YOLO-DC: YOLO-Object Detectors Based on Deformable Convolutions

Paper ID: 4985
Track: IJCAI2024 Main Track

Abstract

Object detection, a crucial aspect of computer vision research, has broad applications in fields like medicine, transportation, and security. However, the majority of existing methods focus on optimizing aspects such as model structure, loss function, and data preprocessing, while overlooking the potential enhancements within the convolution. Consequently, there remains untapped potential for improvement in this domain, presenting an opportunity to further enhance model accuracy. This paper introduces YOLO Object Detectors Based on Deformable Convolutions (YOLO-DC), an enhanced algorithm built upon YOLOv8. It features a Deformable Convolution Module (DCM) and a Contextual Information Fusion Downsampling Module (CFD) to boost performance. The DCM module uses deformable convolution with multi-scale spatial channel attention, expanding the receptive field and improving information extraction. The CFD module integrates contextual and local features post-downsampling, introducing global features to enhance joint learning and mitigate information loss. Compared to YOLOv8-N, YOLO-DC-N achieves a 3.5% increase in AP, reaching 40.8% on the Microsoft COCO2017 dataset. YOLO-DC outperforms other state-of-the-art detection algorithms across diverse datasets, including the underwater dataset RUOD and the PACAL VOC dataset (PACAL VOC2007+PACAL VOC2012). The model code is available at https://github.com/Object-Detection-01/YOLO-DC.git.

1 Introduction

Object detection technology, essential in fields like medicine, security, and autonomous driving, relies on Convolutional Neural Network (CNN) and Transformer-based models. The trade-off between accuracy and speed depends on the application. For real-time processing, as in autonomous driving, speed is critical, while higher accuracy is vital for areas like medical image analysis. Achieving the right balance is essential in different contexts.

Early two-stage CNN-based models (e.g., RCNN [Girshick *et al.*, 2014], Fast RCNN [Girshick, 2015], Faster RCNN [Ren *et al.*, 2015]) exhibit slow performance. In contrast, the one-stage model, You Only Look Once (YOLOv1) [Redmon *et al.*, 2016], directly detects object categories and location information through neural networks, providing faster processing with limited accuracy. Subsequent versions of YOLO (YOLOv2 [Redmon and Farhadi, 2017], YOLOv3 [Redmon and Farhadi, 2018]) markedly enhance accuracy while maintaining increased speed. From YOLOv4 to YOLOv7, the algorithm accuracy was enhanced through continuous innovation, preprocessing optimization, network model refinement, and improved loss computation [Wang *et al.*, 2021; Jocher, 2020; Li *et al.*, 2022; Wang *et al.*, 2023b]. The YOLOX algorithm achieved improved accuracy through a novel label assignment strategy [Ge *et al.*, 2021]. YOLOv8 integrated advancements from various domains and refined the network model, resulting in enhanced accuracy [Jocher *et al.*, 2023]. Recently, the Gold-YOLO algorithm further improved accuracy through an aggregation-distribution mechanism [Wang *et al.*, 2023a]. In addition, BGNet has demonstrated promising results in the domain of Camouflage Object Detection (COD) by leveraging edge semantic information associated with the object [Sun *et al.*, 2022]. Subsequently, PENet has further improved performance by incorporating enhanced texture information [Li *et al.*, 2023].

With the Transformer's encoder-decoder structure, object detection models like DETR [Carion *et al.*, 2020] and DINO [Zhang *et al.*, 2022] can capture long-term dependencies between objects, achieving accuracy that surpasses some CNN models. However, their speed still needs improvement compared to CNN models.

In recent years, numerous methods have emerged to enhance model performance. However, some methods may introduce significant computational overhead, impacting speed while enhancing accuracy, or vice versa.

To tackle these challenges, this paper proposes YOLO-Object Detectors Based on Deformable Convolutions (YOLO-DC), a more precise object detection algorithm built upon YOLOv8. The model employs a newly designed Deformable Convolution Module (DCM), enhancing feature extraction, along with a Contextual Information Fusion Downsampling (CFD) module that effectively utilizes local and global contextual information. The strategic integration of

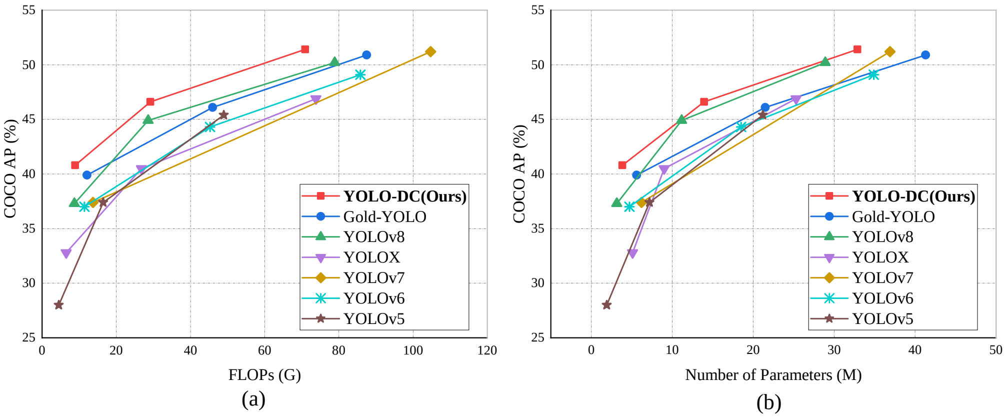


Figure 1: Comparison with other state-of-the-art object detection models on the COCO dataset: (a) AP performance vs. computation (FLOPs), and (b) AP performance vs. Parameters. FLOPs are computed with an input size of 640 × 640. The results clearly demonstrate that YOLO-DC, proposed in this study, achieves an optimal balance between performance and computation.

