Article

http://justc.ustc.edu.cn

Received: August 27, 2021; Accepted: February 09, 2022

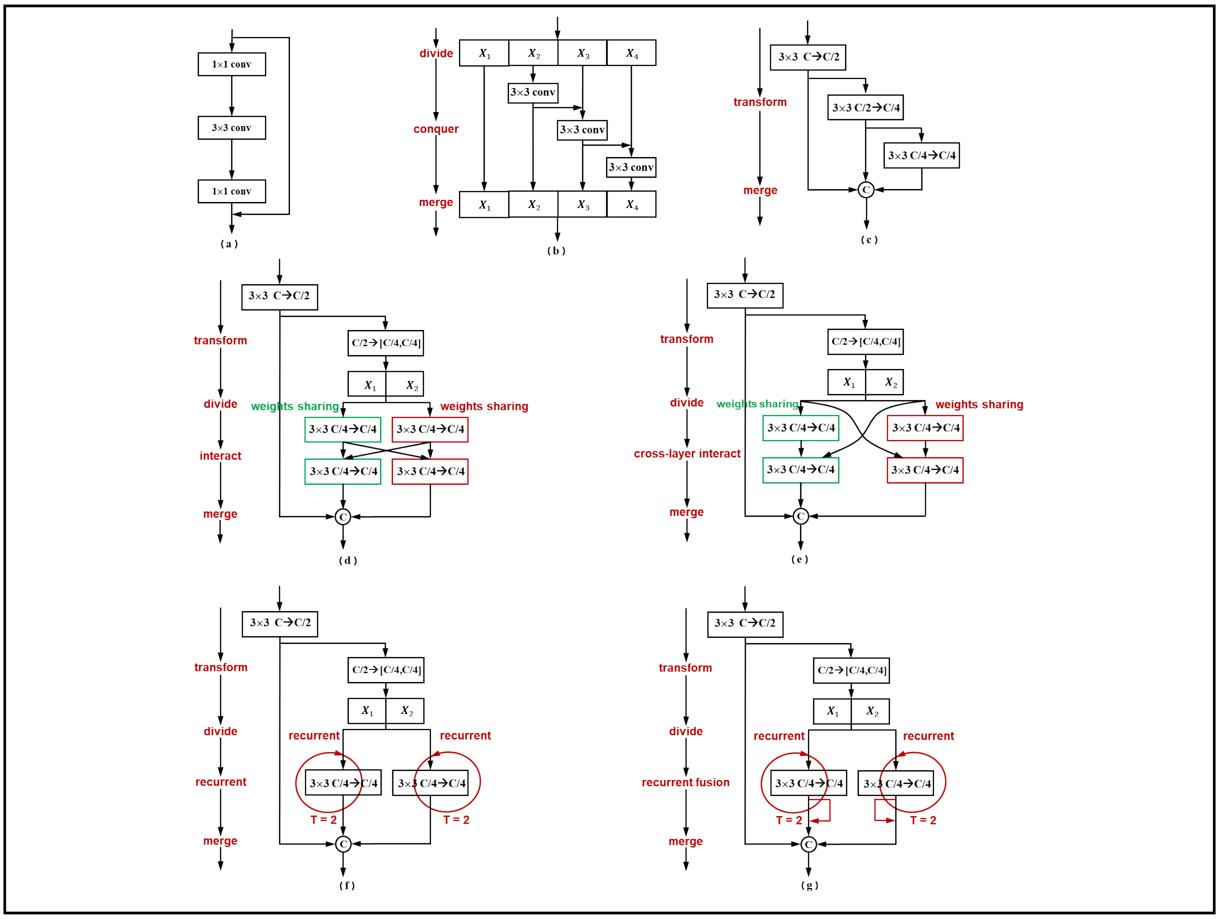
SIS: A new multi-scale convolutional operator

Man Zhou, Xueyang Fu ✉, and Aiping Liu

School of Information Science and Tecnology, University of Science and Technology of China, Hefei 230027 China

✉Correspondence: Xueyang Fu , E-mail: xyfu@ustc.edu.cn

Graphical abstract



We design a series of transformation mechanism to expand the module (a) and Res2Net module (b) into multi-scale module operators (c, d, e, f, g).

Public summary

- A more lightweight and representative scale-in-scale operator, namely SIS is proposed. The operator can be plugged into any promising backbone to replace regular convolution operator.

- To be more efficient, an improved SIS series is proposed. The series maintains promising results within nearly half parameters.

- Extensive experiments demonstrate the superior performance of our SIS operator series compared with state-of-the-art methods on image classification, object detection, key points estimation and semantic segmentation.

Citation: Zhou M, Fu X Y, Liu A P. SIS: A new multi-scale convolutional operator. *JUSTC*, **2022**, 52(4): 7. DOI: 10.52396/JUSTC-2021-0188

JUSTC

Article

http://justc.ustc.edu.cn

Received: August 27, 2021; Accepted: February 09, 2022

SIS: A new multi-scale convolutional operator

Man Zhou, Xueyang Fu ✉, and Aiping Liu

School of Information Science and Tecnology, University of Science and Technology of China, Hefei 230027 China

✉Correspondence: Xueyang Fu , E-mail: xyfu@ustc.edu.cn

Cite This: *JUSTC*, **2022**, 52(4): 7 (9pp)   Read Online

Abstract: Visual features with high potential for generalization are critical for computer vision applications. In addition to the computational overhead associated with layer-by-layer feature stacking to produce multi-scale feature maps, existing approaches also incur high computational costs. To address this issue, we present a compact and efficient scale-in-scale convolution operator called SIS by incorporating an efficient progressive multi-scale architecture into a standard convolution operator. More precisely, the suggested operator uses the channel transform-divide-and-conquer technique to optimize conventional channel-wise computing, thereby lowering the computational cost while simultaneously expanding the receptive fields within a single convolution layer. Moreover, the proposed SIS operator incorporates weight-sharing with split-and-interact and recur-and-fuse mechanisms for enhanced variant design. The suggested SIS series is easily pluggable into any promising convolutional backbone, such as the well-known ResNet and Res2Net. Furthermore, we incorporated the proposed SIS operator series into 29-layer, 50-layer, and 101-layer ResNet as well as Res2Net variants and evaluated these modified models on the widely used CIFAR, PASCAL VOC, and COCO2017 benchmark datasets, where they consistently outperformed state-of-the-art models on a variety of major vision tasks, including image classification, key point estimation, semantic segmentation, and object detection.

