Neurocomputing 625 (2025) 129507

Contents lists available at ScienceDirect

Neurocomputing

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

# Attention-modulated frequency-aware pooling via spatial guidance ☆

Yunzhong Si [a], Huiying Xu [a,b,*], Xinzhong Zhu [a,b,c], Rihao Liu [a], Hongbo Li [c]

[a] College of Computer Science and Technology, Zhejiang Normal University, Jinhua, 321004, China
[b] Research Institute of Hangzhou Artificial Intelligence, Zhejiang Normal University, Hangzhou, 311231, China
[c] Beijing Geekplus Technology Co., Ltd, Beijing, 100101, China

ARTICLE INFO

Communicated by W. Wang

*Keywords:*
Multi-domain pooling
Frequency domain learning
Multi-scale spatial guidance
Effective receptive field

ABSTRACT

Pooling is widely used in computer vision to expand the receptive field and enhance semantic understanding by reducing spatial resolution. However, current mainstream downsampling methods primarily rely on local spatial aggregation. While they effectively reduce the spatial resolution of feature maps and extract discriminative features, they are still limited by the constraints of the receptive field and the inadequacy of single-domain information, making it challenging to effectively capture fine details while suppressing noise. To address these limitations, we propose a Dual-Domain Downsampling (D3) method, which leverages the complementarity of spatial and frequency domains. We employ an invertible local two-dimensional Discrete Cosine Transform (2D DCT) transformation to construct a frequency domain pooling window. In the spatial domain, we design an Inverted Multiform Attention Modulator (IMAM) that expands the receptive field through multiform convolutions, while adaptively constructing dynamic frequency weights guided by rich spatial information. This allows for fine-grained modulation of different frequency components, either amplifying or attenuating them in different spatial regions, effectively reducing noise while preserving detail. Extensive experiments on ImageNet-1K, MSCOCO, and complex scene detection datasets across various benchmark models consistently validate the effectiveness of our approach. On the ImageNet-1K classification task, our method achieve up to a 1.95% accuracy improvement, with significant performance gains over state-of-the-art methods on MSCOCO and other challenging detection scenarios. The code will be made publicly available at: https://github.com/HZAI-ZJNU/D3.

## 1. Introduction

In convolutional neural networks (CNNs), spatial pooling layers reduce spatial resolution by aggregating information from local $K \times K$ regions in the feature maps. This process is pivotal in constructing multi-scale features, promoting scale invariance, and retaining critical spatial details, thus playing a key role in enhancing the network's overall performance. It is widely used in computer vision tasks that require multi-level semantic information and discriminative details, such as image classification [1,2], object detection [3–8] and semantic segmentation [9–12].

Although traditional pooling methods, such as average pooling [14], max pooling [15], and strided convolution, are fast and memory-efficient, they have limitations in extracting discriminative features and expanding receptive fields. Recently, according to the local importance theory, LIP [16] analyzes how to enhance discriminative features during downsampling from the perspective of local importance. Building on this, SoftPool [17] leverages the advantages of parameter-free softmax to enhance and suppress spatial pixels within each pooling window. To further expand the receptive field, Self-Attentive Pooling [18] utilizes multi-head self-attention (MHSA) [19,20] at the patch level to model long-range dependencies among non-local patches.

Nevertheless, the aforementioned methods only partially overcome the limitations of traditional pooling [14,15,21–26] in terms of discriminative features and receptive fields. For example, LIP [16] and SoftPool [17] address the problem solely from the perspective of local enhancement, Self-Attentive Pooling [18], while modeling the global receptive field, incurs higher computational costs and memory overhead due to the quadratic complexity of MHSA with respect to sequence length, as well as the potential inclusion of non-target background information. Additionally, these methods only consider the spatial

☆ This work was supported by the National Natural Science Foundation of China (62376252); Key Project of Zhejiang Provincial Natural Science Foundation, China (LZ22F030003).

* Corresponding author at: College of Computer Science and Technology, Zhejiang Normal University, Jinhua, 321004, China.
*E-mail addresses:* siyunzhong@zjnu.edu.cn (Y. Si), xhy@zjnu.edu.cn (H. Xu), zxz@zjnu.edu.cn (X. Zhu), Lrihao0413@zjnu.edu.cn (R. Liu), jason.li@geekplus.com (H. Li).

https://doi.org/10.1016/j.neucom.2025.129507
Received 9 November 2024; Received in revised form 31 December 2024; Accepted 18 January 2025
Available online 27 January 2025

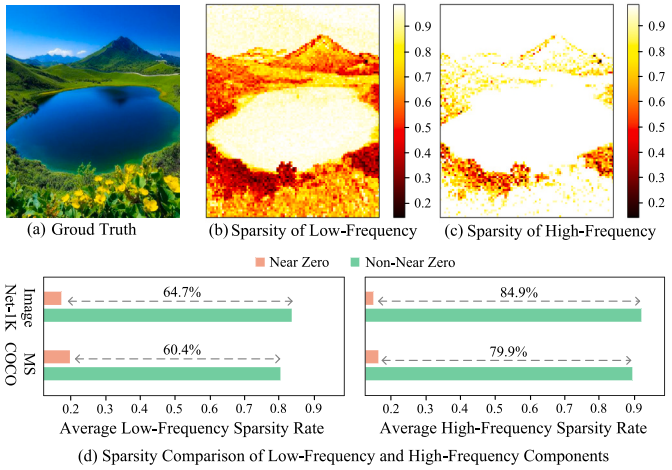



**Fig. 1.** Visualization of frequency domain sparsity characteristics. For the input image (a), figures (b) and (c) illustrate the low-frequency and high-frequency sparsity ratio maps obtained from the local 2D DCT [13] transformation. The results indicate that most regions exhibit significant sparsity characteristics. (d) shows the average sparsity ratios of high-frequency and low-frequency components across the ImageNet-1K [1] and MSCOCO [3] benchmarks. The results indicate that high-frequency components are sparser than low-frequency ones, with non-near-zero high-frequency components potentially containing both details and noise.

domain information, where distinguishing between details and noise remains challenging. On one hand, the local windows extended from traditional pooling have small receptive fields and apply a uniform pooling strategy across all regions, which hinders the utilization of larger contextual information to aid the aggregation process. On the other hand, texture details and noise often have similar pixel values in the spatial domain. This makes it difficult to distinguish subtle structural or pattern differences between them when relying solely on spatial domain processing. Without the introduction of learnable parameters, such as those in AvgPool [14], MaxPool [15], SoftPool [17], and ConditionalPool [27], this issue becomes even more challenging.

In this paper, we conduct an in-depth analysis of the respective advantages of the frequency domain and spatial domain in feature processing: the frequency domain can generate components of varying frequency intensities, allowing further decomposition of high-frequency components such as details and noise, while the spatial domain, through efficient convolutional networks, not only extracts detailed features but also provides richer contextual information to guide the aggregation of different frequency components. Additionally, as illustrated in Fig. 1(a), (b), and (c), we visualize the frequency sparsity ratio of each $8 \times 8$ region in the image, revealing that the image exhibits sparsity in the frequency domain, closely resembling the feature maps. Based on these insights, we propose a novel dual-domain downsampling method (D3). Unlike traditional downsampling, we redefine the pooling operation from spatial windows to frequency windows, meaning that we can perform local aggregation in the frequency domain. Specifically, we first decompose the feature map into $P \times P$ local regions and apply an efficient two-dimensional discrete cosine transform [13] (2D DCT) to each, converting spatial signals into frequency representations to construct frequency-domain windows. Since the basis functions of DCT are fixed, prior research [28] has shown that average pooling is proportional to the lowest frequency component of the 2D DCT. This suggests that average pooling primarily retains low-frequency information while potentially weakening the model's ability to capture discriminative high-frequency features. To further investigate the distribution of different frequency components, we apply 2D DCT to the ImageNet-1K [1] and MSCOCO [3] benchmarks to compute the average sparsity rates of frequency components. Our results reveal that both low-frequency and high-frequency components exhibit varying degrees of sparsity, as shown in Fig. 1(d). Consequently, we provide new insights into the design of frequency filters, framing it as a problem of optimizing the sparsity of different frequency components, and we design an Inverted Multiform Attention Modulator (IMAM) with a larger multi-scale receptive field to function as a frequency filter, adaptively assigning dynamic weights to different frequency components using richer contextual information. Finally, the aggregated modulated frequency components are transformed back into the spatial domain. By modulating the energy of different frequency components with richer context, for different local regions, we can either strengthen the low-frequency components to enlarge the effective receptive field or enhance high-frequency components to capture fine details. To validate the effectiveness of our proposed pooling method, we replace the pooling layers in various backbone networks [2,29–31], as well as in advanced detectors [4–7] and real-time detectors [8,32–35] with our D3 module. Extensive experiments on image classification and object detection tasks demonstrate significant improvements across different benchmark datasets [1,3,36–39]. Our contributions can be summarized as follows:

- We analyze the limitations of existing pooling methods that rely solely on the spatial domain for detail and noise recognition, explore the advantages of the spatial and frequency domains in feature extraction, and redefine spatial pooling as frequency domain pooling guided by the spatial domain.
- We propose a dual-domain pooling method (D3), combining the reversible and efficient 2D DCT with the context-rich Inverted Multiform Attention Modulator (IMAM). The 2D DCT enables efficient and reversible transformations between the spatial and frequency domains for each local region, while IMAM, as a frequency filter, uses depth-wise convolution with multi-scale receptive field to capture the unique semantic patterns of each feature channel, adaptively modulating the energy of different frequency components within varying frequency windows.
- We validate our pooling method through comprehensive experiments on image classification and object detection, consistently demonstrating its effectiveness and generality, surpassing existing state-of-the-art pooling methods with a simple replacement of the pooling layers in different networks.

## 2. Related work

### 2.1. Spatial pooling

Most CNN backbone networks utilize pooling layers for feature aggregation, reducing resolution while enhancing semantic levels. Classic architectures such as VGG [40], ResNet [2], MobileNets [29,41–43], CSPNet [44], EfficientNets [45,46], RepVGG [30] and MobileOne [31] employ average pooling [14], max pooling [15], or strided convolutions with a stride greater than 1 for hierarchical downsampling. However, these traditional pooling methods are constrained by local receptive fields, which may not adequately retain useful discriminative features. Additionally, as the convolutional receptive field increases, the corresponding pooling stride also grows, potentially leading to the loss of important information. This phenomenon has prompted the emergence of several novel approaches in recent years. Stochastic Pooling [21] and S3Pool [25] retain representative pixel information through probability distribution and feature importance, respectively. LIP [16], founded on a locally learnable importance theory, captures key information, while the parameter-free weighting method via Softmax [17] has been proven to highlight salient features in image classification and video action recognition. Additionally, Conditional Pooling [27] dynamically assesses pixel distribution relative to the mean, optimizing the aggregation in average pooling [14]. To further expand receptive fields, Self-Attentive Pooling [18] leverages a global multi-head self-attention (MHSA) [19] mechanism to model long-range