these two modules leads to a further increase in the model's accuracy. Compared to other state-of-the-art (SOTA) models, as depicted in Fig. 1, YOLO-DC exhibits superior performance, achieving an optimal balance between computational load and parameter count. The main contributions of this paper include:

1. We present the Deformable Convolution Module (DCM) as the key convolution module responsible for feature extraction in the model. DCM employs an enhanced deformable convolution compared to traditional methods using standard convolution. This improvement includes integrating multi-scale spatial channel attention, enhancing the generation of deformable convolutional offsets. As a result, it expands the convolution's receptive field and strengthens the feature extraction capability.
2. We introduce the Contextual Information Fusion Downsampling (CFD) module, designed to integrate contextual information and reduce redundancy after downsampling, enhancing information utilization. CFD incorporates a method for contextual information fusion, enabling it to learn joint features from both local and surrounding contexts.
3. We use DCM and CFD to improve the baseline model. In the model architecture, CFD precedes DCM, integrating information post-downsampling before feeding it into the primary convolutional module. This enhances the model's feature extraction, improving accuracy. YOLO-DC outperforms existing object detection algorithms (e.g., YOLOv8, Gold-YOLO) with comparable parameters and computational load.

2 Related Work

2.1 Object detectors

The YOLO models have evolved over the years, showing significant progress in real-time object detection. Building on YOLOv1, subsequent iterations improved the backbone network, data augmentation strategies, and loss functions for enhanced speed and accuracy. YOLOv8, the latest iteration, signifies the culmination of experience, reaching a new level of integration technology and establishing itself as the state-of-the-art (SOTA) within the YOLO family.

YOLOv8, a significant update over YOLOv5, incorporates new features and enhancements to build upon previous successes. The backbone, grounded in the Cross Stage Partial Network (CSP) concept [Wang *et al.*, 2020], utilizes Darknet-53. However, unlike YOLOv5, YOLOv8 replaces 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. In the neck of the YOLOv8 model, Feature Pyramid Network with Path Aggregation Network is employed.

Unlike YOLOv5, this model adopts a decoupled header structure that segregates the classification and detection headers. The classification loss is determined by BCE loss, while the recognition loss incorporates DFL loss and CIoU loss. YOLOv8 embraces the Anchor-Free approach in its architecture, deviating from the Anchor-Based approach utilized in the earlier YOLO series. It utilizes a Task-Aligned [Feng *et al.*, 2021] dynamic label assignment strategy instead of the edge-length proportional matching method.

YOLOv8 employs mosaic data augmentation, featuring random rotation and translation for input images. In alignment with YOLOX [Ge *et al.*, 2021], mosaic data augmentation is turned off in the final 10 epochs to enhance training efficiency, as well as improve model localization and classification performance.

Despite being the current SOTA model in the YOLO family, YOLOv8, while incorporating advancements from its predecessors, overlooks the potential improvements in the convolutional aspect and fails to harness contextual information during the feature extraction process. This leaves room for enhancing accuracy. In this paper, YOLOv8 is chosen as the baseline model, with the aim of delving into potential improvements in convolution and contextual information to enhance model accuracy within acceptable computational costs.

2.2 Deformable Conv

In the field of computer vision, addressing the geometric transformations of the same object across different scenes and perspectives presents a considerable challenge. To address this issue, Dai et al. [2017] proposed Deformable Convolution (DCNv1) as a solution.

Despite commendable features, DCNv1 has challenges, introducing redundant regions affecting feature extraction. To address this, Zhu X et al. [2019] introduced enhanced deformable convolution (DCNv2), incorporating the weights of each sampling point from DCNv1 to enhance modeling capability. Wang W et al. [2023c] introduced deformable convolution (DCNv3) for large-scale CNN-based models, extending DCNv2 with weight sharing, a multi-group mechanism, and normalization. However, DCNv3 is more suitable for larger models.

2.3 Attention Mechanism

In recent years, attention mechanisms have become integral for enhancing model performance. The SE channel attention mechanism, introduced by Hu J et al. [2018b], applies attention mechanisms to the channel dimension in computer vision. It dynamically adjusts feature responses between channels through feature recalibration. Extending this idea, the CBAM attention mechanism incorporates spatial information encoding through large-scale kernel convolution [Woo *et al.*, 2018]. Subsequent studies, such as GENet [Hu *et al.*, 2018a] and GALA [Park *et al.*, 2019], have further advanced attention mechanisms by employing different spatial attention techniques or designing sophisticated attention blocks.