Keywords: multi-scale convolutional operator; image classification; key point estimation; semantic segmentation; object detection

CLC number: TP391   **Document code:** A

1 Introduction

Multi-scale patterns are pervasive in natural scenes[1−4]. Therefore, it is well recognized that acquiring scale-invariant feature representation is of critically imperative for various computer vision tasks, such as object detection[5], key point estimation[6], panoptic segmentation[7], image classification[8], salient object detection, depth estimation[9], image restoration[10, 11], and scene analysis. In recent years, considerable progress has been made in handcrafted scale-invariant feature operators and architectures. For instance, scale-invariant feature transform(SIFT)[12] is the most discriminative operator that constructs a difference of Gaussian pyramid(DOG) and searches for extreme points as feature representations. Following this, image pyramid and feature pyramid architecture series have also been proposed. Image pyramid and feature pyramid are implemented by rescaling the current images or feature maps to be stacked as a pyramid and then operating at each layer. Despite these promising achievements, these designs suffer from a redundant computing burden.

Instead of explicitly operating, deep learning has been introduced into multi-scale feature designs, and has advocated a series of promising works. The representative module, namely the inception module[13], employs a multi-path mechanism to design multiple convolution operators of different receptive fields, for example, 1×1, 3×3, and 5×5. Accordingly, the output of each sub-path is concatenated to implicitly achieve a multi-scale expression. However, compared with a single receptive operator, multiple paths increase the computational cost. Therefore, dense connections have bridged the feature flow of different scales, incurring additional memory resource consumption and inference time. Recently, the Res2Net[14] block was proposed to address this issue by constructing hierarchical residual-like connections on a channel-splitting feature map. However, Res2Net achieves multiscale representation with a single-layer block at the cost of memory resources and inference time.

To this end, we propose a more lightweight and representative scale-in-scale operator, namely SIS, by transforming a regular convolution operator into an efficient gradual multi-scale architecture. More specifically, the proposed operator exploits the channel transform in a divide-and-conquer fashion to optimize the computing of the entire channel, which relieves the computing burden and expands the range of receptive fields within a single layer. Moreover, we improved the proposed SIS operator by integrating the weight-sharing mechanism and split-and-interaction mechanism to develop a more lightweight and efficient design. Furthermore, the proposed SIS series can be plugged into any promising convolution backbone, such as ResNet and Res2Net. We deployed the proposed SIS operator series into the 29-layer, 50-layer, and 101-layer variants of both ResNet as well as Res2Net and evaluated the models on several major computer vision tasks, including image classification, key point estimation, semantic segmentation, and object detection. Extensive experimental results confirm that the proposed SIS operator series can effectively enhance feature representation and achieve state-of-the-art performance while reducing the number of parameters by almost half. In summary, the major contributions of this study are as follows:

(Ⅰ) A lightweight and representative scale-in-scale operator, namely the SIS, is proposed. The operator can be plugged into any promising backbone to replace a regular convolution operator.

(Ⅱ) An improved SIS series was proposed that can achieve high efficiency. The series maintained promising results for nearly half of the parameters.

(Ⅲ) Extensive experiments demonstrated the superior performance of the proposed SIS operator series compared with state-of-the-art methods in image classification, object detection, key point estimation, and semantic segmentation.

2 Related work

CNN backbone. At present, the backbone is an integral component in a convolutional neural network. It fundamentally confirms the effectiveness of extracting features for computer vision tasks, such as object detection, segmentation, and classification. A basic convolutional neural networks(CNN) comprises three structures: convolution, activation, and pooling. The output of the CNN was the specific feature space of each image. When handling image classification tasks, we take the feature space of the CNN output as the input of the fully connected neural network(FCN) and use the FCN to complete the mapping from the input image to the label set, that is, classification. Undoubtedly, the most significant aspect of the entire process is how to iteratively adjust the network weight through the training data, which is called the back-propagation algorithm. Currently, mainstream CNN, such as VGg and RESNET, are adjusted and combined using a simple CNN. Additionally, a more effective backbone can enhance the performance of these tasks. AlexNet[15] is a pioneer in exploiting CNN to a deep network, which achieves exemplary performance on classification, exceeding that of traditional methods. VGG[16] was designed to stack smaller convolution operations to increase the network depth and reduce the number of network parameters. As the depth of the network increased, network degradation occurred, which led to worse experimental results. To address this issue, Res-Net[17] promotes network depth with a bottleneck module that can be simply applied to improve the model performance. Furthermore, Res-Next[18] utilizes group convolution to conjointly learn different representation subspaces. Similarly, DenseNet[19] devises densely connected layers to mutually connect all layers, making more effective use of features and strengthens the feature transfer. Several lightweight backbones have been proposed to reduce the computational consumption of CNN while maintaining accuracy, such as SqueezeNet[20], MobileNet[21], and ShuffleNet[22].

Multi-scale operator. It is worth mentioning that multi-scale features include a multitude of information, resulting in increased accuracy. NIN[23] used multilayer perceptron convolution to improve the discriminability of feature abstractions for local networks, rather than the traditionally used basic convolution layer. Additionally, GoogleNet[13] uses an inception module to extract multi-scale features. MSDNet[8] performed budget prediction by integrating multi-scale feature maps with dense connections. Although FishNet[24] retains multi-scale features at various depths, it also refines them to boost feature variety via an up-sampling process that has been demonstrated to be successful for detection. As the number of network layers increases, DLA[25] builds an iterative deep aggregation module for aggregating and refining different scales and resolutions, thereby improving multiscale representation. HRNet[6] proposed the use of exchange units to connect disparate sub-networks. These exchange units collect feature information from other subnetworks through parallel multi-scale. LanczosNet[26] implemented the Lanczos algorithm in a multiscale graph convolution network, which enabled effective matrix power computation and, hence, facilitates the collection of multi-scale data. Additionally, Res2Net[14] uses multi-scale analysis to extract information and expand receptive fields at a more granular level.