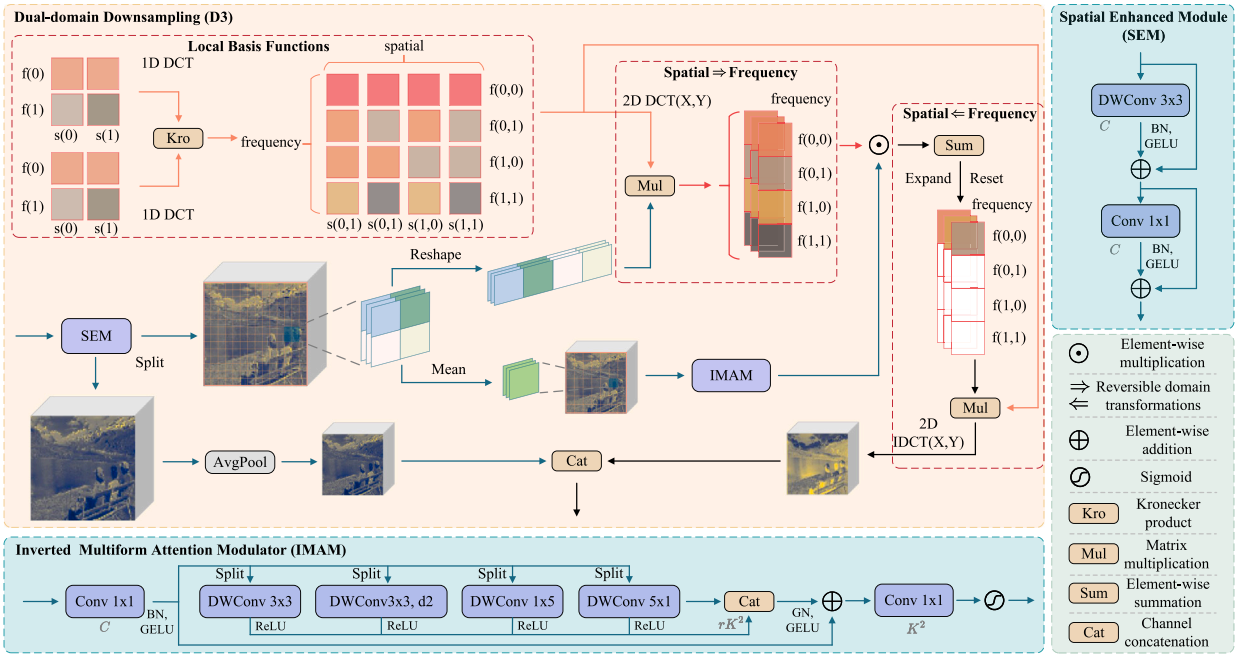



**Fig. 2.** Overview of the proposed Dual-Domain Downsampling Method (D3). "Split" refers to splitting along the channel dimension, "Mean" indicates averaging over local regions, and "Reshape" denotes feature reorganization. The red dashed box indicates the 2D DCT basis functions and the invertible frequency domain transformation.

dependencies among patches. Despite their demonstrated effectiveness, these methods are limited by either the lack of semantic richness in local receptive fields or the introduction of extraneous background information in global receptive fields. Therefore, in our work, we explore the practicality and effectiveness of lightweight multi-scale contextual information from neighboring regions, combined with inductive bias of locality [47], to extract more relevant feature information.

### 2.2. Frequency domain representation learning

Traditional signal processing predominantly relies on frequency domain analysis for feature extraction and processing [48,49]. Frequency domain methods have been extensively utilized in deep learning to assess optimization strategies [50] and the generalization capabilities [51] of deep neural networks. Moreover, these techniques have been integrated into downstream visual tasks, such as image deblurring [52,53] and image reconstruction [54–56], with the goal of dynamically balancing low-frequency and high-frequency components through various transformation functions [13,57–59], from which noise is reduced and model performance is improved. Recent study [60] has also shown that frequency domain awareness can adaptively adjust the dilation rate of convolutions. Therefore, our method offers a dual perspective based on frequency and spatial domains for pooling methods, leveraging rich contextual information in the spatial domain to effectively modulate different local frequency components in the frequency domain, thereby addressing the limitations of single spatial domain aggregation.

## 3. Method

The overall structure of the proposed D3 is shown in Fig. 2. In this section, we begin by revisiting the reversible DCT formula and explaining the construction of frequency windows. Then, we introduce frequency sparsity and delve into the implementation details of IMAM, with a focus on how region-aware contextual information is leveraged to optimize the energy distribution of different components within the frequency domain windows. Finally, based on the sequence of local aggregation and inverse transform, we propose two local aggregation strategies across different domains.

### 3.1. Frequency window division based on DCT

In this work, we adopt the reversible DCT [13] as a bridge between the spatial and frequency domains, primarily due to its simplicity and efficiency in decomposing spatial signals into different frequency components. Specifically, for a given input $X \in \mathbb{R}^{C\times H\times W}$, we first divide it into local windows of size $K \times K$ in the spatial domain, denoted as $X^{(i,j)} \in \mathbb{R}^{C\times K\times K}$, where $i \in \{0, 1, \dots, H' - 1\}$, $j \in \{0, 1, \dots, W' - 1\}$, with $W' = \frac{W}{K}$, $H' = \frac{H}{K}$. Based on these spatial windows, we define the local 2D DCT basis functions as follows:

$$C^{(x,y)}_{(u,v)} = \alpha(u)\alpha(v)cos[\frac{\pi u(2x+1)}{2K}]cos[\frac{\pi v(2y+1)}{2K}] \tag{1}$$

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{K}}, & if\ u = 0 \\ \sqrt{\frac{2}{K}}, & if\ u \neq 0 \end{cases} \tag{2}$$

Hear, $u$ and $v$ are represent the frequency components along the $H$ and $W$ dimensions respectively, and $\alpha(\cdot)$ denotes the normalization factor. The process of constructing the frequency domain window based on local 2D DCT can be written as:

$$F^{(i,j)}_{(u,v)} = \sum_{x=0}^{K-1}\sum_{y=0}^{K-1} X^{(i,j)}_{(x,y)} C^{(x,y)}_{(u,v)} \tag{3}$$

where $F^{(i,j)}_{(u,v)} \in \mathbb{R}^{K\times K}$ represents the 2D DCT frequency coefficient matrix corresponding to the region indexed by $(i, j)$. Accordingly, the 2D inverse DCT (IDCT) from the frequency domain back to the spatial domain is expressed as follows:

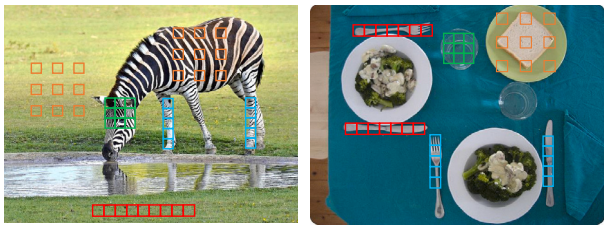
$$X^{(i,j)}_{(x,y)} = \sum_{u=0}^{K-1}\sum_{v=0}^{K-1} F^{(i,j)}_{(u,v)} C^{(x,y)}_{(u,v)} \tag{4}$$

Notably, in our implementation, as shown in Fig. 2, we apply two local 1D DCTs and compute their Kronecker product [61] to efficiently construct the 2D DCT basis functions.

### 3.2. Spatially-guided frequency filters

*Sparsity of local frequency components*

Although we employ the efficient 2D DCT to transform each $K \times K$ spatial block into the frequency domain, the basis function coefficients



(a) (b)

**Fig. 3.** Illustrations of diverse receptive field shapes for distinct spatial semantic information. Different colors are used to annotate distinct perceptual shapes.

in DCT are fixed and do not adapt dynamically to the input signal, resulting in the frequency characteristics being represented as linear or nearly linear. This constrains the ability to effectively distinguish between details and noise in different high-frequency components and lacks adaptability to sparse signals. As shown in Fig. 1, we observe that real-world images, due to varying semantic richness across spatial regions, should exhibit a highly sparse energy distribution when transformed from the spatial to the frequency domain. Frequency domain sparsity refers to the concentration of energy in a few frequency components, while other components have very small or near-zero coefficients. For example, in a smoothly gradient sky, the energy is mainly concentrated in the low-frequency components, whereas for dense leaves with complex textures, it is focused in the high-frequency components. To finely differentiate between details and noise in the high-frequency component, as well as dynamically adjust the energy weighting of low-frequency components to better integrate high receptive field features, we need to enhance the sparsity characteristics of these components. To this end, we approach frequency filtering design as a problem of optimizing sparsity across different frequency components and propose a novel context-aware Inverted Multiform Attention Modulator (IMAM).

*Spatial Information Enhancement (SEM)*

As indicated by prior studies [62,63], pooling operations are typically associated with halving feature resolution and deepening semantic hierarchies, inevitably leading to some information loss, and may even induce shifts in gradient flow [34]. Considering that convolution can be regarded as a selective enhancement process in feature extraction, and local convolution is often treated as a high-pass filter that amplifies high-frequency signals [64], we propose a simple residual-based Spatial Information Enhancement Module (SEM), which first applies a $3 \times 3$ depthwise convolution to enhance spatial information, followed by a $1 \times 1$ regular convolution to facilitate channel interaction. The enhanced features are then split and fed into IMAM for adaptive modulation of local frequency components. Additionally, we employ residual connections to enrich the gradient paths. This approach injects finer-grained information into the subsequent modulation of high-frequency components by IMAM. The specific process is as follows:

$$X_{s1} = X + \mathcal{G}(BN(DWConv_{3\times3}^{C\to C}(X))) \tag{5}$$

$$X_{s2} = X_{s1} + \mathcal{G}(BN(Conv_{1\times1}^{C\to C}(X_{s1}))) \tag{6}$$

where $\mathcal{G}$, $BN$, $DWConv$ and $Conv$ represent GELU, Batch Normalization, depthwise convolution and regular convolution, respectively. Superscript $^{C\to C}$ denotes the transformation from input channels to output channels, and subscript $_{3\times3}$ represents the kernel size.

*Inverted multiform attention modulator*

Based on the analysis of the sparse characteristics of local frequency components in Fig. 1, a critical issue arises: how to selectively enhance the sparsity of different frequency bands for each semantic part $X^{(i,j)}$. Traditional frequency filtering methods typically operate on the global space, selecting fixed thresholds based on specific tasks [65], adaptively adjusting frequency responses using simple composite coefficients, or even utilizing high-cost NAS searches [28] to find the optimal filter. In contrast, we consider that a larger receptive field offers greater robustness when dealing with local noise, allowing for better decision-making by leveraging appropriate surrounding regions. To this end, we propose a novel Inverted Multiform Attention Modulator (IMAM), which utilizes richer spatial contextual information to generate dynamic weights, thereby effectively modulating different frequency components. Specifically, we first apply spatial pooling over each $K \times K$ local region to align with its frequency window, expanding the receptive field while retaining a certain level of low-frequency information, i.e., the primary semantic content. Subsequently, the features are evenly split along the channel dimension into two parts: one part captures the information from spatial pooling, while the other part is processed in the frequency domain, thereby reducing the computational overhead introduced by the frequency transformation.

$$P^{(i,j)} = X_p^{(i,j)} = \mathcal{P}(X_{s2}^{(i,j)}) \tag{7}$$

$$\begin{aligned} X_l, X_r &= Split(X_p, 2) \\ &= X_{:\frac{C}{2},:}, X_{\frac{C}{2}:,:} \end{aligned} \tag{8}$$