From a multi-scale perspective, Li X et al. [2019] proposed SK-Net, introducing multiple parallel convolutional kernel branches with varying receptive fields to learn feature map weights at different scales, enabling the network to select more suitable multi-scale feature representations. Conversely, self-attention focuses on constructing spatial or channel attention, with examples like NLNet [Wang *et al.*, 2018], GCNet [Cao *et al.*, 2019], A2Net [Chen *et al.*, 2018b], SCNet [Liu *et al.*, 2020], GsopNet [Gao *et al.*, 2019], and CCNet [Huang *et al.*, 2019], all utilizing the Non-local mechanism to capture diverse spatial information. The CA attention, later proposed by Hou Q et al. [2021], captures location information and channel relationships by decomposing the 2D global pooling operation into two 1D encoding processes, enhancing the feature representation of the network.

2.4 Context Information

Theoretically, various forms of contextual information have been demonstrated to play a pivotal role in the realms of computer vision and image processing, leading to a substantial enhancement in recognition accuracy.

Ding H et al. [2018] introduced a novel local feature model, CCL, designed for context comparison. This model not only leverages contextual information but also emphasizes local details in comparison to the context. DPC employs architecture search techniques to discover efficient multi-scale architectures [Chen *et al.*, 2018a]. However, most of these works focus on exploring contextual information at the decoder stage, overlooking the surrounding context. The CGNet model, proposed by Wu T et al. [2020], learns joint features incorporating both local features and surrounding environmental context. It further enhances the learning of joint features by introducing global context features.

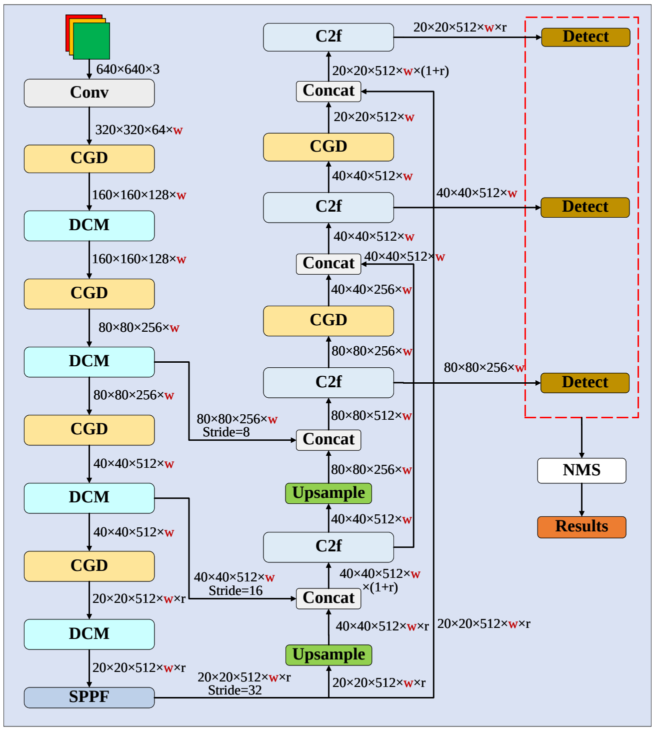


Figure 2: Diagram of the YOLO-DC network structure

3 Method

3.1 YOLO-Object Detectors Based on Deformable Convolutions (YOLO-DC)

In this paper, we present a novel object detector, YOLO-DC, based on deformable convolutions. Figure 2 illustrates the network structure. Building on the success of the YOLO series, especially YOLOv8 as the state-of-the-art (SOTA) model, YOLO-DC is optimized and enhanced from YOLOv8.

The improvement strategy includes replacing the base model's downsampling with the CFD module. In the YOLOv8 backbone, the DCM module replaces the C2f module, leveraging collaborative context information from CFD to enhance features. DCM's powerful feature extraction and superior receptive field markedly boost the model's performance over the standard convolutional module.

YOLO-DC is offered in three versions for diverse scenarios: N (lightweight), S (general), and M (high accuracy). The dimension 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 tailored for lightweight scenarios with minimal equipment demands, the S version is crafted for general scenarios, and the M version, featuring increased computational load and parameters, is well-suited for high-precision scenarios with advanced equipment.

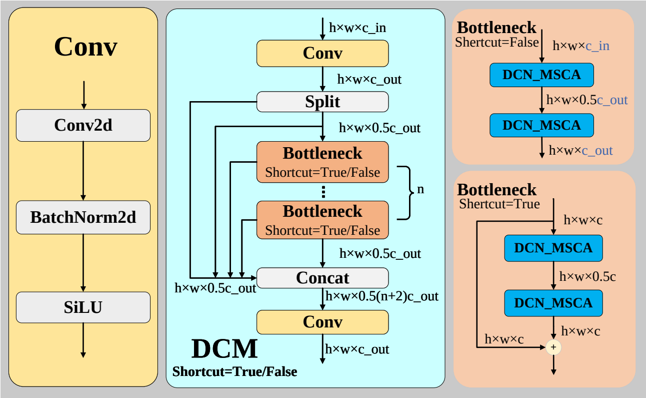


Figure 3: Deformable Convolution Module (DCM)

3.2 Deformable Convolution Module (DCM)

The DCM module incorporates design principles from the C2f module, preserving its rich gradient flow. It integrates the residual concept [He *et al.*, 2016] with the Cross Stage Partial structural design to enhance the convolutional network's learning capacity and reduce parameters. The DCM module conducts a convolutional operation, followed by a Split operation that divides the input into two branches. One branch traverses n Bottleneck modules with residual connections. It then concatenates and fuses with the other branch and the output residuals of the Bottleneck module. The Bottleneck module employs DCNv2 convolution with multi-scale attention, denoted as DCN_MSCA, enhancing feature extraction while minimizing irrelevant regions. Refer to Fig. 3 for the specific schematic of the DCM module.