Computer vision. Computer vision refers to the study of the vision capability of a computer, or the ability of a machine to visually analyze its environment and stimuli. Generally, machine vision is employed to evaluate images and movies. Machine vision, as defined by the british machine vision association(BMVA), is an “automated extraction, analysis, and comprehension of relevant information from a single image or a series of images.” The primary tasks in computer vision include picture classification and localization, target recognition and tracking, semantic segmentation, and instance segmentation. In terms of image classification, determining whether an image contains an object and how to describe its features have been the primary study topics. Generally, the object classification algorithm defines the entire image using either hand-drawn features or feature learning and then uses a classifier to determine whether a particular type of object exists. Convolutional neural networks(CNNs) are the most frequently used method for image categorization. The CNN network structure is fundamentally built of three layers: convolution, pooling, and full connection. Typically, the input image is delivered to a CNN, and the network extracts features via a convolution layer. Subsequently, the details are filtered using a pooling layer(generally, maximum pooling and average pooling). Ultimately, the feature is enlarged in the entire connection layer, and the classification results are transmitted to the relevant classifier. Specifically, it depends on high-level abstract semantic information. Object detection is the act of locating a target inside a scene(picture), which includes two processes: Detection(where) and recognition(what). The complexity of a task involves extracting and recognizing candidate locations for detection. Consequently, the framework of the task is as follows: First, the model for extracting candidate regions from the scene is built, followed by the model for identifying candidate regions. Thereafter, the parameters of the classification model and the location of the effective candidate frames were refined. The term ‘target detection and recognition’ refers to the process of extracting candidate regions from photographs or videos. It is a computer vision problem that entails distinguishing the target from the uninterested component, determining whether there is a target, and subsequently determining the location of a target if a target exists. Numerous scholars have concentrated their efforts on feature fusion using the FPN architecture. Although it achieves some performance gains, it has a high computational cost. Other computer vision tasks have similar impacts.

3 Scale-in-scale operator series

In this section, we present the proposed SIS operator series, which is a lightweight and representative scale-in-scale operator. The complete technical pipeline is illustrated in Fig.1. The operator series can be divided into three types: ① Basic SIS with a channel transform-and-merge mechanism; ② Interactive SIS with inter-path and cross-path interactions; ③ Recurrent SIS with inter-layer and cross-layer fusing. All parts are illustrated in the following subsections.

3.1 Basic SIS

The basic SIS module is illustrated in Fig.1c. Compared with a conventional convolution operating on the entire channel, the basic SIS module employs a channel transform-and-merge mechanism to reduce the computational burden. Referred by Res2Net, the Res2Net module constructs residual-like multipath operations to obtain multi-scale features while achieving promising performance and almost increasing computing load. It can be attributed to the feature channel splitting technique used to relieve the multi-path operation. However, each subpath of Res2Net still requires a regular convolution operation. It is explicitly necessary to continue the optimization. Therefore, we propose a scale-in-scale mechanism to replace regular multi-path convolution with a basic SIS. As shown in Figs. 1b and 1h, the channel-splitting-guided multiscale information passes through the channel transform-and-merge multi-scale module. Therefore, we refer to the pipeline as a scale-in-scale mechanism. Formally, the regular convolution operator is defined as

$$y_i = k_i * X = \sum_{j=1}^{C_{\text{in}}} k_i^j * x_j \tag{1}$$

Here, $X = [\ x_1, \ldots, x_{C_{\text{in}}}] \in R^{C_{\text{in}} \times H \times W}$ represents the input feature, $Y = [\ y_1, \ldots, y_{C_{\text{out}}}] \in R^{C_{\text{out}} \times H \times W}$ represents the output feature, $K = [\ k_1, \ldots, k_{C_{\text{out}}}] \in R^{C_{\text{out}} \times C_{\text{in}} \times k \times k}$ denotes the convolution filters, and each $K_i \in R^{C_{\text{in}} \times k \times k}$ transforms the input feature with all channels by summation to a single dimension of the output feature. Finally, all outputs of each filter were merged as the output feature. Instead of operating all the channels, we propose a basic SIS module, which is elaborated below:

$$Y = [\ Y_1, Y_2, Y_3\ ] = [\ K_1, K_2, K_3] * [\ X, Y_1, Y_2\ ]^{\mathrm{T}} \tag{2}$$

where T symbolizes the matrix transpose operation and the remainder is described as follows:

$$Y_1 = K_1 * X = \sum_{j=1}^{C_{\text{in}}} K_1^j * X_j \tag{3}$$

$$Y_2 = K_2 * Y_1 = \sum_{j=1}^{C_{\text{in}}/2} K_2^j * Y_1^j \tag{4}$$

$$Y_3 = K_3 * Y_2 = \sum_{j=1}^{C_{\text{in}}/4} K_3^j * Y_2^j \tag{5}$$

Here, X is denoted as above, and $Y = [\ Y_1, Y_2, Y_3\ ] \in R^{C_{\text{out}} \times H \times W}$ represents the output feature, where the $K_1 \in R^{C_{\text{out}/2} \times C_{\text{in}} \times H \times W}$ filter transforms the input feature $X \in R^{C_{\text{in}} \times H \times W}$ to $Y_1 \in R^{C_{\text{out}/2} \times H \times W}$ by 1/2 channel reduction and $K_2 \in R^{C_{\text{out}/4} \times C_{\text{out}/2} \times H \times W}$ operates the intermediate result $Y_1 \in R^{C_{\text{out}/2} \times H \times W}$ as $Y_2 \in R^{C_{\text{out}/4} \times H \times W}$ by 1/4 original channel, $K_3 \in R^{C_{\text{out}/4} \times C_{\text{out}/4} \times H \times W}$ operates the intermediate result $Y_2 \in R^{C_{\text{out}/4} \times H \times W}$ as $Y_3 \in R^{C_{\text{out}/4} \times H \times W}$ by 1/4 original channel. In this study, we only consider the input channel C_{in} identical to the output channel number C_{out}, that is, $C = C_{\text{in}} = C_{\text{out}}$. Observing the above equation to analyze the computing complexity, the calculation is as

