/10.1109/CVPR52729.2023.01385.

[28] Y. Xiong, Z. Li, Y. Chen, F. Wang, X. Zhu, J. Luo, W. Wang, T. Lu, H. Li, Y. Qiao, L. Lu, J. Zhou, J. Dai, Efficient deformable ConvNets: Rethinking dynamic and sparse operator for vision applications, 2024, arXiv preprint arXiv:2401.06197.

[29] H. Fengqi, C. Ming, F. Guofu, Improved YOLO object detection algorithm based on deformable convolution, Comput. Eng. 47 (10) (2021) 269–275,282.

[30] S. Dong, W. Xu, H. Zhang, L. Gong, Cot-DCN-YOLO: Self-attention-enhancing YOLOv8s for detecting garbage bins in urban street view images, Egypt. J. Remote. Sens. Space Sci. 28 (1) (2025) 89–98, http://dx.doi.org/10.1016/j.ejrs.2025.01.002.

[31] G.Y. YUAN Zhixiang, InternDiffuseDet:Object detection method combining deformable convolution and diffusion model, Comput. Eng. Appl. 60 (12) (2024) 203–215, http://dx.doi.org/10.6041/j.issn.1000-1298.2024.11.015.

[32] L.C. YUAN Hongchun, Research on FDC-YOLO v8 underwater biological object detection method improved by deformable convolution, Trans. Chin. Soc. Agric. Mach. 55 (11) (2024) 140, http://dx.doi.org/10.6041/j.issn.1000-1298.2024.11.015.

[33] X. Wang, Z. Zhu, Context understanding in computer vision: A survey, Comput. Vis. Image Underst. (CVIU) 229 (2023) 103646, http://dx.doi.org/10.1016/j.cviu.2023.103646.

[34] X. Li, W. Wang, X. Hu, J. Yang, Selective kernel networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 510–519, http://dx.doi.org/10.1109/CVPR.2019.00060.

[35] Y. Cao, J. Xu, S. Lin, F. Wei, H. Hu, GCNet: Non-local networks meet squeeze-excitation networks and beyond, Int. Conf. Comput. Vis. Work. (ICCVW) (2019) 1971–1980, http://dx.doi.org/10.1109/ICCVW.2019.00246.

[36] J.-J. Liu, Q. Hou, M.-M. Cheng, C. Wang, J. Feng, Improving convolutional networks with self-calibrated convolutions, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 10096–10105, http://dx.doi.org/10.1109/CVPR42600.2020.01011.

[37] Z. Gao, J. Xie, Q. Wang, P. Li, Global second-order pooling convolutional networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 3024–3033, http://dx.doi.org/10.1109/CVPR.2019.00314.

[38] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, W. Liu, Ccnet: Criss-cross attention for semantic segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, 2019, pp. 603–612, http://dx.doi.org/10.1109/ICCV.2019.00069.

[39] Q. Hou, D. Zhou, J. Feng, Coordinate attention for efficient mobile network design, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 13713–13722, http://dx.doi.org/10.1109/CVPR46437.2021.01350.

[40] D. Ouyang, S. He, G. Zhang, M. Luo, H. Guo, J. Zhan, Z. Huang, Efficient multi-scale attention module with cross-spatial learning, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2023, pp. 1–5, http://dx.doi.org/10.1109/ICASSP49357.2023.10096516.

[41] K.W. Lau, L.-M. Po, Y.A.U. Rehman, Large separable kernel attention: Rethinking the large kernel attention design in cnn, Expert. Syst. Appl. (ESWA) 236 (2024) 121352, http://dx.doi.org/10.1016/j.eswa.2023.121352.

[42] G. Ioannides, A. Chadha, A. Elkins, Gaussian adaptive attention is all you need: Robust contextual representations across multiple modalities, 2024, arXiv preprint arXiv:2401.11143.

[43] L.-C. Chen, M. Collins, Y. Zhu, G. Papandreou, B. Zoph, F. Schroff, H. Adam, J. Shlens, Searching for efficient multi-scale architectures for dense image prediction, Adv. Neural Inf. Process. Syst. (NIPS) 31 (2018) http://dx.doi.org/10.48550/arXiv.1809.04184.

[44] T. Wu, S. Tang, R. Zhang, J. Cao, Y. Zhang, Cgnet: A light-weight context guided network for semantic segmentation, IEEE Trans. Image Process. (TIP) 30 (2020) 1169–1179, http://dx.doi.org/10.1109/TIP.2020.3042065.

[45] J.-S. Lim, M. Astrid, H.-J. Yoon, S.-I. Lee, Small object detection using context and attention, in: International Conference on Artificial Intelligence in Information and Communication, ICAIIC, 2021, pp. 181–186, http://dx.doi.org/10.1109/ICAIIC51459.2021.9415217.

[46] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A ConvNet for the 2020s, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 11966–11976, http://dx.doi.org/10.1109/CVPR52688.2022.01167.

[47] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I.S. Kweon, S. Xie, Convnext v2: Co-designing and scaling convnets with masked autoencoders, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2023, pp. 16133–16142, http://dx.doi.org/10.1109/CVPR52729.2023.01548.

[48] İ. Yağ, A. Altan, Artificial intelligence-based robust hybrid algorithm design and implementation for real-time detection of plant diseases in agricultural environments, Biology 11 (12) (2022) http://dx.doi.org/10.3390/biology11121732.

[49] A. Sezer, A. Altan, Detection of solder paste defects with an optimization-based deep learning model using image processing techniques, Solder. Surf. Mount Technol. 33 (5) (2021) 291–298, http://dx.doi.org/10.1108/SSMT-04-2021-0013.

[50] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, ECCV, 2014, pp. 740–755, http://dx.doi.org/10.1007/978-3-319-10602-1_48.

[51] C. Fu, R. Liu, X. Fan, P. Chen, H. Fu, W. Yuan, M. Zhu, Z. Luo, Rethinking general underwater object detection: Datasets, challenges, and solutions, Neurocomputing 517 (2023) 243–256, http://dx.doi.org/10.1016/j.neucom.2022.10.039.

[52] M. Tan, R. Pang, Q.V. Le, Efficientdet: Scalable and efficient object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 10778–10787, http://dx.doi.org/10.1109/CVPR42600.2020.01079.

[53] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, ICCV, 2017, pp. 2980–2988.

[54] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) 39 (2016) 1137–1149, http://dx.doi.org/10.1109/TPAMI.2016.2577031.

[55] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, ICCV, 2017, pp. 618–626, http://dx.doi.org/10.1109/ICCV.2017.74.