The fixed geometric structure of standard convolution in neural networks limits its ability to model geometrically transform-rich objects, impacting Object detection performance. Deformable convolution (DCN), especially DCNv2, effectively addresses this limitation by generating offsets for each sampling point through additional learning. This enhances adaptability to geometric variations and includes a weight coefficient mask for offset adjustment [Zhu *et al.*, 2019]. While DCNv2 performs well, its limited learning capacity, derived from obtaining offsets through a single convolutional learning process, restricts accurate determination of its receptive field. This constraint may lead to the inclusion of irrelevant regions, diminishing the offset's discernible impact in different scenarios. To overcome this, we introduce a Multi-scale Channel Spatial Attention (MSCA) mechanism to enhance offset attention in both the X and Y directions.

The improved DCNv2 convolution, named DCN-MSCA, enhances offset precision by integrating MSCA mechanism. This modification broadens the receptive field, reducing the inclusion of irrelevant regions compared to standard DCNv2 convolution. DCN-MSCA exhibits enhanced information extraction capabilities, enabling more accurate contextual region extraction, as depicted in Fig. 4.

The MSCA mechanism extends Coordinate Attention (CA) to tackle information loss by incorporating multiscale fusion. While CA attention decomposes the global pool into a one-dimensional feature encoding of two spatial directions

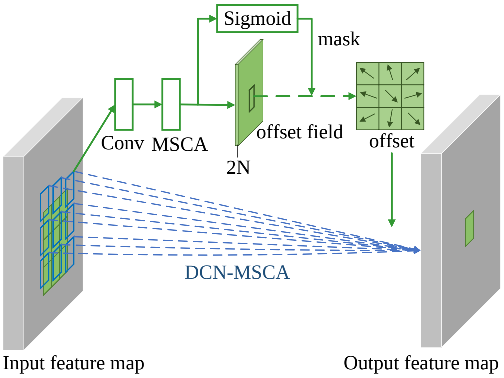


Figure 4: Schematic diagram of DCN-MSCA

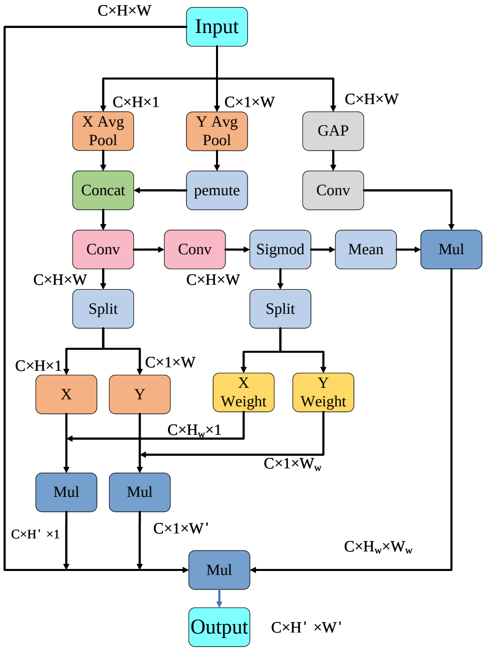


Figure 5: MSCA mechanism schematic

for an orientation-aware feature map, its direct decomposition of orientation information using two global pooling operations is somewhat crude. The subsequent single splicing-then-convolution operation inadequately facilitates information exchange after separation, contributing to suboptimal performance on certain models with CA.

The proposed MSCA mechanism integrates multiscale information based on CA and extracts weights from intermediate processes for bidirectional fusion. This enhances information interaction, with global information from global pooling guiding both directions to mitigate information loss. The MSCA mechanism consists of three branches, each handling multi-scale information at different gradients. The first branch transforms global pooling into one-dimensional codes for X and Y directions. The second branch convolves the first, extracting X_{weight} and Y_{weight} after Sigmoid activation to refine fusion accuracy. The third branch conducts global pooling on initial inputs, and its output guides encoded information from the second branch. The integrated output results from combining information from all three branches. Refer to Fig. 5 for a detailed illustration.

In MSCA attention, decomposing the global pooling into a pair of one-dimensional coding operations can be represented by Equation (1). Where Z_c denotes the output in the height or width direction, H and W denote the height and width respectively, i and j are denoted as one of their specific channels, and x_c denotes the input feature map.