$$\frac{P_{\text{SIS}}}{P_{\text{regular}}} = \frac{1/4 \times 3 \times [C \times C/2 \times k \times k + C/2 \times C/4 \times k \times k + C/4 \times C/4 \times k \times k]}{C \times C \times k \times k} = \frac{33}{64} \tag{6}$$

$$\frac{P_{\text{SIS}}}{P_{\text{Res2Net}}} = \frac{1/4 \times 3 \times [C \times C/2 \times k \times k + C/2 \times C/4 \times k \times k + C/4 \times C/4 \times k \times k]}{1/4 \times 3 \times [C \times C \times k \times k]} = \frac{11}{16} \tag{7}$$

Here, P_{regular} and P_{SIS} describe the number of parameters of the regular convolution and the proposed basic scale-in-scale convolution operator, respectively.

3.2 Interactive SIS

To achieve higher efficiency, the basic SIS operator was improved using the channel splitting and cross-path interaction techniques. The corresponding details are demonstrated in Figs. 1d and 1e. Comparing the basic SIS and interactive SIS modules, the transformation from 1/2 to 1/4 of the original channel is replaced with a channel divide(1/2 of the original channel is split into 1/4 and 1/4 of the original channel). The uniformly split feature is passed through a two-path convolution operation, each of which shares a weight.

Specifically, the $K_1 \in R^{C_{\text{out}/2} \times C_{\text{in}} \times H \times W}$ filter transforms the input feature $X \in R^{C_{in} \times H \times W}$ to $Y_1 \in R^{C_{\text{out}/2} \times H \times W}$, and assuming that the input feature $Y_1 \in R^{C/2 \times H \times W}$ is uniformly split into $Y_{11} \in R^{C/4 \times H \times W}$ and $Y_{12} \in R^{C/4 \times H \times W}$, the two-path convolution operator is denoted as $K_2 \in R^{C/4 \times C/4 \times H \times W}$ and $K_3 \in R^{C/4 \times C/4 \times H \times W}$, respectively. Considering Figs. 1d and 1e, the interlayer interactive workflow is described as follows:

$$Y_{111} = K_2 * Y_{11} \tag{8}$$

$$Y_{121} = K_3 * Y_{12} \tag{9}$$

$$Y_{112} = K_2 * (Y_{111} + Y_{121}) = K_2 * K_2 * Y_{11} + K_2 * K_3 * Y_{12} \tag{10}$$

$$Y_{122} = K_3 * (Y_{111} + Y_{121}) = K_3 * K_3 * Y_{12} + K_3 * K_2 * Y_{11} \tag{11}$$

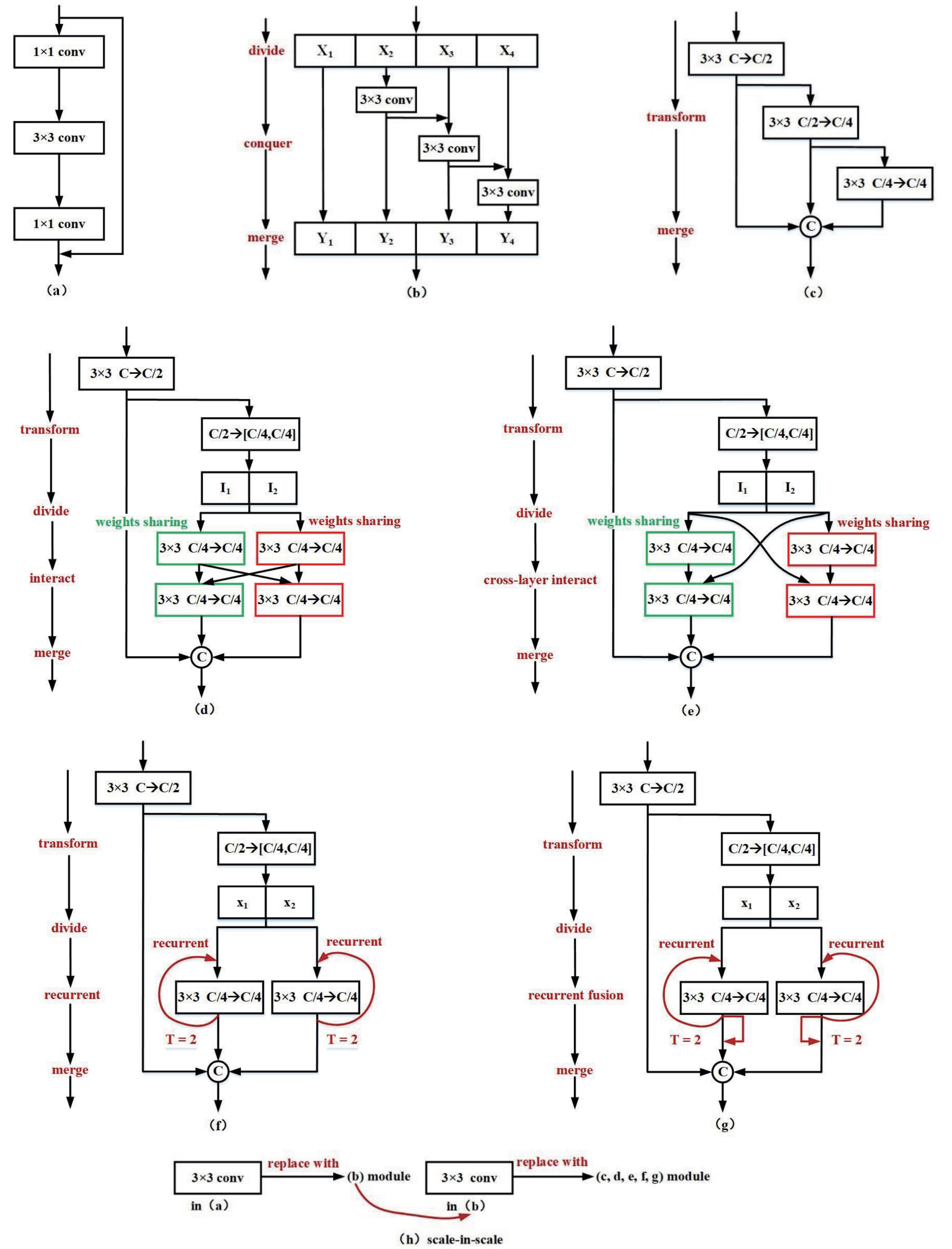


Fig. 1. Comparison between bottleneck modules(a), Res2Net module(b) and SIS module series(c, d, e, f, g) and illustration of scale-in-scale mechanism (h).