Here, $\mathcal{P}(\cdot)$ denotes the average pooling, and $Split(X_p, N)$ represents the uniform splitting of $X_p$ into $N$ equal parts along the channel dimension. After generating an aggregation point for each region, we center at pixel $P^{(i,j)}$ and combine depthwise convolutions of varying shapes and sizes with two pointwise convolutions of size $1 \times 1$ to efficiently capture contextual information from the appropriate neighborhood of $P^{(i,j)}$. Specifically, as shown in Fig. 3, objects with different shapes or semantics exhibit distinct perceptual regions, which capture both local and global features of the target. To effectively leverage these diverse semantic features, we decompose the feature map into multiple non-overlapping sub-features and apply lightweight depthwise convolutions with varying receptive fields to extract complementary sub-features with different semantic patterns. These sub-features are then combined using a $1 \times 1$ convolution for dimensionality reduction, followed by a sigmoid normalization function to generate dynamic weights for the frequency components. Our IMAM adaptively adjusts the weights of frequency components based on the semantic information of the input features. Each sub-feature, depending on its perceptual region and semantic pattern, contributes differently to the frequency components. This enables IMAM to precisely enhance or suppress frequency components, optimizing their contribution to the relevant semantic contexts. Additionally, considering that IMAM is designed to learn the frequency response within a local $K \times K$ region, its output channels are set to $K^2$. However, since $K$ is typically 2 in the pooling layer, this small number of channels may limit the learning capacity for representation. Inspired by the inverted residual structure in MobileNetV2 [29], We set the intermediate channel size of multiform depthwise convolutions to $r \times K^2$, where $r$ is a hyperparameter, in order to enhance the modulation of local frequency responses. This process can be written as follows:

$$X_{f1} = \mathcal{G}(BN(Conv_{1\times1}^{C\to r*K^2}(X_r))) \tag{9}$$

$$\begin{aligned} X_{f1}, X_{f2}, X_{f3}, X_{f4} &= Split(X_r, 4) \\ &= X_{:g,:}, X_{g:2g,:}, X_{2g:3g,:}, X_{3g:,:} \end{aligned} \tag{10}$$

$$X'_{f1} = \mathcal{R}(DWConv_{k_s\times k_s}^{g\to g}(X_{f1})) \tag{11}$$

$$X'_{f2} = \mathcal{R}(DWConv_{k_s\times k_s,d=2}^{g\to g}(X_{f2})) \tag{12}$$

$$X'_{f3} = \mathcal{R}(DWConv_{1\times(2k_s-1)}^{g\to g}(X_{f3})) \tag{13}$$

$$X'_{f4} = \mathcal{R}(DWConv_{(2k_s-1)\times1}^{g\to g}(X_{f4})) \tag{14}$$

$$X_f = Concat(X'_{f1}, X'_{f2}, X'_{f3}, X'_{f4}) \tag{15}$$

$$X'_f = \mathcal{G}(GN_4(X_f)) \tag{16}$$

$$Attn = \sigma(Conv^{r*K^2 \to K^2}_{1\times 1}(X'_f)) \tag{17}$$

where $\sigma(\cdot)$ represents the Sigmoid normalization, $\mathcal{R}$ denotes the ReLU activation function, and $GN_4$ indicates Group Normalization with 4 groups. $_{d=2}$ representing a dilation rate of 2 for the dilated convolution [66], with the default value being 1 if not specified. The aforementioned $K_s \times K_s$ and $r$ are hyperparameters of IMAM. Their value selection and analysis will be thoroughly discussed in the ablation studies in Tables 8 and 9. This dynamic frequency weighting generation method, guided by spatial domain contextual information, enhances the network's fine-grained responsiveness to different frequency components.

*Local aggregation strategies across different domains*

After completing the dimension-preserving 2D DCT transformation from spatial to frequency domain, IMAM generates dynamic weights for different frequency components by leveraging the rich contextual spatial awareness. Subsequently, local feature aggregation is applied to fully exploit the varying energy intensities across frequency components. Notably, the 2D IDCT is also dimension-preserving, meaning that a $K \times K$ frequency domain window remains $K \times K$ when transformed back into the spatial domain. Based on this property, we propose two local aggregation strategies, one in the frequency domain and the other in the spatial domain. Specifically, for frequency domain aggregation, as shown in Fig. 2, we first compute the weighted sum of frequency components in each frequency window, and then reconstruct it into a new frequency representation $F_{2^{(i,j)}_{(u,v)}}$ as follows:

$$F_{1^{(i,j)}_{(u,v)}} = Attn^{(i,j)} \times F^{(i,j)}_{(u,v)} \tag{18}$$

$$F_{2^{(i,j)}_{(u,v)}}(x,y) = \begin{cases} sum(F_{1^{(i,j)}_{(u,v)}}), & if\,(u,v) = (0,0) \\ 0, & otherwise \end{cases} \tag{19}$$

Here, $sum(\cdot)$ represents the weighted summation of frequency components within the $K \times K$ frequency domain window, and $F_{2^{(i,j)}_{(u,v)}}(x,y)$ denotes the coefficient at position $(x,y)$ in the DCT coefficient matrix indexed by $(i,j)$, functioning as a low-frequency filter, where $x, y \in \{0, 1, \ldots, K-1\}$. In Eq. (19), the weighted sum of $F_{1^{(i,j)}_{(u,v)}}$ is reassigned to the lowest frequency component, while all other components are set to zero. Next, we apply the IDCT to $F_{2^{(i,j)}_{(u,v)}}$, and take the average as the final pooling output for the local window. The process can be written as follows:

$$X^{(i,j)}_a = \mathcal{P}(\sum_{u=0}^{K-1}\sum_{v=0}^{K-1} F_{2^{(i,j)}_{(u,v)}} C^{(x,y)}_{(u,v)}) \tag{20}$$

For spatial domain aggregation, the frequency components $F_{1^{(i,j)}_{(u,v)}}$ modulated by IMAM are first inverse transformed back to the spatial domain, followed by averaging to obtain the aggregated result.

$$X^{(i,j)}_a = \mathcal{P}(\sum_{u=0}^{K-1}\sum_{v=0}^{K-1} F_{1^{(i,j)}_{(u,v)}} C^{(x,y)}_{(u,v)}) \tag{21}$$

Finally, the output of the S3 module is expressed as

$$Output = Concat(X_l, X_a) \tag{22}$$

## 4. Experiments

### 4.1. Experiments settings

*Datasets*

We validate the effectiveness of our method across four visual tasks. For the image classification, we select the widely used ImageNet-1K [1] dataset. In the object detection, we employ several challenging detection datasets, including MSCOCO [3], Pascal VOC [38], VisDrone [37], ExDark [36], and HazyDet [39]. The specific details of the dataset are presented in Table 2.

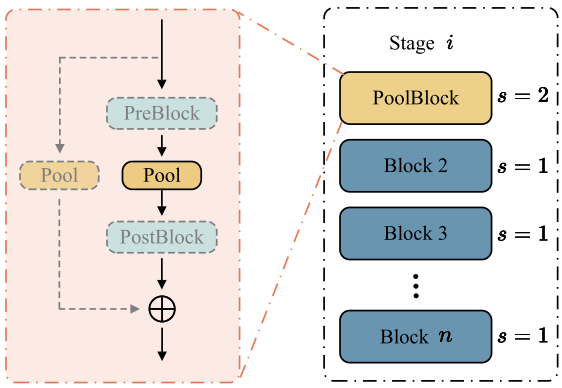



**Fig. 4.** Illustration of the generalized hierarchical structure with pooling method. The structures depicted by gray dashed lines are conditionally included, indicating the optional existence of residual branches. When these branches are present, pooling layers are likewise incorporated within them. PreBlock and PostBlock indicate optional operations, such as convolution for channel scaling transformations.

*Metrics*

We use Top-1 and Top-5 metrics to measure image classification, Average Precision (AP) to evaluate object detection, and report Parameter Count (Params) and Floating Point Operations Per Second (FLOPs).

*Implement details*

To validate that our method contributes to enhancing the representational capacity of the model, we select four mainstream backbone networks based on CNN architecture and conduct a quantitative comparison of different pooling methods based on them. We maintain the same configurations for ResNet [2], MobileNetV2 [29], RepVGG [30], and MobileOne [31] as presented in their original papers [2,29–31]. We uniformly replace all pooling layers with a stride greater than 1, except for the first pooling layer in the stem layer. It is worth noting that, to ensure fairness, we apply the replacement method shown in Fig. 4 for all pooling layers. For network structures that include residual branches with a stride greater than 1, such as ResNet [2], we also apply the proposed pooling method for replacement.

To evaluate the advantages of our pooling method in dense detection tasks, we explore its integration into various detectors, including the two-stage detectors Faster R-CNN [5] and Cascade R-CNN [4], the one-stage detector TOOD [67], the latest real-time YOLO series detectors [8,32–35], and the improved DINO [6] and DDQ [7] based on DETR [68]. For the MSCOCO [3] dataset, we follow their respective hyperparameter settings [4–7,67] and fine-tune the pre-trained models for 12 epochs (1x schedule) on object detection. For the real-time YOLO series detectors, we select representative datasets focusing on low-light [36] and small object surveillance [37] scenarios. We train the models from scratch for 300 epochs, using a batch size of 16 and a learning rate of 0.01, with the remaining hyperparameters and data augmentation settings following the default configurations of the YOLO series [8,32–35].

We train the above models using 8 NVIDIA 2080 Ti GPUs with the MMPretrain [69], MMDetection [70], and Ultralytics [8] toolboxs.

### 4.2. Main results

In this section, we first evaluate the effectiveness of our method in feature extraction using standard image classification benchmarks. Next, We report the detection results fine-tuned on the MSCOCO [3] dataset using advanced detectors. Finally, we integrate the proposed D3 module into the mainstream YOLO series real-time detectors to further demonstrate its generalization ability in complex scenarios, such as low-light and small-object surveillance.

**Table 1**
Comparison of our proposed D3 with other state-of-the-art pooling across multiple benchmark models at a 224 × 224 resolution on the ImageNet-1K validation set [1].

| Backbone | Pooling | Params (M) | FLOPs (G) | Top-1 (%) | Top-5 (%) |
|---|---|---|---|---|---|
| ResNet-50 | Original (MaxPool + Strided Conv) | 25.56 | 4.11 | 76.44 | 93.27 |
| | StochasticPool | 22.46 | 3.77 | 76.27 | 93.01 |
| | $L_p$ | 22.46 | 3.76 | 76.38 | 93.20 |
| | MixedPool | 22.46 | 3.76 | 76.40 | 93.28 |
| | WaveletPool | 22.46 | 3.77 | 76.44 | 93.31 |
| | GaussianPool | 25.07 | 3.77 | 76.61 | 93.27 |
| | S3Pool | 22.46 | 4.24 | 76.85 | 93.36 |
| | ConditionalPool | 22.46 | 3.76 | 76.89 | 93.32 |
| | SoftPool | 22.46 | 3.76 | 77.06 | 93.38 |
| | LDW-Pool | 26.72 | 4.55 | 77.10 | 93.31 |
| | LIP | 23.86 | 5.34 | 77.25 | 93.50 |
| | **D3 (Ours)** | 24.66 | 4.70 | **77.72** | **93.74** |
| ResNet-101 | Original (MaxPool + Strided Conv) | 44.55 | 7.83 | 77.76 | 93.81 |
| | ConditionalPool | 41.45 | 7.49 | 77.81 | 93.83 |
| | SoftPool | 41.45 | 7.49 | 78.01 | 93.92 |
| | LIP | 42.86 | 9.06 | 78.13 | 93.99 |
| | **D3 (Ours)** | 43.65 | 8.42 | **78.58** | **94.17** |
| MobileNetV2-1.0 | Original (Strided Conv) | 3.51 | 0.31 | 71.88 | 90.28 |
| | GaussianPool | 4.01 | 0.31 | 72.15 | 90.35 |
| | LIP | 3.51 | 0.34 | 72.39 | 90.49 |
| | ConditionalPool | 3.49 | 0.31 | 72.40 | 90.55 |
| | SoftPool | 3.49 | 0.31 | 72.44 | 90.60 |
| | SAPool | 3.86 | 0.96 | 72.74 | 91.03 |
| | **D3 (Ours)** | 3.94 | 0.64 | **73.20** | **91.30** |
| RepVGG-A1 | Original (Strided Conv) | 14.09 | 2.64 | 74.16 | 91.63 |
| | ConditionalPool | 12.46 | 2.51 | 74.66 | 91.98 |
| | SoftPool | 12.46 | 2.51 | 74.98 | 92.14 |
| | LIP | 15.82 | 4.38 | 75.49 | 92.55 |
| | **D3 (Ours)** | 14.27 | 3.03 | **76.11** | **92.98** |
| MobileOne-S0 | Original (Strided Conv) | 2.08 | 0.27 | 71.34 | 89.88 |
| | **D3 (Ours)** | 2.19 | 0.36 | **73.08** | **91.01** |
| MobileOne-S3 | Original (Strided Conv) | 10.08 | 1.89 | 78.01 | 93.78 |
| | **D3 (Ours)** | 10.84 | 2.23 | **78.89** | **94.21** |