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

Specifically, 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), respectively. Therefore, the outputs in the height and width directions can be represented by Equations (2) and (3), 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.

$$Z_c^h(h) = \frac{1}{W} \sum_{0 \le i \le W} x_c(h,j) \tag{2}$$

$$Z_c^W(w) = \frac{1}{H} \sum_{0 \le j \le H} x_c(j,w) \tag{3}$$

The obtained outputs in the width and height directions are spliced to create a tensor with the same number of channels as the input. Subsequently, the tensor undergoes convolution, passes through the activation function, and is then separated into outputs g_c^h and g_c^w in the X and Y directions, respectively (where g_c^h represents the height and g_c^w represents the width). Simultaneously, the tensor is split into weighting coefficients H_c^h and W_c^w in the X and Y directions, compensating for information loss in the intermediate process and guiding the outputs in the X and Y directions. This can be expressed in Equations (4) and (5), 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.

$$g_c^{'h}(i) = g_c^h(i) \times W_c^h(i) \tag{4}$$

$$g_c^{'w}(j) = g_c^w(j) \times W_c^w(j) \tag{5}$$

Therefore, the output of the MSCA attention can be expressed as Equation (6), where x_c denotes the feature map of the original input, and a_c is the globally complemented information under global pooling.

$$y_c(i,j) = x_c(i,j) \times g_c^{'h}(i) \times g_c^{'w}(j) \times a_c(i,j) \tag{6}$$

3.3 Contextual Information Fusion Downsampling Module (CFD)

The CFD module plays a key role in downsampling and information integration, ensuring the timely incorporation of information post-downsampling. By introducing joint features that combine local characteristics with contextual semantic information from the surroundings and refining these features with global contextual information, the CFD module effectively integrates information after downsampling, thereby enhancing information utilization efficiency. Positioned before the DCM module and spanning the entire network from spatial to semantic levels, the CFD module enables the prompt transfer of more accurate features to the DCM module for further processing, preventing information wastage.

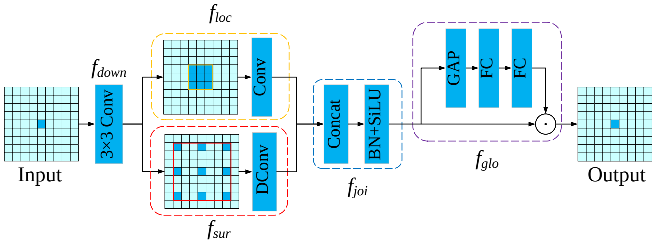


Figure 6: Diagram illustrating the CFD

The CFD module comprises five components: (1) the downsampling operation f_{down} implemented as a 3×3 convolution; (2) the local feature extractor f_{loc}, a regular convolution gathering local features; (3) the surrounding context feature extractor f_{sur}, an inflationary convolution capturing contextual features; (4) the joint feature extractor f_{joi}, incorporating a splicing layer, Batch Normalization layer, and SilU activation function to fuse local and contextual features; (5) the global feature extractor f_{glo}, housing a global pooling layer and two fully-connected layers to extract features, generating a weight vector for guiding joint feature fusion. The detailed diagram of the CFD module is presented in Figure 6.

The CFD module design is inspired by the Context Guided Block (CG Block) used in the lightweight semantic segmentation model, Context Guided Network (CGNet) [Wu *et al.*, 2020]. In CGNet, the CG block serves as the primary convolutional module, extracting features by combining surrounding contextual information with local features. While suitable for a lightweight model, CG blocks may not exhibit a particularly pronounced feature extraction capability. To maximize the benefits of contextual information utilization, the CFD module incorporates downsampling operations, strategically positioned before the main convolutional module to facilitate comprehensive contextual information integration. The fused information is then input to the main convolutional module, enhancing feature extraction and improving the model's overall information extraction capability without a significant increase in computational load.

4 Experiment evaluation

4.1 Experiment setups

Datasets

In this thesis, the performance of YOLO-DC is validated through extensive experiments on the Microsoft COCO 2017 dataset [Lin *et al.*, 2014]. To comprehensively evaluate the model, additional experiments were conducted on the RUOD underwater object detection dataset [Fu *et al.*, 2023] and the PASCAL VOC dataset (07+12). In the ablation study, the COCO 2017 dataset was still utilized for the experiments.

Method	AP^{val}	AP^{val}_{50}	Params	FLOPs
YOLOv5-N	28.0%	45.7%	1.9 M	4.5 G
YOLOv5-S	37.4%	56.8%	7.2 M	16.5 G
YOLOv5-M	45.4%	64.1%	21.2 M	49.0 G
YOLOv6-N	37.0%	52.4%	4.7 M	11.4 G
YOLOv6-S	44.3%	61.2%	18.5 M	45.3 G
YOLOv6-M	49.1%	66.1%	34.9 M	85.8 G
YOLOv7-Tiny	37.4%	55.2%	6.2 M	13.7 G
YOLOvX-N	32.8%	50.3%	5.1 M	6.5 G
YOLOvX-S	40.5%	59.3%	9.0 M	26.8 G
YOLOvX-M	46.9%	65.6%	25.3 M	73.8 G
YOLOv8-N	37.3%	52.6%	3.2 M	8.7 G
YOLOv8-S	44.9%	61.8%	11.2 M	28.6 G
YOLOv8-M	50.2%	67.2%	28.9 M	78.9 G
Gold-YOLO-N	39.9%	55.9%	5.6 M	12.1 G
Gold-YOLO-S	46.1%	63.3%	21.5 M	46.0 G
Gold-YOLO-M	**50.9%**	**68.2%**	41.3 M	87.5 G
YOLO-DC-N	**40.8%**	**56.9%**	3.9 M	8.9 G
YOLO-DC-S	**46.6%**	**63.5%**	13.9 M	29.2 G
YOLO-DC-M	50.4%	67.3%	32.9 M	70.9 G

Table 1: Comparison of YOLO-DC with other state-of-the-art (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. All model metrics were assessed with an input resolution of 640 × 640.