$$Y =[Y_1, Y_{112}, Y_{122}] = [K_1|K_2 * (K_2 + K_3), K_3 * (K_2 + K_3)] * [X|Y_{11}, Y_{12}]^{\mathrm{T}} \tag{12}$$

Meanwhile, the cross-layer interactive workflow is described as follows:

$$Y_{112} = K_2 * (Y_{111} + Y_{12}) = K_2 * K_2 * Y_{11} + K_2 * Y_{12} \tag{13}$$

$$Y_{122} = K_3 * (Y_{11} + Y_{121}) = K_3 * K_3 * Y_{12} + K_3 * Y_{11} \tag{14}$$

$$Y = [Y_1, Y_{112}, Y_{122}] = [K_1|K_2 * (K_2+1), K_3 * (1+K_3)] * [X|Y_{11}, Y_{12}]^{\mathrm{T}} \tag{15}$$

When comparing the two interactives, the cross-layer interaction retains the original data, while the inter-layer interaction further modifies the feature. However, regardless of the type, it makes full use of this information.

In terms of the weight-sharing approach, interactive SIS parameters are lighter than those of basic SIS, Res2Net, and regular convolution. The equations above can be used to determine the difficulty of the computation, the calculation is as follows:

$$\frac{P_{\mathrm{SIS}}}{P_{\mathrm{regular}}} = \frac{1/4 \times 3 \times [C \times C/2 \times k \times k + C/4 \times C/4 \times k \times k + C/4 \times C/4 \times k \times k]}{C \times C \times k \times k} = \frac{15}{32} \tag{16}$$

$$\frac{P_{\mathrm{I-SIS}}}{P_{\mathrm{Res2Net}}} = \frac{1/4 \times 3 \times [C \times C/2 \times k \times k + C/4 \times C/4 \times k \times k + C/4 \times C/4 \times k \times k]}{1/4 \times 3 \times C \times C \times k \times k} = \frac{5}{8} \tag{17}$$

Here, P_{regular}, P_{Res2Net}, and P_{SIS} describe the number of parameters of regular convolution, Res2Net, and the proposed basic scale-in-scale convolution operator, respectively. Therefore, the improved interactive SIS block significantly exploits the channel transform-divide-inter or cross-layer-and-interactive-merge for a more lightweight and efficient design. The workflow is described as

$$Y = [Y_1, Y_{112} + Y_{111}, Y_{122} + Y_{121}] = [K_1|K_2 * (K_2+2)|K_3 * (2+K_3)] * [X|Y_{11}|Y_{12}]^{\mathrm{T}} \tag{18}$$

Here, | represents partial block matrix computing. The difference between both recurrent variants is that the latter combines the output features of different recurrent time steps as a multi-scale output. Recurrent SIS can make features more representative. The complexity is almost consistent with interactive SIS.

3.3 Recurrent SIS

The interactive analyses solely analyzed the cross-path connections of information flow. However, the self-interaction paths with several layer generation processes also benefit the multi-scale feature map. Inspired by the recurrent FPN and backbone[27], we consider the recurring characteristic to enhance the reusability benefit. To this end, we propose an enhanced alternative, recurrent SIS. This process is depicted in Figs. 1f and 1g.

The recurrent technique may be classified into two types: those that simply output the final result and those that output the intermediate and final results as a multi-scale output. The former corresponds to the transform-divide-recurrent-merge pipeline depicted in Fig.1f, whereas the latter corresponds to the transform-divide-recurrent-fusion-merge flowchart depicted in Fig.1g. The interactive SIS definitions are identical. In formal terms, the first description is as follows.

$$Y_{111} = K_2 * Y_{11} \tag{19}$$

$$Y_{121} = K_3 * Y_{12} \tag{20}$$

$$Y_{112} = K_2 * (Y_{111} + Y_{11}) = K_2 * Y_{11} + K_2 * K_2 * Y_{11} \tag{21}$$

$$Y_{122} = K_3 * (Y_{12} + Y_{121}) = K_3 * Y_{12} + K_3 * K_3 * Y_{12} \tag{22}$$

$$Y = [Y_1, Y_{112}, Y_{122}] = [K_1|K_2 * (K_2+1)|K_3 * (1+K_3)] * [X|Y_{11}|Y_{12}]^{\mathrm{T}} \tag{23}$$

Meanwhile, the latter workflow is described as

$$Y = [Y_1, Y_{112} + Y_{111}, Y_{122} + Y_{121}] = [K_1|K_2 * (K_2+2)|K_3 * (2+K_3)] * [X|Y_{11}|Y_{12}]^{\mathrm{T}} \tag{24}$$

Here, | represents partial block matrix computing. Comparing both recurrences, the difference is that the latter combines the different recurrent steps as a multiscale output. The recurrent SIS makes the features more representative. This complexity is almost consistent with interactive SIS.

3.4 Scale-in-scale pipeline

Res2Net divides the input feature equally into several components and creates many subpaths to process each component. Therefore, rather than using a standard convolution, the multi-scale output is aggregated into a single block. Accordingly, we integrated the SIS series into a multi-path of Res2Net to further reduce the parameters and expand the receptive field. The divided multi-scale feature flows through a multi-scale channel transform-divide-recur/interactive-merge(SIS) algorithm. Thus, the flowchart of the nested multiscale pattern is referred to in this paper as a scale-in-scale pipeline, as illustrated in Fig.1h.