**Table 2**
Datasets used in our work.

| Name | train | val | test | category |
|---|---|---|---|---|
| ImageNet-1K | 1 281 167 | 50 000 | 100 000 | 1000 |
| MSCOCO 2017 | 118 287 | 5000 | 40 670 | 80 |
| VOC 07+12 | 16 551 | 4952 | – | 20 |
| VisDrone2019-DET | 6471 | 548 | 1610 | 10 |
| ExDark | 4712 | 1178 | 1473 | 12 |
| HazyDet | 8000 | 1000 | 2000 | 3 |

*Standard image classification*

As shown in Table 1, we compare the proposed D3 with several existing pooling methods, including Max Pooling (MaxPool) [15], Strided Convolution (Strided Conv), Stochastic Pooling (StochasticPool) [21], Learned-norm Pooling ($L_p$) [71], Mixed Pooling (MixedPool) [22], Gaussian-based Pooling (GaussianPool) [72], S3Pool [25], Local Importance-based Pooling (LIP) [16], SoftPool [17], Self-attentive Pooling (SAPool) [18], and Conditional Pooling (ConditionalPool) [27]. On the widely-used ImageNet-1K classification dataset, replacing traditional pooling layers (stride > 1) with our D3 method improved the Top-1 accuracy by 1.28%, 0.82%, 1.32%, and 1.95% on ResNet-50, ResNet-101, MobileNetV2-1.0, and RepVGG-A1, respectively. On the ResNet-50 backbone, D3 outperformed the probabilistically importance-based StochasticPool and S3Pool methods by 1.45% and 0.87%, respectively. It also outperformed wavelet-based pooling methods, including WaveletPool [73] and LDW-Pool [74]. When applied to MobileNetV2-1.0 and RepVGG-A1, D3 demonstrates significant advantages over several state-of-the-art methods, particularly outperforming parameter-free poolings such as SoftPool (+0.76% and +1.13%) and ConditionalPool (+0.80% and +1.45%). Our results further underscore the necessity of incorporating convolutions within the pooling process to enhance feature extraction capabilities. Additionally, D3 demonstrates strong performance on MobileOne-S0 with a 1.74% improvement and even achieves a 0.88% gain over Strided Convolution on the larger MobileOne-S3.

*Pretrained-based object detection*

To validate the effectiveness of D3 in detail-intensive detection and its adaptability across different detectors, we select two two-stage detectors and two recently improved DETR-based detectors, using the ResNet-50 backbone pretrained on ImageNet-1K for fine-tuning. As shown in Table 3, our D3 improves the performance of Cascade R-CNN based on ResNet-50 and ResNet-101 by 1.4% and 1.0%, respectively, over the combination of MaxPool and Strided Conv. In detectors based on DINO-4scale [6] and DDQ-4scale [7], D3 also achieves notable improvements of 0.5% and 0.7%, respectively. This further demonstrates the effectiveness of our method in object detection.

*Scratch-based real-time object detection*

Real-time object detectors are crucial for security surveillance [75] and autonomous driving [76] applications. We further evaluate the performance of the proposed D3 on YOLOv7 ∼ YOLOv10 [32–35], as well as lastest YOLO11 [8], within low-light, small-object surveillance scenarios, as shown in Table 4. Replacing all pooling layers with the proposed D3 (including PAFPN [77] neck layers), except for the first layer with stride > 1, improves detection accuracy by 1.0% to 2.6% on the Pascal VOC benchmark, 0.5% to 1.4% on the VisDrone dataset, and 0.7% to 4.1% in ExDark low-light scenarios. Furthermore, compared to the original structures across multiple models, the proposed method reduces the parameter count with only a minor increase in FLOPs, effectively demonstrating its lightweight efficacy.

**Table 3**
Performance comparison of various pooling methods for object detection on MSCOCO val2017 [3], using detectors like Faster R-CNN [5], Cascade R-CNN [4], TOOD [67], DINO [6], and DDQ [7]. All models were fine-tuned under the 1× schedule.

| Detector | Pooling | Backbone | Type | Params (M) | $AP$ (%) | $AP_{50}$ (%) | $AP_{75}$ (%) | $AP_S$ (%) | $AP_M$ (%) | $AP_L$ (%) |
|---|---|---|---|---|---|---|---|---|---|---|
| Faster R-CNN | MaxPool + Strided Conv | ResNet-50 | Two-stage | 41.8 | 37.4 | 58.3 | 40.8 | 21.6 | 41.0 | 48.1 |
| | ConditionalPool | ResNet-50 | Two-stage | 38.7 | 37.5 | 58.5 | 40.8 | 21.8 | 41.2 | 48.4 |
| | SoftPool | ResNet-50 | Two-stage | 38.7 | 37.7 | 58.8 | 41.1 | 21.7 | 41.5 | 48.4 |
| | LIP | ResNet-50 | Two-stage | 40.1 | 38.2 | 59.0 | 41.4 | 22.6 | 42.1 | 48.9 |
| | **D3 (Ours)** | ResNet-50 | Two-stage | 40.9 | **38.9** | **60.1** | **42.2** | **23.7** | **42.9** | **49.2** |
| Cascade R-CNN | MaxPool + Strided Conv | ResNet-50 | Two-stage | 69.4 | 40.1 | 58.6 | 43.4 | 22.6 | 43.5 | 51.4 |
| | ConditionalPool | ResNet-50 | Two-stage | 66.3 | 40.4 | 58.7 | 43.7 | 22.6 | 43.9 | 52.0 |
| | SoftPool | ResNet-50 | Two-stage | 66.3 | 40.5 | 59.1 | 43.8 | 22.8 | 44.0 | 51.9 |
| | LIP | ResNet-50 | Two-stage | 67.7 | 40.9 | 59.7 | 44.1 | 23.3 | 44.7 | 52.2 |
| | **D3 (Ours)** | ResNet-50 | Two-stage | 68.5 | **41.5** | **60.4** | **45.3** | **24.6** | **45.0** | **53.0** |
| | MaxPool + Strided Conv | ResNet-101 | Two-stage | 88.4 | 42.3 | 60.8 | 46.3 | 24.8 | 46.1 | 55.3 |
| | ConditionalPool | ResNet-101 | Two-stage | 85.3 | 42.1 | 60.8 | 46.2 | 24.7 | 46.5 | 55.6 |
| | SoftPool | ResNet-101 | Two-stage | 85.3 | 42.6 | 61.1 | 46.7 | 25.1 | 46.5 | 55.8 |
| | LIP | ResNet-101 | Two-stage | 86.7 | 43.0 | 61.7 | 47.0 | 25.2 | 46.6 | 56.0 |
| | **D3 (Ours)** | ResNet-101 | Two-stage | 87.5 | **43.3** | **62.1** | **47.5** | **25.2** | **47.0** | **56.3** |
| TOOD | MaxPool + Strided Conv | ResNet-50 | One-stage | 32.2 | 42.4 | 59.5 | 46.1 | 25.1 | 45.5 | 55.5 |
| | ConditionalPool | ResNet-50 | One-stage | 29.1 | 42.4 | 59.3 | 46.3 | 24.9 | 45.5 | 56.0 |
| | SoftPool | ResNet-50 | One-stage | 29.1 | 42.8 | 59.9 | 47.0 | 25.3 | 45.8 | 56.1 |
| | LIP | ResNet-50 | One-stage | 30.5 | 42.9 | 60.3 | **47.2** | **25.5** | 46.3 | 56.1 |
| | **D3 (Ours)** | ResNet-50 | One-stage | 31.3 | **43.3** | **60.9** | 47.1 | 25.3 | **47.0** | **56.3** |
| DINO-4scale | MaxPool + Strided Conv | ResNet-50 | End-to-end | 47.7 | 48.8 | 66.1 | 53.3 | 31.9 | 51.9 | 62.6 |
| | ConditionalPool | ResNet-50 | End-to-end | 44.6 | 48.6 | 65.9 | 53.3 | 31.5 | 52.0 | 62.5 |
| | SoftPool | ResNet-50 | End-to-end | 44.6 | 48.8 | 66.2 | 53.5 | 31.8 | 52.2 | 62.9 |
| | LIP | ResNet-50 | End-to-end | 46.0 | 49.0 | 66.2 | 53.7 | 31.9 | 52.4 | 63.0 |
| | **D3 (Ours)** | ResNet-50 | End-to-end | 46.8 | **49.3** | **67.0** | **53.7** | **32.2** | **52.8** | **63.4** |
| DDQ-4scale | MaxPool + Strided Conv | ResNet-50 | End-to-end | 47.5 | 51.1 | 68.2 | 56.5 | 34.2 | 54.5 | 65.0 |
| | ConditionalPool | ResNet-50 | End-to-end | 44.4 | 51.0 | 68.3 | 56.6 | 34.0 | 55.0 | 65.1 |
| | SoftPool | ResNet-50 | End-to-end | 44.4 | 51.3 | 68.3 | 56.7 | 34.4 | 54.8 | 65.4 |
| | LIP | ResNet-50 | End-to-end | 46.8 | 51.5 | 68.6 | 56.8 | 34.5 | 55.0 | 65.7 |
| | **D3 (Ours)** | ResNet-50 | End-to-end | 46.6 | **51.8** | **69.0** | **57.1** | **34.7** | **55.4** | **66.0** |

**Table 4**
Comparison of different pooling methods in complex scenes [36–38], conducted using the latest real-time detectors [8,32–35]. All models were trained from scratch for 300 epochs.