The Microsoft COCO 2017 dataset is a comprehensive public dataset for object detection, including a training set of 118,287 images and an additional 5,000 test images. The PASCAL VOC dataset (07+12) merges the training and validation sets of PASCAL VOC 2007 and PASCAL VOC 2012, utilizing the test set from PASCAL VOC 2007, with 16,551 images in the training set and 4,952 in the test set. To address diverse challenges in underwater detection, the RUOD dataset's training set comprises 9,800 images, and the test set comprises 4,200 images. This configuration facilitates a comprehensive evaluation of detector performance.

Implementation details

The experiments utilize YOLOv8 as the baseline model, closely aligning with its training configurations, except for the integrated enhancements. YOLO-DC undergoes 500 epochs of training, adopting the remaining hyperparameters from YOLOv8's default settings. Pre-training weights are avoided in all experiments to ensure the model is trained from scratch for precise performance evaluation. Single-scale images (640×640) serve as inputs for consistency, and the evaluation metrics adhere to COCO standards. The reported results emphasize standardized average precision (AP) across different Intersection over Union (IoU) thresholds.

All experiments with the models were conducted using 2 NVIDIA RTX 3090 GPUs. The model code is written in Python, and the deep learning framework used is PyTorch 2.0.

4.2 Comparisons

YOLOv5 [Jocher, 2020], YOLOv6 [Li *et al.*, 2022], YOLOv7 [Wang *et al.*, 2023b], YOLOX [Ge *et al.*, 2021], Gold-YOLO [Wang *et al.*, 2023a], and the baseline model YOLOv8 [Jocher *et al.*, 2023] were chosen for comparative experiments with the YOLO-DC model on the Microsoft COCO 2017 dataset under the same experimental settings.

Method	AP^{val}	AP^{val}_{50}	Params	FLOPs
YOLOv5-N	45.1%	72.5%	1.9 M	4.5 G
YOLOv5-S	53.0%	76.2%	7.2 M	16.5 G
YOLOv6-N	60.5%	82.0%	4.7 M	11.4 G
YOLOv7-Tiny	53.8%	79.3%	6.2 M	13.7 G
YOLOv8-N	59.1%	79.9%	3.2 M	8.7 G
YOLO-DC-N	**62.6%**	**82.4%**	3.9 M	8.9 G

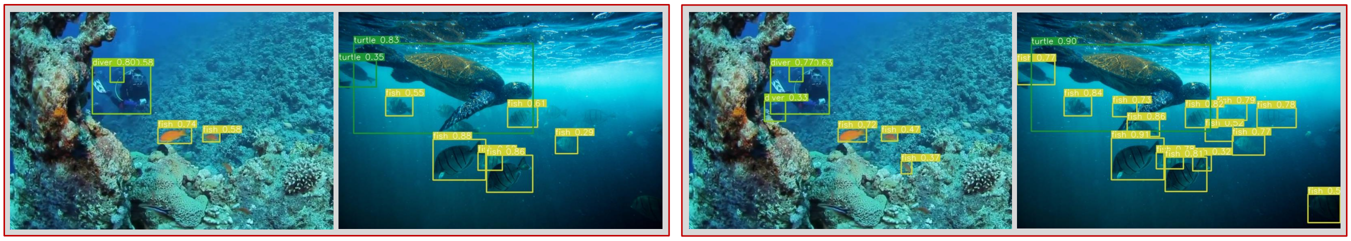
Table 2: Comparison of YOLO-DC-N with other state-of-the-art (SOTA) models on the joint PASCAL VOC dataset (07+12).

The experimental results in Table 1 demonstrate that each version of YOLO-DC outperforms other models on the Microsoft COCO 2017 dataset, showcasing significant improvements. Enhanced information utilization from the CFD module and robust feature extraction by the DCM module contribute to YOLO-DC achieving a 3.5% increase in AP, reaching 40.8%, with comparable computation and a similar parameter count. Notably, compared to YOLOv8-S and YOLOv8-M, YOLO-DC exhibits AP increases of 1.7% and 0.2%, respectively. In comparison to YOLOv5-N, YOLOv6-N, YOLOv7-Tiny, YOLOX-N, and Gold-YOLO-N, YOLO-DC achieves AP boosts of 12.8%, 3.8%, 3.4%, 8%, and 0.9%, respectively. Moreover, the S and M versions of YOLO-DC demonstrate significant AP improvements over the corresponding S and M versions of these SOTA models. While YOLO-DC shows a marginal increase in parameters compared to YOLOv8, it achieves a about 1% to 3% AP boost, considered an acceptable trade-off. Despite Gold-YOLO's use of self-distillation during training, both N and S versions of YOLO-DC consistently outperform it, with only a slightly lower AP for YOLO-DC-L. Importantly, the computational and parametric quantities of YOLO-DC are lower than those of Gold-YOLO.