4 Experiments

To demonstrate the efficiency of the proposed SIS operator series, all experiments in this section were conducted using the freely available Pytorch framework. Moreover, the SIS operator was adapted to replace widely used 3×3 kernels in a variety of vision applications, including image classification, object detection, keypoint estimation, and image segmentation, using ResNet and Res2Net. Additionally, we re-implemented the original baselines in the same manner as described in the original study, except that the baseline and SIS-operator-equipped models were trained from the start without loading pre-trained processes. As a comparison, the 29-layer, 50-layer, and 101-layer equipped versions of ResNet and Res2Net were enhanced.

4.1 Results on classification

Following the same setting as reported in Res2Net, the ResNext-29, 8c×64w variant was employed as the classification baseline for the CIFAR-100 dataset. As described in the original work, the ResNeXt-29 basic block was replaced with the Res2Net module, while the remaining configurations remained unaltered. Moreover, the partial Res2Net module was upgraded by replacing them with basic SIS to demonstrate the efficacy of our proposed SIS operator series, while the input channel was divided by four without a remainder or with a recurrent and interactive variant, while the width factor in Res2Net was divided exactly. In addition, all experiments were trained from the beginning, with no pre-trained procedures. Unless otherwise specified, the re-implemented baseline model and its SIS series-equipped variation were trained on four NVIDIA Titian GPUs using the default data augmentation and training method. Additionally, as illustrated in Table 1 and Fig.1, the recurrent or interactive versions were separated into variants I and II. The top-1 test errors and model capacities of the CIFAR-100 dataset are listed in Table 1. The Top-1 error is the predicted label. The largest value in the last probability vector is considered as the prediction result. If the classification of the one with the largest probability in the prediction result is correct, the prediction is correct. Else, the prediction is incorrect. Moreover, Top-5 error represent the top five with the largest probability vector in the last place. If there is a correct probability, the prediction is correct. Otherwise, the prediction is incorrect. Our experimental results demonstrate that the proposed SIS series-enhanced method outperforms the baseline method and other methods that use fewer parameters. This increase in performance demonstrates the efficacy of the proposed SIS operator series. The effect of the time steps of the recurrent SIS is reported in Table-1-1. We inserted the recurrent SIS variants at different time steps into ResNet-29 and ResNext-29 for evaluation. It can be observed that increasing the time steps T yields improvements, but it increases the computational burden. Considering this trade-off, we set $T = 2$.

4.2 Results on object detection

To evaluate the generalizability of our proposed SIS operator series, an SIS-equipped backbone was deployed for the object detection task on the commonly used MS COCO and PASCAL VOC benchmarks. It is well known that multi-scale visual features are beneficial for improving detection results, primarily because the object in the detection dataset exhibits sharp scale variation. The object of the MS COCO dataset was divided into small, medium, and large scales. Furthermore, the PASCAL VOC is the same as above with a relatively weaker scale variation. Therefore, the upgraded backbone with a common convolution operator replaced with the SIS operator can obtain a larger receptive field and more discriminative feature representation. In this work, we selected a promising Faster *R*-CNN architecture as the baseline. All our experiments were implemented on the mmdection using default settings. Moreover, the re-implemented baseline and SIS-equipped versions were trained from scratch without loading pre-trained weights on ResNet and Res2Net of 50 and 101 versions, respectively.

Table 1. Comparison results of ResNet, Res2Net and SIS-equipped variants on CIFAR-100 dataset.

Models	Params	flops	Top-1 error
ResNet-29	36.5M	4.2G	20.50
ResNext-29	34.4M	4.2G	17.90
Res2NeXt-29	33.8M	4.2G	16.93
Interactive SIS-I variant	26.3M	4.36G	16.82
Interactive SIS-II variant	26.5M	4.36G	16.79
Recurrent SIS-I variant	25.7M	4.52G	16.87
Recurrent SIS-II variant	25.9M	4.52G	16.85
Basic SIS variant	28.3M	4.2G	16.74

Table 1-1. Comparison results of ResNet, Res2Net of recurrent SIS on CIFAR-100 dataset.

Models	T=2	T=3	T=4
ResNet-29	20.50	20.45	20.38
ResNext-29	17.90	17.74	17.68

With regard to the COCO dataset, we conducted training on a union of 80 k COCO training images and 35 k validation images (trainval 35 k), and then evaluated the remaining 5 k validation photos (minival) as testing results. We present our findings using the typical COCO metrics, which include average mean, average precision, and recall over different IoU thresholds.

AP is defined as

$$\mathrm{AP}(t) = \frac{\mathrm{TP}(t)}{\mathrm{TP}(t)+\mathrm{FP}(t)},$$

where t denotes the threshold, and AP.5, AP.75, and the threshold (t) are 0.5, 0.75 respectively. While adhering to the same implementation details as the COCO benchmark for the PASCAL VOC dataset, we evaluated the results solely using the AP metric. The results of the comparison of the two object detection benchmarks are listed in Table 2. It can be observed that SIS-equipped variations outperform the ResNet and Res2Net baselines. More precisely, when AP is applied to the COCO dataset, the basic SIS produces the greatest results, outperforming ResNet and Res2Net by 2.6 and 1.2 points, respectively, as well as outperforming AP by 3.0 and 1.3 points.5. For the PASCAL VOC dataset, the basic SIS outperforms ResNet and Res2Net by 2.6 and 0.8 points, respectively. The results indicate that a backbone equipped with SIS may acquire more representative features for object detection, confirming the generalizability of the SIS series.