| Detector | Pooling | VOC 07+12 [38] | | | VisDrone2019 [37] | | | ExDark [36] | | | Latency |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Params (M) | FLOPs (G) | $AP$ (%) | Params (M) | FLOPs (G) | $AP$ (%) | Params (M) | FLOPs (G) | $AP$ (%) | (ms) |
| YOLOv7-tiny | Original (Strided Conv) | 6.1 | 13.3 | 54.6 | 6.0 | 13.3 | 18.5 | 6.0 | 13.3 | 38.2 | 4.4 |
| | SoftPool | 5.7 | 12.6 | 55.8 | 5.7 | 12.6 | 19.1 | 5.7 | 12.6 | 40.8 | 4.8 |
| | LIP | 7.3 | 29.7 | 56.2 | 7.3 | 29.7 | 19.3 | 7.3 | 29.7 | 41.4 | 6.2 |
| | SAPool | 7.0 | 16.6 | 56.4 | 7.0 | 16.6 | 19.3 | 7.0 | 16.6 | 41.7 | 12.5 |
| | **D3 (Ours)** | 5.9 | 15.1 | **57.2** | 5.9 | 15.0 | **19.9** | 5.9 | 15.0 | **42.3** | 6.8 |
| YOLOv8-n | Original (Strided Conv) | 3.0 | 8.2 | 59.4 | 3.0 | 8.2 | 19.5 | 3.0 | 8.2 | 43.1 | 3.9 |
| | SoftPool | 2.5 | 7.5 | 60.1 | 2.5 | 7.5 | 19.6 | 2.5 | 7.5 | 43.0 | 4.2 |
| | LIP | 3.5 | 16.0 | 60.6 | 3.4 | 16.0 | 19.8 | 3.5 | 16.0 | 43.5 | 5.4 |
| | SAPool | 3.6 | 9.5 | 60.8 | 3.6 | 9.4 | 19.9 | 3.6 | 9.4 | 43.1 | 11.8 |
| | **D3 (Ours)** | 2.6 | 8.9 | **61.3** | 2.6 | 8.8 | **20.0** | 2.6 | 8.9 | **44.1** | 5.9 |
| YOLOv9-t | Original (AConv) | 2.0 | 7.9 | 61.4 | 2.0 | 7.9 | 19.7 | 2.0 | 7.9 | 43.3 | 9.8 |
| | SoftPool | 1.8 | 7.4 | 61.7 | 1.8 | 7.4 | 19.7 | 1.8 | 7.4 | 43.4 | 10.0 |
| | SAPool | 2.3 | 8.8 | 61.8 | 2.3 | 8.8 | 20.0 | 2.3 | 8.8 | 43.4 | 17.2 |
| | LIP | 2.2 | 13.5 | 62.2 | 2.2 | 13.5 | 20.4 | 2.2 | 13.5 | 43.5 | 11.5 |
| | **D3 (Ours)** | 1.8 | 8.5 | **62.8** | 1.8 | 8.4 | **21.0** | 1.8 | 8.4 | **44.1** | 11.9 |
| YOLOv10-n | Original (SCDown) | 2.3 | 6.7 | 60.6 | 2.3 | 6.7 | 19.3 | 2.3 | 6.7 | 42.6 | 5.7 |
| | GaussianPool | 2.4 | 6.5 | 58.9 | 2.4 | 6.5 | 19.0 | 2.4 | 6.5 | 42.1 | 7.2 |
| | ConditionalPool | 2.2 | 6.3 | 60.5 | 2.2 | 6.3 | 19.3 | 2.2 | 6.3 | 42.7 | 6.8 |
| | SoftPool | 2.2 | 6.3 | 61.0 | 2.2 | 6.3 | 19.5 | 2.2 | 6.3 | 42.9 | 6.3 |
| | **D3 (Ours)** | 2.3 | 7.6 | **62.0** | 2.3 | 7.6 | **20.1** | 2.3 | 7.6 | **43.3** | 7.5 |
| YOLO11-n | Original (Strided Conv) | 2.6 | 6.5 | 60.5 | 2.6 | 6.5 | 19.3 | 2.6 | 6.5 | 42.9 | 4.9 |
| | GaussianPool | 2.1 | 5.1 | 60.0 | 2.1 | 5.1 | 19.1 | 2.1 | 5.1 | 42.2 | 6.6 |
| | ConditionalPool | 2.0 | 5.0 | 60.3 | 2.0 | 5.0 | 19.2 | 2.0 | 5.0 | 42.6 | 6.2 |
| | SoftPool | 2.0 | 5.0 | 60.6 | 2.0 | 5.0 | 19.2 | 2.0 | 5.0 | 43.3 | 5.2 |
| | **D3 (Ours)** | 2.1 | 6.4 | **61.5** | 2.1 | 6.4 | **19.8** | 2.1 | 6.4 | **44.1** | 7.1 |

### 4.3. Ablation study

To gain a deeper understanding of how D3 impacts model performance across multiple aspects, we conduct a comprehensive ablation study on each component of D3 using the ResNet-50 [2] architecture.

*Pooling layer replacement across depths*

From the Top-1 and Top-5 results in Table 6, along with the changes in parameters and FLOPs, we observe that as the depth of D3 pooling layer replacements increases, accuracy improves correspondingly, parameters decrease, and FLOPs slightly increase. The two most

**Table 5**
Comparison of the Params, FLOPS, and AP of different single-stage, two-stage, and end-to-end detectors on the HazyDet [39] dataset.

| Detector | Backobone | Pooling | Params (M) | FLOPs (G) | $AP^{test}$ (%) |
|---|---|---|---|---|---|
| YOLOX-s | CSPDarkNet | Strided Conv | 9.0 | 26.8 | 42.3 |
| Centernet | ResNet-50 | MaxPool + Strided Conv | 32.1 | 191 | 47.2 |
| TOOD | ResNet-50 | MaxPool + Strided Conv | 32.0 | 193 | 51.4 |
| Faster RCNN | ResNet-50 | MaxPool + Strided Conv | 41.4 | 201.7 | 48.7 |
| Cascade RCNN | ResNet-50 | MaxPool + Strided Conv | 69.2 | 230.4 | 51.6 |
| Conditional DETR | ResNet-50 | MaxPool + Strided Conv | 43.6 | 94.2 | 30.5 |
| Deformable DETR | ResNet-50 | MaxPool + Strided Conv | 43.7 | 192.5 | 51.9 |
| YOLO11-n | CSPDarkNet | Strided Conv | 2.6 | 6.5 | 47.5 |
| YOLO11-n | CSPDarkNet | **D3 (Ours)** | 2.1 | 6.4 | 49.2 |
| YOLO11-l | CSPDarkNet | Strided Conv | 25.3 | 87.3 | 56.2 |
| YOLO11-l | CSPDarkNet | **D3 (Ours)** | 18.3 | 79.3 | **57.1** |
| YOLO11-x | CSPDarkNet | Strided Conv | 56.8 | 195.5 | 58.2 |
| YOLO11-x | CSPDarkNet | **D3 (Ours)** | 41.0 | 176.4 | **58.8** |

**Table 6**
Substitution of pooling layers in ResNet-50 [2], with experiments conducted on the ImageNet-1K validation set [1]. Different pooling layers selected for substitution are marked with ✓.

| Layer | Pooling layers substitution with D3 | | | | | | |
|---|---|---|---|---|---|---|---|
| | N | I | II | III | IV | V | VI |
| $pool_{stem}$ | | ✓ | | | | | ✓ |
| $stage_1$ | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| $stage_2$ | | | | ✓ | ✓ | ✓ | ✓ |
| $stage_3$ | | | | | ✓ | ✓ | ✓ |
| $stage_4$ | | | | | | ✓ | ✓ |
| Top-1 (%) | 76.44 | 76.79 | 76.49 | 76.59 | 76.73 | 77.36 | **77.72** |
| Top-5 (%) | 93.27 | 93.29 | 93.22 | 93.17 | 93.25 | 93.54 | **93.74** |
| Params (M) | 25.56 | 25.56 | 25.56 | 25.52 | 25.35 | 24.61 | 24.66 |
| FLOPs (G) | 4.10 | 4.18 | 4.11 | 4.29 | 4.46 | 4.62 | 4.70 |

**Table 7**
Ablation studies on varying frequency window sizes using ResNet-50 [2] on the ImageNet-1K validation set [1].

| Frequency window size | 2 | 3 | 4 |
|---|---|---|---|
| Top-1 (%) | **77.72** | 77.25 | 77.27 |
| Top-5 (%) | **93.74** | 93.52 | 93.52 |
| Params (M) | 24.66 | 24.78 | 24.97 |
| FLOPs (G) | 4.70 | 4.79 | 4.98 |

**Table 8**
Ablation studies on different hidden factor $r$ using ResNet-50 [2] on the ImageNet-1K validation set [1].

| Hidden factor $r$ | 4 | 8 | 16 | 32 |
|---|---|---|---|---|
| Top-1 (%) | 77.38 | 77.56 | **77.72** | 77.50 |
| Top-5 (%) | 93.64 | 93.70 | **93.74** | 93.69 |
| Params (M) | 24.59 | 24.61 | 24.66 | 24.76 |
| FLOPs (G) | 4.68 | 4.69 | 4.70 | 4.73 |

**Table 9**
Comparison of performance based on kernel sizes in different stages, conducted on the ImageNet-1K validation set [1].

| Stages kernel size | Params (M) | FLOPs (G) | Top-1 (%) | Top-5 (%) |
|---|---|---|---|---|
| (3, 3, 3, 3) | 24.66 | 4.70 | **77.72** | **93.74** |
| (3, 3, 5, 7) | 24.69 | 4.71 | 77.41 | 93.66 |
| (7, 7, 7, 7) | 24.70 | 4.73 | 77.39 | 93.61 |
| (9, 9, 9, 9) | 24.73 | 4.76 | 77.33 | 93.61 |

significant gains, 0.35% and 0.57%, are achieved by replacing max pooling in the stem and adopting strided convolution in the semantically deepest fourth stage, respectively. The primary reason lies in the enhanced information richness from multi-domain learnability and the greater benefit of receptive field enlargement in deeper layers compared to shallow ones. Compared to single-domain, non-learnable pooling methods, cooperative learning between frequency and spatial domains effectively boosts the model's representational capacity. Additionally, deeper layers provide higher levels of semantic abstraction through the IMAM module's multiform convolution, receptive field expansion not only improves the model's ability to capture broader features but also enhances its comprehension of high-level semantic information.

*Ablation on frequency window size*

Different sizes of frequency windows yield varying levels of frequency response. As shown in Table 7, we conducted ablation studies on common pooling window sizes, and the results indicate that a frequency window size of 2 significantly outperforms other sizes in terms of accuracy, parameter count, and FLOPs. This advantage can be partly attributed to the fact that odd window sizes involve padding with irrelevant pixels, which may impact effectiveness. Additionally, while a stride of 2 with a non-overlapping window size of 2 might suggest limited coverage, the IMAM module's integration of multi-scale overlapping information effectively mitigates the reduced generalization to unseen data typically associated with non-overlapping downsampling [25]. The improvement in generalization is further validated by the complex scene detection results shown in Table 4.

*Selection of expansion hidden factor*

The number of local frequency components is constrained by the size of the frequency window. To address this, we introduced an expansion factor to enhance the model's ability to learn the weights of local frequency components. As shown in Table 8, accuracy consistently improves with an increase in the expansion factor, reaching its peak performance at an expansion factor of $r = 16$. However, further increases in the expansion factor lead to a decline in accuracy. This is because a small expansion factor may hinder effective feature extraction, whereas an overly large expansion factor introduces redundant feature channels, which can interfere with learning critical features and increase the risk of overfitting.

*Kernel size for neighboring spatial area*

The size of the neighboring spatial kernel largely determines the receptive range of D3. A kernel that is too small may fail to effectively leverage contextual information, limiting the ability to distinguish between details and noise; conversely, a kernel that is too large may introduce irrelevant background noise into the local aggregation. As shown in Table 9, we conducted ablation studies on the convolutional kernel sizes across four stages. Interestingly, in the pooling process, a larger convolutional kernel size is not necessarily optimal. For the multiform convolution in IMAM, kernel sizes are selected from $\{K, 2K_s-1\}$. Results indicate that both shallow and deep layers achieve optimal performance and minimal computational cost with appropriately chosen kernel sizes.