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), YOLOv6-N, YOLOv7-Tiny, and the baseline model YOLOv8-N.

Table 2 presents the experiments conducted on the PASCAL VOC dataset (07+12). The results indicate a significant enhancement in average precision (AP) with YOLO-DC-N, reaching 62.6%, reflecting a 3.5% improvement compared to YOLOv8-N. This outperforms all other selected state-of-the-art 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.

Table 3 shows the experimental results on the RUOD dataset. Benefiting from deformable convolution's exceptional performance in handling geometric deformations, YOLO-DC consistently surpasses other SOTA models when assessed on the RUOD dataset. In comparison to YOLOv8-N, YOLO-DC-N attains a 1.1% increase in average precision



(a) YOLOv8

(b) YOLO-DC

Figure 7: Comparison of image detection results selected from the RUOD dataset. (a) is the detection result of YOLOv8. (b) is the detection result of YOLO-DC.

Method	AP^{val}	AP^{val}_{50}	Params	FLOPs
YOLOv5-N	53.6%	72.1%	1.9 M	4.5 G
YOLOv5-S	58.8%	79.4%	7.2 M	16.5 G
YOLOv6-N	59.7%	84.2%	4.7 M	11.4 G
YOLOv7-Tiny	57.2%	85.3%	6.2 M	13.7 G
YOLOv8-N	61.9%	85.3%	3.2 M	8.7 G
YOLO-DC-N	**62.8%**	**85.6%**	3.9 M	8.9 G

Table 3: Comparison of YOLO-DC-N with other state-of-the-art (SOTA) models on the underwater dataset RUOD.

Method	AP^{val}	AP^{val}_{50}	Params	FLOPs
YOLOv8(Baseline)	37.3%	52.6%	3.2 M	8.7 G
YOLOv8(DCNv2)	38.6%	54.4%	3.2 M	7.4 G
YOLOv8(DCM)	39.0%	55.0%	3.2 M	7.9 G
YOLOv8(CFD)	38.4%	53.8%	3.8 M	9.8 G
YOLO-DC(DCM+CFD)	**40.8%**	**56.9%**	3.9 M	8.9 G

Table 4: Ablation Study on Different Modules Using YOLOv8-N as the Baseline Model: Evaluating Performance on the Microsoft COCO 2017 Dataset.

(AP), reaching 62.8%. Notably, despite having half the parameters of YOLOv5-S, YOLO-DC-N achieves a 4% higher AP. Figure 7 illustrates the comparison of detection results between YOLO-DC and the baseline model YOLOv8, highlighting YOLO-DC's significantly superior detection capabilities in challenging environments.

Experiments demonstrate that the YOLO-DC model outperforms multiple SOTA algorithms on multiple datasets in terms of the number of parameters, computational effort, and detection accuracy.

4.3 Ablation study

In order to validate the effectiveness of the DCM module, and CFD module, experiments were conducted on the Microsoft COCO 2017 dataset for each module separately. Detailed experimental data is provided in Table 4.

In this study, YOLOv8-N serves as the baseline model with an initial AP of 37.3%. Replacing the convolution in the C2f module with DCNv2 boosts the model's AP by 1.3% to 38.6%, indicating the expanded receptive field and improved feature extraction capabilities of DCNv2 convolution over regular convolution. Independently using the DCM module raises the AP by 1.7% to 39.0% compared to the baseline, showcasing the superior information extraction capabilities of DCN-MSCA convolution and confirming the efficacy of MSCA attention. When applied independently, the CFD module enhances the model's AP by 1.1%, reaching 38.4% compared to the baseline, affirming the significant performance benefits of timely integration of characteristic information from both surrounding and global contexts into the main convolution module post-downsampling.

Integration of the DCM and CFD modules in the YOLO-DC model yielded a 3.5% boost in AP, achieving 40.8% compared to the baseline. The CFD module enhances information utilization by incorporating contextual details, while the DCM module, serving as the primary convolution module, broadens the receptive field, enhancing feature extraction. In this configuration, the DCM module manages feature extraction, and the CFD module, strategically positioned before the DCM module in the model's backbone network, facilitates efficient transmission of crucial information for timely feature extraction. The synergistic effect explains the improvement with both modules surpassing the sum of individual improvements. At this stage, the model's parameter count increased by 0.7 M, and computational load rose by 0.2 G. This increase is attributed to additional computational overhead from the MSCA Attention and CFD modules. Nevertheless, the rise in computational overhead seems negligible compared to the substantial performance improvement.