4.3 Results on semantic segmentation

In terms of semantic segmentation, we start with DeepLab v3+[28] and evaluate it on the PASCAL VOC dataset. Table 3 presents a comparison of the different configurations of SIS operators for semantic segmentation. For evaluation purposes, the mean IoU was employed. The mean IoU aims to calculate the ratio of the intersection and union of two sets of ground truth and predicted values, where the IOU of each class is calculated, accumulated, and averaged over all classes. More specifically, the re-implemented and SIS-equipped techniques were trained in the same manner as Res2Net. The findings demonstrate that our proposed SIS operator is capable of enhancing semantic segmentation performance and generating more representative features. In addition, the basic SIS produces superior results to those of other SIS-equipped approaches.

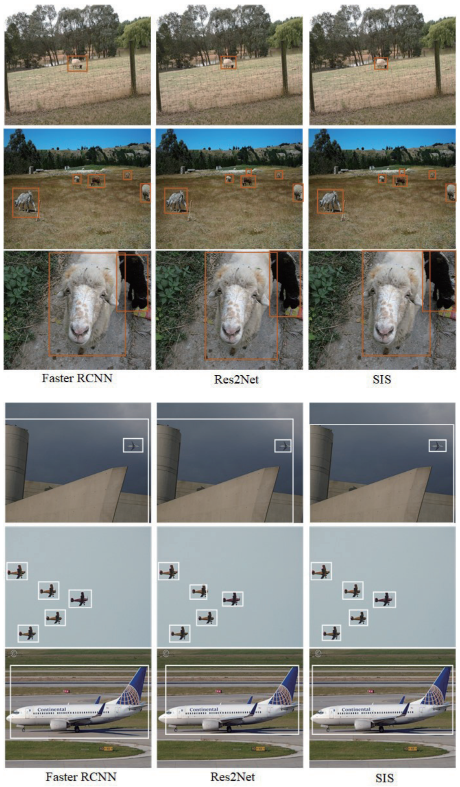


Fig. 2. Visual comparisons between Faster RCNN and SIS equipped Faster RCNN.

4.4 Results on key points estimation

In this study, we demonstrate that the proposed SIS operator series multi-scale representation capability on a keypoint estimation problem. The simple baseline[29] was adapted in a similar fashion to Res2Net[14], except that the backbone was replaced with the proposed SIS series. The COCO keypoint detection dataset and validation set were used to train and test the model, respectively. The solution utilizes both the 50-layer and 101-layer SIS-equipped variants of Res2Net and ResNet. The performance of the keypoint estimation on the COCO validation set with various configurations is presented in Table 4. In particular, our methods beat their counterparts by a considerable margin, demonstrating the capability of multi-scale representation, where AR is the average recall ratio. Specifically, AR.5, AR.75 denote the thresholds as 0.5, 0.75 respectively, while AR(M) and AR(L) represent the recall ratio of medium and large objects in the detection datasets. Similarly, AP is defined as

$$\mathrm{AP}(t)=\frac{\mathrm{TP}(t)}{\mathrm{TP}(t)+\mathrm{FP}(t)},$$

where t denotes the threshold, and AP.5, AP.75 denote the threshold (t) at 0.5, 0.75, respectively. Additionally, FP denotes a false positive, that is, a prediction error (the algorithm predicts a non-existent object). TP denotes a true positive, which means that the prediction is correct (the algorithm predicts the object within the specified range). Furthermore, AP(M) and AP(L) represent the recall ratios of medium and large objects in the detection datasets, respectively.

Table 2. Comparison between the different configurations of the SIS operator on object detection.

Backboneversion	Basic SIS	Interactive SIS		Recurrent SIS		COCO		VOC07
		I	II	I	II	AP	AP.5	AP
ResNet-50						29.8	49.7	71.5
Res2Net-50						31.2	51.4	73.3
SIS-50 variant					✔	31.2	51.4	73.3
				✔		31.2	51.4	73.4
			✔			31.4	51.6	73.6
		✔				31.5	51.7	73.7
	✔					32.4	52.7	74.1

Table 3. Comparison between the different configurations of SIS operator on semantic segmentation.

Backboneversion	Basic SIS	Interactive SIS		Recurrent SIS		Mean IoU	
		I	II	I	II	50 variant	101 variant
ResNet						0.751	0.770
Res2Net						0.767	0.781
SIS variant				✔		0.769	0.773
					✔	0.770	0.775
		✔				0.772	0.785
			✔			0.773	0.786
	✔					0.775	0.791

Table 4. Comparison between the different configurations of SIS operator on key points estimation on MS COCO dataset by average precision and recall measurement.

Backboneversion	Basic SIS	Interactive SIS		Recurrent SIS		AR.5	AR.75	AR(M)	AR(L)	AR	AP(M)	AP(L)	AP.5	AP.75	AP
		I	II	I	II										
ResNet-50						0.711	0.915	0.784	0.685	0.753	0.743	0.923	0.806	0.711	0.792
Res2Net-50						0.721	0.916	0.802	0.696	0.761	0.752	0.929	0.821	0.723	0.797
				✔		0.723	0.916	0.803	0.697	0.765	0.754	0.929	0.823	0.725	0.800
					✔	0.724	0.917	0.804	0.698	0.767	0.756	0.929	0.824	0.726	0.801
SIS-50 variant		✔				0.726	0.916	0.805	0.701	0.767	0.757	0.929	0.827	0.727	0.803
			✔			0.727	0.925	0.804	0.700	0.767	0.758	0.931	0.825	0.726	0.805
	✔					0.731	0.925	0.814	0.706	0.772	0.761	0.932	0.831	0.731	0.808
Res2Net-101						0.722	0.919	0.794	0.700	0.754	0.760	0.931	0.828	0.733	0.803
				✔		0.733	0.923	0.815	0.709	0.776	0.764	0.932	0.834	0.735	0.811
					✔	0.735	0.924	0.817	0.715	0.779	0.767	0.933	0.836	0.736	0.813
SIS-101variant		✔				0.737	0.924	0.818	0.715	0.785	0.769	0.934	0.838	0.738	0.815
			✔			0.738	0.925	0.819	0.715	0.785	0.770	0.935	0.840	0.739	0.816
	✔					0.742	0.925	0.825	0.715	0.785	0.773	0.935	0.843	0.743	0.819

5 Conclusions

In this work, we propose a compact and representative scale-in-scale convolution operator, referred to as SIS, by incorporating an efficient progressive multi-scale design into a traditional convolution operator to maximize entire-channel computation. Moreover, the improved variants are weight shared and incorporate split-and-interaction and recurrent-fuse mechanisms. The suggested SIS series is easily pluggable into any promising convolutional backbone, such as the well-known ResNet and Res2Net. Furthermore, we implemented the proposed SIS operator series in 29-layer, 50-layer, and 101-layer ResNet and Res2Net architecture variations. In conclusion, the experimental results reveal that the proposed SIS operator series is more effective and generalizable than existing state-of-the-art approaches with fewer parameters for numerous computer vision tasks.