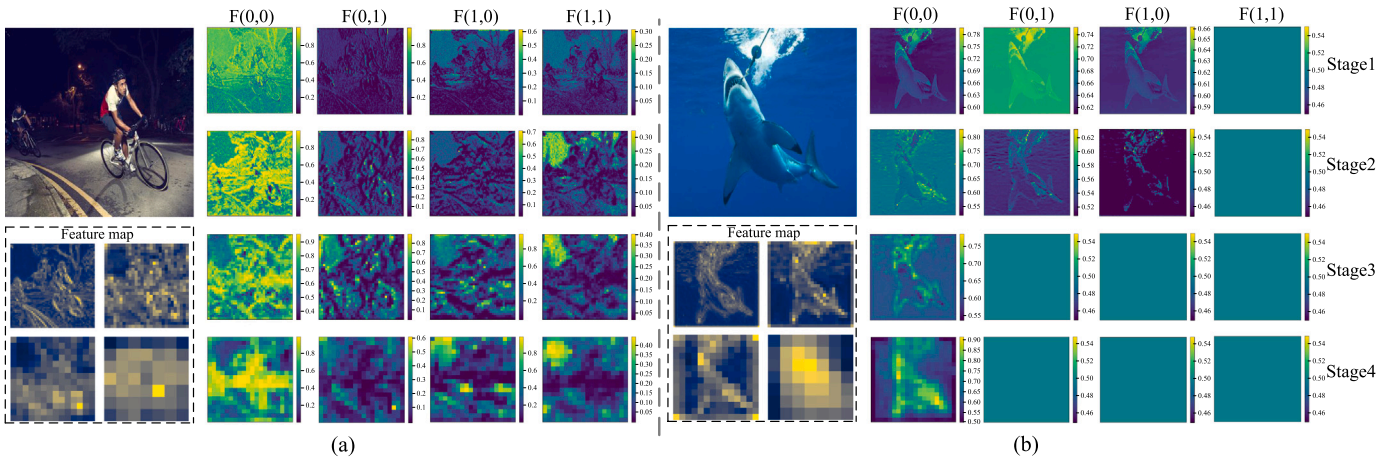



**Fig. 5.** Heatmap of frequency component weight importance at different pooling layers and locations using the proposed D3 method, with the feature maps for each stage shown in the lower left corner. The importance is measured using the Sigmoid function, with brighter colors indicating relatively higher importance. (a) generated from YOLO11 [8] and the ExDark [36] dataset. (b) generated from ResNet-50 [2] and the ImageNet-1K dataset [1].

**Table 10**
Comparison of the four aggregation strategies in different domains on the ImageNet-1K validation set [1].

| Domain aggregation method | Params (M) | FLOPs (G) | Top-1 (%) | Top-5 (%) |
|---|---|---|---|---|
| Mean w frequency domain | 24.57 | 4.67 | 76.93 | 93.35 |
| Sum w frequency domain | 24.57 | 4.67 | 76.31 | 93.15 |
| Learned w frequency domain | 24.66 | 4.70 | **77.72** | **93.74** |
| Learned w sptial domain | 24.66 | 4.70 | 77.41 | 93.66 |

**Table 11**
Performance of different structures in the D3 module on the ImageNet-1K validation set [1], including ablation studies of proposed SEM and IMAM.

| Ablations | Params (M) | FLOPs (G) | Top-1 (%) | Top-5 (%) |
|---|---|---|---|---|
| **D3** (baseline) | 24.66 | 4.70 | **77.72** | **93.74** |
| sigmoid → softmax | 24.66 | 4.70 | 77.28 | 93.55 |
| w/o SEM | 22.90 | 3.84 | 76.99 | 93.42 |
| w/o IMAM | 24.57 | 4.66 | 76.82 | 93.23 |

*Domain aggregation strategy and single-domain learning*

We compared four pooling aggregation methods: two static frequency domain aggregation methods (mean and sum) and two dynamically weighted aggregation methods based on learning across different domains. As shown in Table 10, the frequency-domain-based dynamic aggregation method significantly outperforms static aggregation methods, with an accuracy improvement of 0.79% to 1.39%, and it also surpasses the spatial aggregation method by 0.31%. This result suggests that feature aggregation in the frequency domain better preserves subtle variations in frequency components, while effectively capturing their excitatory and inhibitory properties. This is because frequency component coefficients and dynamic frequency weights are more naturally aligned with the frequency domain, allowing for a more accurate representation of their interactions and variations. We also conducted ablation studies on the SEM and IMAM modules to evaluate their feature extraction capabilities in the frequency and spatial domains. As shown in Table 11, pooling solely in the spatial domain results in a 0.55% increase over traditional strided convolution. Pooling solely in the frequency domain, without spatial information enhancement, also outperforms strided convolution by 0.38%. When both domains are combined, the Top-1 accuracy reaches a peak of 77.72%. Additionally, replacing the attention normalization function from sigmoid to softmax significantly decreases accuracy, further indicating the synergistic effect of multiple frequency components rather than the influence of a single one.

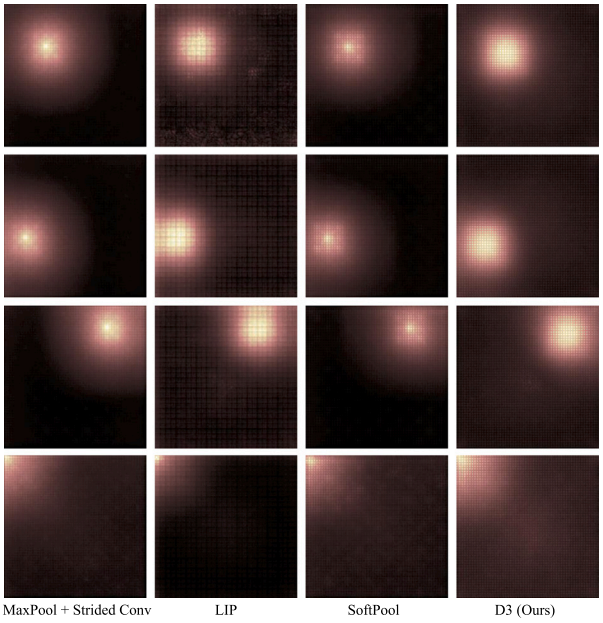



**Fig. 6.** Visual comparison of effective receptive fields (ERF) among various pooling methods. The brightness represents the contribution strength of each region, while the coverage reflects the size of the ERF. The analysis is based on the ImageNet-1K validation set [1] and the ResNet-50 [2] model, with four positions randomly selected for evaluation.

## 5. Analysis and discussion

We utilize ResNet-50 [2] and YOLO11 [8] as baselines and conduct an in-depth analysis of the proposed D3.

### 5.1. Analysis of frequency component weights

As shown in Fig. 5, we visualize the frequency component weights generated by the IMAM module across four stages. In Fig. 5(a), the input image size is 640 × 640, with resolutions at the four stages being 160×160, 80 × 80, 40×40, and 20 × 20. In contrast, in Fig. 5(b), the input image size is 224 × 224, and the resolutions at the four stages are 56×56, 28 × 28, 14×14, and 7 × 7. For each stage, we display the weight heatmaps from low-frequency components to high-frequency components from left to right. We observe that at different stages,

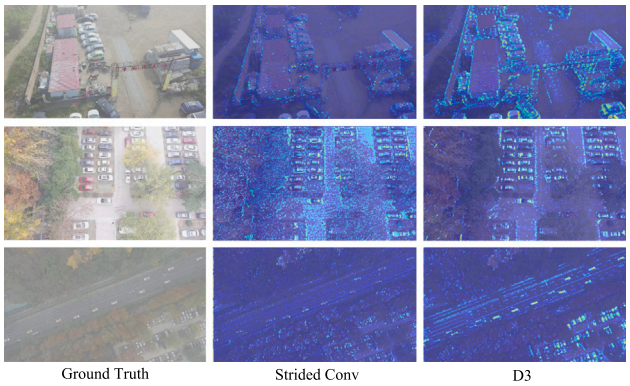



**Fig. 7.** Heatmap visualizations on the HazyDet test set [39] based on YOLO11 [8]. The heatmaps for the shallow layers are generated using the Grad-CAM [78] tool, with brighter colors indicating higher attention values.

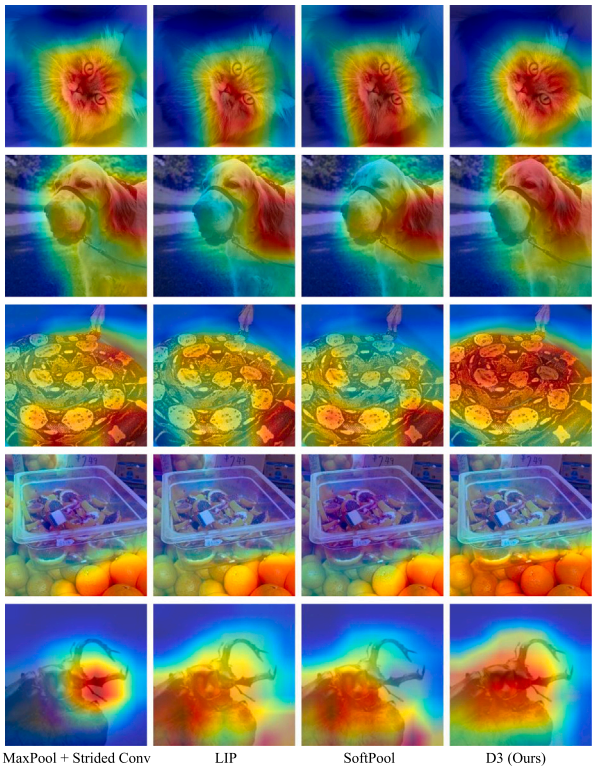



**Fig. 8.** Visualized results on the ImageNet-1K validation set [1]. The heatmaps are generated for layer 4.2 using the Grad-CAM [78] tool. Brighter colors indicate higher attention values.

the data distribution of weights produced by IMAM varies not only across different frequency components but also somewhat resembles the feature maps displayed in the lower left corner, facilitating the distinction between the main objects and background regions. This is primarily due to our use of contextual information from the neighborhood of each pixel, which modulates different frequency responses through spatial information. Specifically, $F(0,0)$ represents the weight of the lowest frequency component, and across all stages, the model significantly enhances important targets, with this weight being the highest among all components. This further emphasizes the importance of low-frequency information throughout the feature extraction process. Additionally, high-frequency components contain more detailed information (as shown in Stage 2 of Fig. 5(a), where mid-to-high frequency components capture the contours of the bicycle and rider, assigning them relatively higher weights, and in Stage 2 of Fig. 5(b), which captures the shark's eyes and teeth). However, as the frequency increases, the weights of the components gradually diminish, with some high-frequency components being suppressed. This suggests that the model selectively attends to high-frequency components, considering only a few critical details. In contrast, the mid-frequency and low-frequency components capture the majority of relevant features. Thus, the reduction in high-frequency component weights can effectively minimize feature redundancy and filter out irrelevant noise, reinforcing the model's focus on essential information. It is also noteworthy that, as seen in Fig. 5(b), high-frequency components are less likely to be modulated as the network progresses to deeper stages. A direct reason for this is that deeper networks tend to focus on extracting high-level semantic features, such as object categories or scene information, rather than detailed textures. Moreover, at deeper stages, the feature representation becomes more sparse and compact, and the model has already extracted sufficient edges and textures from mid-frequency and high-frequency components in the earlier stages. Therefore, high-frequency information no longer provides significant discriminative power in deeper layers, leading to a reduction in modulation strength.

### 5.2. Analysis of effective receptive fields

The importance of expanding the effective receptive field for improving representations [79–83] has been widely demonstrated to be highly effective. The IMAM employs multiform convolutions to capture the spatial context of each pixel across different scales. By leveraging the captured multi-scale contextual information, the model adaptively generates appropriate weights for various frequency components within the corresponding regions, effectively modulating both low-frequency and high-frequency components and optimizing frequency sparsity. As depicted in Fig. 6, we randomly select four locations for effective receptive field (ERF) visualization, which clearly demonstrates that D3 method exhibits higher contribution and broader coverage. Although D3 follows a relatively small local frequency domain window, it leverages the advantages of multiform convolutions to achieve a superior ERF compared to other state-of-the-art pooling methods.