5 Conclusion

This paper introduces YOLO-DC, an object detection model utilizing deformable convolution. YOLO-DC features a novel DCM and a CFD module compared to the baseline model. Addressing a common oversight in contemporary object detection, where the convolution's inherent improvement potential is frequently disregarded, DCM integrates deformable convolution with multi-scale spatial channel attention, enhancing its feature extraction. Meanwhile, the CFD module optimally leverages local and surrounding context, reducing information loss post-downsampling. Integrating information prior to the main convolution module aids in feature extraction, contributing to heightened model accuracy. These advancements position YOLO-DC ahead of other SOTA models in the field.

A plethora of experiments demonstrate that the proposed model excels not only in general scenarios but also in underwater object detection. This provides a robust solution for research and practical applications in related fields, with the potential for further advancements through subsequent research.

References

[Cao *et al.*, 2019] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[Carion *et al.*, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, pages 213–229. Springer, 2020.

[Chen *et al.*, 2018a] Liang-Chieh Chen, Maxwell Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jon Shlens. Searching for efficient multi-scale architectures for dense image prediction. *Advances in Neural Information Processing Systems (NIPS)*, 31, 2018.

[Chen *et al.*, 2018b] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. A^ 2-nets: Double attention networks. *Advances in Neural Information Processing Systems (NIPS)*, 31, 2018.

[Dai *et al.*, 2017] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017.

[Ding *et al.*, 2018] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2393–2402, 2018.

[Feng *et al.*, 2021] Chengjian Feng, Yujie Zhong, Yu Gao, Matthew R Scott, and Weilin Huang. Tood: Task-aligned one-stage object detection. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3490–3499. IEEE Computer Society, 2021.

[Fu *et al.*, 2023] Chenping Fu, Risheng Liu, Xin Fan, Puyang Chen, Hao Fu, Wanqi Yuan, Ming Zhu, and Zhongxuan Luo. Rethinking general underwater object detection: Datasets, challenges, and solutions. *Neurocomputing*, 517:243–256, 2023.

[Gao *et al.*, 2019] Zilin Gao, Jiangtao Xie, Qilong Wang, and Peihua Li. Global second-order pooling convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3024–3033, 2019.

[Ge *et al.*, 2021] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[Girshick *et al.*, 2014] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[Girshick, 2015] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[Hou *et al.*, 2021] Qibin Hou, Daquan Zhou, and Jiashi Feng. Coordinate attention for efficient mobile network design. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13713–13722, 2021.

[Hu *et al.*, 2018a] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. *Advances in Neural Information Processing Systems (NIPS)*, 31, 2018.

[Hu *et al.*, 2018b] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7132–7141, 2018.

[Huang *et al.*, 2019] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 603–612, 2019.

[Jocher *et al.*, 2023] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLOv8. https://github.com/ultralytics/ultralytics, January 2023.

[Jocher, 2020] Glenn Jocher. YOLOv5 by Ultralytics. https://github.com/ultralytics/yolov5, May 2020.

[Li *et al.*, 2019] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 510–519, 2019.

[Li *et al.*, 2022] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, et al. Yolov6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 2022.

[Li *et al.*, 2023] Xiaofei Li, Jiaxin Yang, Shuohao Li, Jun Lei, Jun Zhang, and Dong Chen. Locate, refine and restore: a progressive enhancement network for camouflaged object detection. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1116–1124, 2023.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[Liu *et al.*, 2020] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Changhu Wang, and Jiashi Feng. Improving convolutional networks with self-calibrated convolutions. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 10096–10105, 2020.

[Park *et al.*, 2019] Jiwoong Park, Minsik Lee, Hyung Jin Chang, Kyuewang Lee, and Jin Young Choi. Symmetric graph convolutional autoencoder for unsupervised graph representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6519–6528, 2019.

[Redmon and Farhadi, 2017] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7263–7271, 2017.

[Redmon and Farhadi, 2018] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[Redmon *et al.*, 2016] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems (NIPS)*, 28, 2015.

[Sun *et al.*, 2022] Yujia Sun, Shuo Wang, Chenglizhao Chen, and Tian-Zhu Xiang. Boundary-guided camouflaged object detection. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1335–1341, 2022.

[Wang *et al.*, 2018] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, 2018.

[Wang *et al.*, 2020] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. Cspnet: A new backbone that can enhance learning capability of cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[Wang *et al.*, 2021] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13029–13038, June 2021.

[Wang *et al.*, 2023a] Chengcheng Wang, Wei He, Ying Nie, Jianyuan Guo, Chuanjian Liu, Yunhe Wang, and Kai Han. Gold-yolo: Efficient object detector via gather-and-distribute mechanism. In *Thirty-seventh Conference on Neural Information Processing Systems (NIPS)*, 2023.

[Wang *et al.*, 2023b] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark 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)*, pages 7464–7475, 2023.

[Wang *et al.*, 2023c] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng 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)*, pages 14408–14419, 2023.

[Woo *et al.*, 2018] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[Wu *et al.*, 2020] Tianyi Wu, Sheng Tang, Rui Zhang, Juan Cao, and Yongdong Zhang. Cgnet: A light-weight context guided network for semantic segmentation. *IEEE Transactions on Image Processing (TIP)*, 30:1169–1179, 2020.

[Zhang *et al.*, 2022] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2022.

[Zhu *et al.*, 2019] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9308–9316, 2019.