Acknowledgements

This work was supported in part by the USTC Research Funds of the Double First-Class Initiative (YD2100002003, YD2100002004).

Conflict of interest

The authors declare that they have no conflict of interest.

Biographies

Man Zhou is currently working toward the PhD degree in University of Science and Technology of China, Hefei, China. His research interests include image/video processing, computer vision.

Xueyang Fu received the PhD degree in signal and information processing from Xiamen University, Xiamen, China, in 2018. He was a Visiting Scholar with Columbia University, New York, USA, sponsored by the China Scholarship Council, from 2016 to 2017. He is currently an Associate Researcher with the Department of Automation, University of Science and Technology of China, Hefei, China. His research interests include machine learning and image processing.

References

[1] Wang Q, Chen W, Wu X, et al. Detail preserving multi-scale exposure fusion. In: 2018 25th IEEE International Conference on Image Processing (ICIP). Athens, Greece: IEEE, 2018: 1713-1717.

[2] Wang B, Lei Y, Li N, et al. Multi-scale convolutional attention network for predicting remaining useful life of machinery. *IEEE Transactions on Industrial Electronics,* **2021**, *68* (8): 7496–7504.

[3] Yu J, Xie H, Xie G, et al. Multi-scale densely U-Nets refine network for face alignment. In: 2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). Shanghai, China: IEEE, 2019: 691–694.

[4] Zhang X, Zhang W. Application of new multi-scale edge fusion algorithm in structural edge extraction of aluminum foam. *IEEE Access,* **2020**, *8*: 15502–15517.

[5] Liu G, Wang C, Hu Y. RPN with the attention-based multi-scale method and the adaptive non-maximum suppression for billboard detection. In: 4th International Conference on Computer and Communications. Chengdu, China: IEEE, 2018: 2018.8780907.

[6] Sun K, Xiao B, Liu D, et al. Deep high-resolution representation learning for human pose estimation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA: IEEE, 2019: 5686–5696.

[7] Fang X, Yan P. Multi-organ segmentation over partially labeled datasets with multi-scale feature abstraction. *IEEE Transactions on Medical Imaging,* **2020**, *39* (11): 3619–3629.

[8] Huang G, Chen D, Li T, et al. Multi-scale dense networks for resource efficient image classification. In: 6th International Conference on Learning Representations. Vancouver, Canada: ICLR, 2018.

[9] Moukari M, Picard S, Simon L, et al. Deep multi-scale architectures for monocular depth estimation. In: 2018 25th IEEE International Conference on Image Processing. Athens, Greece: IEEE, 2018: 2940–2944.

[10] Papyan V, Elad M. Multi-scale patch-based image restoration. *IEEE Transactions on Image Processing,* **2016**, *25* (1): 249–261.

[11] Li J, Fang F, Li J, et al. MDCN: Multi-scale dense cross network for image super-resolution. *IEEE Transactions on Circuits and Systems*

for Video Technology, **2021**, *31* (7): 2547–2561.

[12] Lowe D G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision,* **2004**, *60* (2): 91–110.

[13] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA: IEEE, 2015: 1–9.

[14] Gao S, Cheng M. M, Zhao K, et al. Res2Net: A new multi-scale backbone architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **2019**, *43* (2): 652–662.

[15] Krizhevsky A, Sutskever I, Hinton G. E. ImageNet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems: Volume 1. Red Hook, NY: Curran Associates Inc, 2012: 1097–1105.

[16] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. https://arxiv.org/abs/1409.1556.

[17] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV: IEEE, 2016: 770-778.

[18] Xie S, Girshick R, Dollár P, et al. Aggregated residual transformations for deep neural networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI: IEEE, 2017: 5987–5995.

[19] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI: IEEE, 2017: 2261–2269.

[20] Iandola F N, Han S, Moskewicz M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. In: 5th International Conference on Learning Representations. Toulon, France: ICLR, 2017.

[21] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. https://arxiv.org/abs/1704.04861.

[22] Zhang X, Zhou X, Lin M, et al. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT: IEEE, 2018: 6848–6856.

[23] Lin M, Chen Q, Yan S. Network in network. In: ICLR 2014 Conference.Banff, Canada: ICLR, 2014.

[24] Sun S, Pang J, Shi J, et al. FishNet: A versatile backbone for image, region, and pixel level prediction. Advances in Neural Information Processing Systems 31 (NeurIPS 2018). Montréal, Canada: NeurIPS, 2018: 754–764.

[25] Yu F, Wang D, Shelhamer E, et al. Deep layer aggregation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT: IEEE, 2018: 2403–2412.

[26] Liao R, Zhao Z, Urtasun R, et al. LanczosNet: Multi-scale deep graph convolutional networks. In: Seventh International Conference on Learning Representations. New Orleans, LA: ICLR, 2019.

[27] Liu Y, Wang Y, Wang S, et al. CBNet: A novel composite backbone network architecture for object detection. *Proceedings of the AAAI Conference on Artificial Intelligence,* **2020**, *34* (7): 11653–11660.

[28] Chen L, Zhu Y, Papandreou G, et al. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Computer Vision–ECCV 2018. Cham, Switzerland: Springer, 2018: 833-851.

[29] Xiao B, Wu H, Wei Y. Simple baselines for human pose estimation and tracking. In: Computer Vision–ECCV 2018. Cham, Switzerland: Springer, 2018: 472–487.

DOI: 10.52396/JUSTC-2021-0188