### 5.3. Analysis of scalability and adaptability

We applied the proposed D3 method to image classification and object detection tasks, evaluating its performance on five mainstream backbone networks of different scales using the large-scale ImageNet-1K dataset. On the MSCOCO benchmark, D3 outperformed other state-of-the-art methods in handling fine-grained details across three common detector architectures (two-stage, single-stage, and end-to-end), significantly improving detection accuracy. Additionally, we extended the detection scenarios to noisy, low-light, and UAV perspectives, focusing on small, dense, and distant objects. D3 was also integrated with mainstream YOLO real-time detectors to assess its performance in real-time scenarios. Extensive experiments, as shown in Tables 1, 3, 4, and 5, consistently demonstrate that D3 is highly scalable and adaptable across models and datasets for both classification and detection tasks.

### 5.4. Analysis of inference latency in real-time detectors

We evaluate the inference latency of D3 against other poolings methods on real-time detectors (YOLOv7 to YOLO11) using a single 4090 GPU with FP32. The results in Table 4 show that, while D3 introduces some overhead compared to the baseline and parameter-free methods like SoftPool and ConditionalPool, it requires fewer parameters, less computation, and achieves lower latency than LIP and SAPool. The additional inference latency of D3 is mainly due to the frequency domain transformations in local regions and multi-scale attention modulation. Despite this, the trade-off between accuracy and inference latency in complex scenarios, except for ImageNet-1K and MSCOCO, makes D3 a promising choice for real-time applications, effectively demonstrating the method's efficacy.

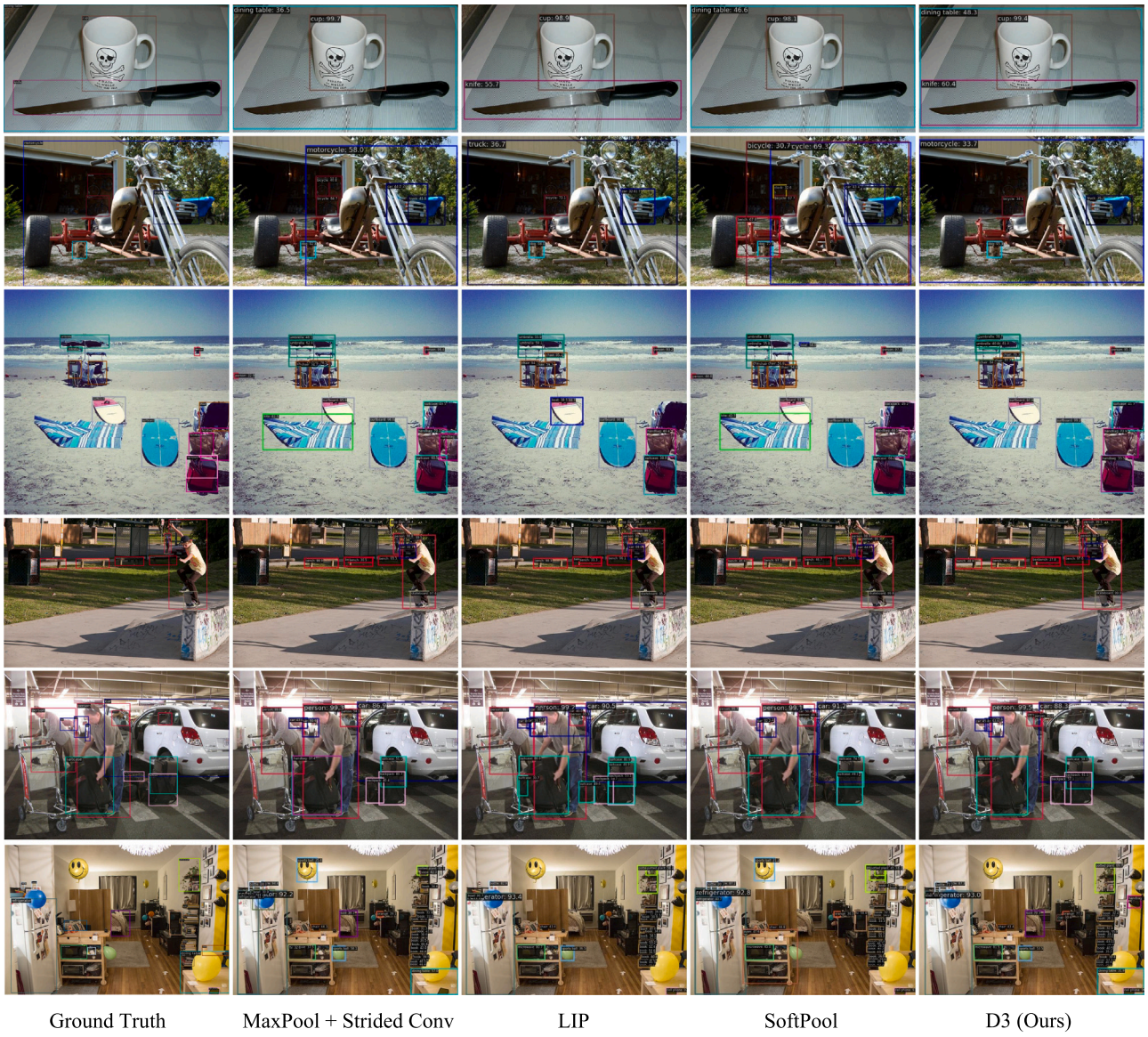



**Fig. 9.** Detection results on the MSCOCO validation set [3] using Cascade R-CNN [4] and RestNet-50 [2].

### 5.5. Analysis of robustness in noisy scenarios

Various types of noise, such as low-light conditions [36], motion blur, occlusion [37], and haze [39], are commonly encountered in real-world scenarios, significantly impacting object recognition. In this work, we evaluate the robustness of the proposed D3 method under these noisy conditions and validate its effectiveness in distinguishing between details and noise. The results in Tables 4 and 5 demonstrate that the proposed D3 outperforms other pooling methods in these challenging scenarios. We also visualize the heatmaps of D3 and the baseline on the HazyDet dataset [39] in Fig. 7, highlighting the advantage of D3 in leveraging the synergistic processing of both frequency and spatial domains to effectively suppress noise while preserving critical details.

### 5.6. Visualization of attention maps

Attention mechanisms represent the model's ability to understand the target by selectively focusing on important parts and suppressing less relevant background. This strategy is widely recognized as an effective way to enhance the model's overall representation capacity [20,84,85]. In our work, IMAM serves as an attention-based frequency modulation mechanism, guiding the adaptive adjustment of local frequency components through multi-scale spatial contextual information, thereby improving feature extraction. In Fig. 8, we visualize the attention maps for different pooling methods. The results indicate that the D3 method effectively extends attention over a broader area, enhancing the focus on significant features. This further substantiates its capability to leverage the advantages of multiform convolutions, thereby capturing more valuable features through a larger contextual associations.

### 5.7. Qualitative results of object detection

We employ Cascade R-CNN and ResNet-50 as baseline models to compare the proposed D3 with MaxPool, Strided Convolution, LIP, and SoftPool, and evaluate their detection performance on MSCOCO, as shown in Fig. 9. Based on the Ground Truth, we can observe that the D3 method effectively reduces the detection of incorrect objects and

categories, such as handbags and beach towels on the beach, and is also able to accurately detect partially occluded objects like bed and bench. For correctly detected objects, it achieves higher confidence scores and covers a larger effective area. This further demonstrates the advantages of proposed method in expanding the receptive field and leveraging the complementary information from both domains.

## 6. Limitations

While the proposed D3 method has demonstrated significant advantages in image classification and object detection tasks, and integrates effectively across various models by directly replacing pooling layers with a stride $> 1$, the visual domain encompasses a broader range of tasks, including instance segmentation, semantic segmentation, 3D reconstruction, and 3D detection. Future work will aim to explore the application potential of our pooling method across these diverse visual tasks.

## 7. Conclusion

In this paper, we analyze the limitations of current pooling methods that rely solely on the spatial domain and explore the advantages of both spatial and frequency domains in handling local features. We propose viewing the design of frequency filtering as a task of optimizing the sparsity of different frequency components. By leveraging rich contextual information from the spatial domain, we guide the dynamic generation of weights for various frequency components, leading to the development of the IMAM filter. Additionally, we investigate the timing sequence of spatial local aggregation and inverse transformation, proposing two distinct aggregation strategies within different domains. Extensive experiments demonstrate that our method outperforms other pooling techniques across multiple visual tasks. We hope this research advances the application of the integration between frequency and spatial domains in diverse real-world applications.

## CRediT authorship contribution statement

**Yunzhong Si:** Writing – review & editing, Writing – original draft, Visualization, Validation, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Huiying Xu:** Writing – review & editing, Supervision, Investigation. **Xinzhong Zhu:** Writing – review & editing, Supervision, Resources, Funding acquisition. **Rihao Liu:** Validation. **Hongbo Li:** Supervision, Resources.

## 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.

## Data availability

Data will be made available on request.

## References


[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255.

[2] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[3] 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: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755.

[4] Z. Cai, N. Vasconcelos, Cascade r-cnn: Delving into high quality object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6154–6162.

[5] 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. 39 (6) (2016) 1137–1149.

[6] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L.M. Ni, H.-Y. Shum, Dino: Detr with improved denoising anchor boxes for end-to-end object detection, 2022, arXiv preprint arXiv:2203.03605.

[7] S. Zhang, X. Wang, J. Wang, J. Pang, C. Lyu, W. Zhang, P. Luo, K. Chen, Dense distinct query for end-to-end object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 7329–7338.

[8] G. Jocher, J. Qiu, Ultralytics YOLO11, 2024, URL: https://github.com/ultralytics/ultralytics.

[9] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, A. Torralba, Scene parsing through ade20k dataset, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 633–641.

[10] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, J. Sun, Unified perceptual parsing for scene understanding, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 418–434.

[11] B. Cheng, A. Schwing, A. Kirillov, Per-pixel classification is not all you need for semantic segmentation, Adv. Neural Inf. Process. Syst. 34 (2021) 17864–17875.

[12] B. Cheng, I. Misra, A.G. Schwing, A. Kirillov, R. Girdhar, Masked-attention mask transformer for universal image segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1290–1299.

[13] E.Y. Lam, J.W. Goodman, A mathematical analysis of the DCT coefficient distributions for images, IEEE Trans. Image Process. 9 (10) (2000) 1661–1666.

[14] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[15] Y.-L. Boureau, J. Ponce, Y. LeCun, A theoretical analysis of feature pooling in visual recognition, in: Proceedings of the 27th International Conference on Machine Learning, ICML-10, 2010, pp. 111–118.

[16] Z. Gao, L. Wang, G. Wu, Lip: Local importance-based pooling, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3355–3364.

[17] A. Stergiou, R. Poppe, G. Kalliatakis, Refining activation downsampling with SoftPool, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 10357–10366.

[18] F. Chen, G. Datta, S. Kundu, P.A. Beerel, Self-attentive pooling for efficient deep learning, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 3974–3983.

[19] A. Vaswani, Attention is all you need, Adv. Neural Inf. Process. Syst. (2017).

[20] A. Dosovitskiy, An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.

[21] M.D. Zeiler, R. Fergus, Stochastic pooling for regularization of deep convolutional neural networks, 2013, arXiv preprint arXiv:1301.3557.

[22] D. Yu, H. Wang, P. Chen, Z. Wei, Mixed pooling for convolutional neural networks, in: Rough Sets and Knowledge Technology: 9th International Conference, RSKT 2014, Shanghai, China, October 24-26, 2014, Proceedings 9, Springer, 2014, pp. 364–375.

[23] B. Graham, Fractional max-pooling, 2014, arXiv preprint arXiv:1412.6071.

[24] Z. Shi, Y. Ye, Y. Wu, Rank-based pooling for deep convolutional neural networks, Neural Netw. 83 (2016) 21–31.

[25] S. Zhai, H. Wu, A. Kumar, Y. Cheng, Y. Lu, Z. Zhang, R. Feris, S3pool: Pooling with stochastic spatial sampling, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4970–4978.

[26] F. Saeedan, N. Weber, M. Goesele, S. Roth, Detail-preserving pooling in deep networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9108–9116.

[27] E. Bayraktar, C.B. Yigit, Conditional-pooling for improved data transmission, Pattern Recognit. (2023) 109978, http://dx.doi.org/10.1016/j.patcog.2023.109978, URL: https://www.sciencedirect.com/science/article/pii/S0031320323006763.

[28] Z. Qin, P. Zhang, F. Wu, X. Li, Fcanet: Frequency channel attention networks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 783–792.

[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.

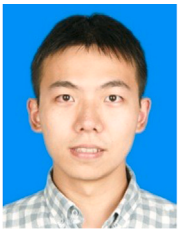
[30] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, J. Sun, Repvgg: Making vgg-style convnets great again, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13733–13742.

[31] P.K.A. Vasu, J. Gabriel, J. Zhu, O. Tuzel, A. Ranjan, Mobileone: An improved one millisecond mobile backbone, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 7907–7917.

[32] 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, 2023, pp. 7464–7475.

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

[34] C.-Y. Wang, I.-H. Yeh, H.-Y.M. Liao, Yolov9: Learning what you want to learn using programmable gradient information, 2024, arXiv preprint arXiv:2402.13616.
[35] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, G. Ding, Yolov10: Real-time end-to-end object detection, 2024, arXiv preprint arXiv:2405.14458.
[36] Y.P. Loh, C.S. Chan, Getting to know low-light images with the exclusively dark dataset, Comput. Vis. Image Underst. 178 (2019) 30–42, http://dx.doi.org/10.1016/j.cviu.2018.10.010.
[37] D. Du, P. Zhu, L. Wen, X. Bian, H. Lin, Q. Hu, T. Peng, J. Zheng, X. Wang, Y. Zhang, et al., VisDrone-DET2019: The vision meets drone object detection in image challenge results, in: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019.
[38] M. Everingham, S.A. Eslami, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, Int. J. Comput. Vis. 111 (2015) 98–136.
[39] C. Feng, Z. Chen, R. Kou, G. Gao, C. Wang, X. Li, X. Shu, Y. Dai, Q. Fu, J. Yang, HazyDet: Open-source benchmark for drone-view object detection with depth-cues in hazy scenes, 2024, arXiv preprint arXiv:2409.19833.
[40] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.
[41] A.G. Howard, Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017, arXiv preprint arXiv:1704.04861.
[42] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., Searching for mobilenetv3, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1314–1324.
[43] D. Qin, C. Leichner, M. Delakis, M. Fornoni, S. Luo, F. Yang, W. Wang, C. Banbury, C. Ye, B. Akin, et al., MobileNetV4-universal models for the mobile ecosystem, 2024, arXiv preprint arXiv:2404.10518.
[44] C.-Y. Wang, H.-Y.M. 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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 390–391.
[45] M. Tan, Efficientnet: Rethinking model scaling for convolutional neural networks, 2019, arXiv preprint arXiv:1905.11946.
[46] M. Tan, Q. Le, Efficientnetv2: Smaller models and faster training, in: International Conference on Machine Learning, PMLR, 2021, pp. 10096–10106.
[47] P.W. Battaglia, J.B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al., Relational inductive biases, deep learning, and graph networks, 2018, arXiv preprint arXiv:1806.01261.
[48] R.C. Gonzales, P. Wintz, Digital Image Processing, Addison-Wesley Longman Publishing Co., Inc., 1987.
[49] I. Pitas, Digital Image Processing Algorithms and Applications, Vol. 2, John Wiley & Sons Inc, 2000, pp. 133–138, Google schola.
[50] D. Yin, R. Gontijo Lopes, J. Shlens, E.D. Cubuk, J. Gilmer, A fourier perspective on model robustness in computer vision, Adv. Neural Inf. Process. Syst. 32 (2019).
[51] H. Wang, X. Wu, Z. Huang, E.P. Xing, High-frequency component helps explain the generalization of convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8684–8694.
[52] L. Kong, J. Dong, J. Ge, M. Li, J. Pan, Efficient frequency domain-based transformers for high-quality image deblurring, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 5886–5895.
[53] Y. Cui, Y. Tao, W. Ren, A. Knoll, Dual-domain attention for image deblurring, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2023, pp. 479–487.
[54] Y. Cui, A. Knoll, Dual-domain strip attention for image restoration, Neural Netw. 171 (2024) 429–439.
[55] Y. Cui, Y. Tao, Z. Bing, W. Ren, X. Gao, X. Cao, K. Huang, A. Knoll, Selective frequency network for image restoration, in: The Eleventh International Conference on Learning Representations, 2023.
[56] Y. Cui, W. Ren, X. Cao, A. Knoll, Image restoration via frequency selection, IEEE Trans. Pattern Anal. Mach. Intell. (2023).
[57] I.N. Sneddon, Fourier Transforms, Courier Corporation, 1995.
[58] H.J. Nussbaumer, H.J. Nussbaumer, The Fast Fourier Transform, Springer, 1982.
[59] C.S. Burrus, R.A. Gopinath, H. Guo, Wavelets and Wavelet Transforms, houston ed., Vol. 98, Rice university, 1998.
[60] L. Chen, L. Gu, D. Zheng, Y. Fu, Frequency-adaptive dilated convolution for semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 3414–3425.
[61] C.F. Van Loan, The ubiquitous Kronecker product, J. Comput. Appl. Math. 123 (1–2) (2000) 85–100.
[62] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1916.
[63] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, M. Li, Bag of tricks for image classification with convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 558–567.
[64] N. Park, S. Kim, How do vision transformers work? in: International Conference on Learning Representations, 2022.
[65] O. Rippel, J. Snoek, R.P. Adams, Spectral representations for convolutional neural networks, Adv. Neural Inf. Process. Syst. 28 (2015).
[66] F. Yu, V. Koltun, T. Funkhouser, Dilated residual networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 472–480.
[67] 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, IEEE Computer Society, 2021, pp. 3490–3499.
[68] 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, Springer, 2020, pp. 213–229.
[69] M. Contributors, OpenMMLab's pre-training toolbox and benchmark, 2023, https://github.com/open-mmlab/mmpretrain.
[70] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C.C. Loy, D. Lin, MMDetection: Open MMLab detection toolbox and benchmark, 2019, arXiv preprint arXiv:1906.07155.
[71] C. Gulcehre, K. Cho, R. Pascanu, Y. Bengio, Learned-norm pooling for deep feedforward and recurrent neural networks, in: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I 14, Springer, 2014, pp. 530–546.
[72] T. Kobayashi, Gaussian-based pooling for convolutional neural networks, Adv. Neural Inf. Process. Syst. 32 (2019).
[73] T. Williams, R. Li, Wavelet pooling for convolutional neural networks, in: International Conference on Learning Representations, 2018.
[74] B.S. Wang, J.W. Hsieh, P.Y. Chen, M.C. Chang, L. Ke, S. Lyu, Ldw-pooling: Learnable discrete wavelet pooling for convolutional networks, in: Proc. Brit. Mach. Vis. Conf., Virtual Conf., British Machine Vision Association, 2021.
[75] K.A. Joshi, D.G. Thakore, A survey on moving object detection and tracking in video surveillance system, Int. J. Soft Comput. Eng. 2 (3) (2012) 44–48.
[76] E. Yurtsever, J. Lambert, A. Carballo, K. Takeda, A survey of autonomous driving: Common practices and emerging technologies, IEEE Access 8 (2020) 58443–58469.
[77] 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, 2018, pp. 8759–8768.
[78] 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, 2017, pp. 618–626.
[79] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A convnet for the 2020s, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11976–11986.
[80] X. Ding, X. Zhang, J. Han, G. Ding, Scaling up your kernels to 31x31: Revisiting large kernel design in cnns, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11963–11975.
[81] T. Huang, L. Yin, Z. Zhang, L. Shen, M. Fang, M. Pechenizkiy, Z. Wang, S. Liu, Are large kernels better teachers than transformers for convnets? in: International Conference on Machine Learning, PMLR, 2023, pp. 14023–14038.
[82] W. Yu, P. Zhou, S. Yan, X. Wang, Inceptionnext: When inception meets convnext, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 5672–5683.
[83] H. Chen, X. Chu, Y. Ren, X. Zhao, K. Huang, PeLK: Parameter-efficient large kernel ConvNets with peripheral convolution, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 5557–5567.
[84] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.
[85] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, Cbam: Convolutional block attention module, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 3–19.



**Yunzhong Si** received B.E. degrees from Hunan University of Technology, China. He is currently pursuing an M.E. degree in electronic information from Zhejiang Normal University, China. His current research focuses are centered around deep learning, computervision, and object detection.

**Huiying Xu** received the M.S. degree from National University of Defense Technology(NUDT), China. She is an associate professor with the School of Computer Science and Technology, Zhejiang Normal University, and also the researcher of Research Institute of Ningbo Cixing Co. Ltd, PR, China. Her research interests include Kernel learning and feature selection, Object Detection, Vision SLAM, Computer vision, Image processing, Pattern recognition, Computer simulation, Deep clustering, Generative Adversarial Network, Diffusion Model, Clustering Ensemble, Multiple Kernel Learning, Learning with incomplete data and their applications. She is a member of the China Computer Federation. She has published papers, including those in highly regarded journals such International Journal of Intelligent Systems, IEEE Transactions on Cybernetics, IEEE Transactions on Multimedia, etc.

**Xinzhong Zhu** received the Ph.D. degree from Xidian University and M.S. degree from National University of Defense Technology (NUDT), China. He is a professor with the School of Computer Science and Technology, Zhejiang Normal University, and also the chief scientist of Beijing Geekplus Technology Co., Ltd. and president of Research Institute of Ningbo Cixing Co., Ltd., China. His research interests include Machine learning, Deep clustering, Computer vision, Manufacturing informatization, Robotics and System integration, Laser SLAM, Vision SLAM, Diffusion Model, Low Quality Data Learning, Multiple Kernel Learning, and Intelligent manufacturing. He is a member of the ACM and certified as CCF distinguished member. Dr. Zhu has published more than 30 peer-reviewed papers, including those in highly regarded journals and conferences such as the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Image Processing, the IEEE Transactions on Multimedia, the IEEE Transactions on Knowledge and Data Engineering, CVPR, NeurIPS, AAAI, IJCAI, etc. He served on the Technical Program Committees of IJCAI 2020 and AAAI 2020.

**Rihao Liu** received B.E. degrees from Hangzhou Dianzi University, China. He is currently pursuing an M.E. degree inelectronic information from Zhejiang Normal University, China. His current research focuses are centered around deep learning, computer vision, and diffusion models

**Hongbo Li**, received his Ph.D. degree in computer science from Tsinghua University in 2009. Currently, he holds the position of the Chief Technology Officer and Co-founder of Beijing Geek+ Technology Co., Ltd China. In addition, he also serves as the secretary-general of Chinese Intelligent service Society and is an Editorial Board Member of several high-profile journals. His research interests include the design and application of intelligent robots, intelligent information process, and intelligent logistic systems. He has published more than 70 papers in prestigious journals and conference, and has been awarded more than 120 patents, including 46 international invention patents.