

# Res2Net: A New Multi-scale Backbone Architecture

Shang-Hua Gao\*, Ming-Ming Cheng\*, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr

**Abstract**—Representing features at multiple scales is of great importance for numerous vision tasks. Recent advances in backbone convolutional neural networks (CNNs) continually demonstrate stronger multi-scale representation ability, leading to consistent performance gains on a wide range of applications. However, most existing methods represent the multi-scale features in a layer-wise manner. In this paper, we propose a novel building block for CNNs, namely Res2Net, by constructing hierarchical residual-like connections within one single residual block. The Res2Net represents multi-scale features at a granular level and increases the range of receptive fields for each network layer. The proposed Res2Net block can be plugged into the state-of-the-art backbone CNN models, *e.g.*, ResNet, ResNeXt, and DLA. We evaluate the Res2Net block on all these models and demonstrate consistent performance gains over baseline models on widely-used datasets, *e.g.*, CIFAR-100 and ImageNet. Further ablation studies and experimental results on representative computer vision tasks, *i.e.*, object detection, class activation mapping, and salient object detection, further verify the superiority of the Res2Net over the state-of-the-art baseline methods. The source code and trained models are available on <https://mmcheng.net/res2net/>.

**Index Terms**—Multi-scale, deep learning.

## INTRODUCTION

VISUAL patterns occur at multi-scales in natural scenes as shown in Fig. 1. First, objects may appear with different sizes in a single image, *e.g.*, the sofa and cup are of different sizes. Second, the essential contextual information of an object may occupy a much larger area than the object itself. For instance, we need to rely on the big table as context to better tell whether the small black blob placed on it is a cup or a pen holder. Third, perceiving information from different scales is essential for understanding parts as well as objects for tasks such as fine-grained classification and semantic segmentation. Thus, it is of critical importance to design good features for multi-scale stimuli for visual cognition tasks, including image classification [28], object detection [43], attention prediction [45], target tracking [63], action recognition [46], semantic segmentation [6], salient object detection [2], [24], object proposal [12], [43], skeleton extraction [67], stereo matching [42], and edge detection [37], [57].

Unsurprisingly, multi-scale features have been widely used in both conventional feature design [1], [39] and deep learning [10], [51]. Obtaining multi-scale representations in vision tasks requires feature extractors to use a large range of receptive fields to describe objects/parts/context at different scales. Convolutional neural networks (CNNs) naturally learn coarse-to-fine multi-scale features through a stack of convolutional operators. Such inherent multi-scale feature extraction ability of CNNs leads to effective representations for solving numerous vision tasks. How to design a more efficient network architecture is the key to further improving the performance of CNNs.

In the past few years, several backbone networks, *e.g.*, [10],

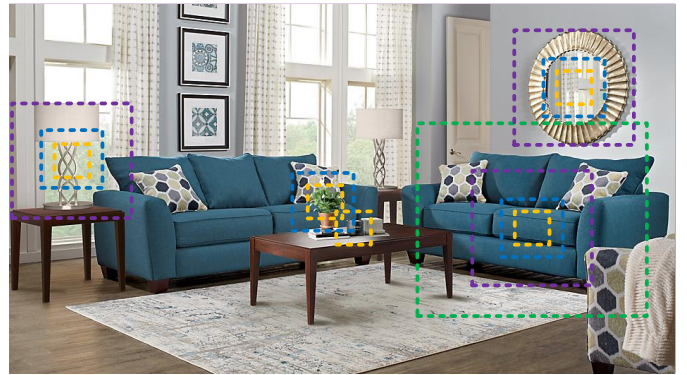


Fig. 1: Multi-scale representations are essential for various vision tasks, such as perceiving boundaries, regions, and semantic categories of the target objects. Even for the simplest recognition tasks, perceiving information from very different scales is essential to understand parts, objects (*e.g.*, sofa, table, and cup in this example), and their surrounding context (*e.g.*, ‘on the table’ context contributes to recognizing the black blob).

[15], [23], [25], [26], [28], [47], [51], [56], [60], have made significant advances in numerous vision tasks with state-of-the-art performance. Earlier architectures such as AlexNet [28] and VGGNet [47] stack convolutional operators, making the data-driven learning of multi-scale features feasible. The efficiency of multi-scale ability was subsequently improved by using conv layers with different kernel size (*e.g.*, InceptionNets [50], [51], [52]), residual modules (*e.g.*, ResNet [23]), shortcut connections (*e.g.*, DenseNet [26]), and hierarchical layer aggregation (*e.g.*, DLA [60]). The advances in backbone CNN architectures have demonstrated a trend towards more effective and efficient multi-scale representations.

In this work, we propose a simple yet efficient multi-scale processing approach. Unlike most existing methods that enhance

\*Equal contribution

- S.H. Gao, M.M. Cheng, K. Zhao, and X.Y. Zhang are with the TKLNDST, College of Computer Science, Nankai University, Tianjin 300350, China.
- M.H. Yang is with UC Merced.
- P. Torr is with Oxford University.
- M.M. Cheng is the corresponding author (cmm@nankai.edu.cn).

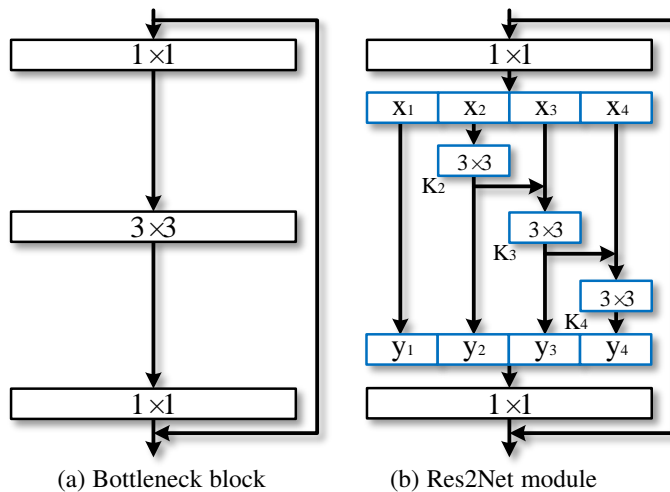


Fig. 2: Comparison between the bottleneck block and the proposed Res2Net module (the scale dimension  $s = 4$ ).

the layer-wise multi-scale representation strength of CNNs, we improve the multi-scale representation ability at a more granular level. Different from some concurrent works [5], [9], [11] that improve the multi-scale ability by utilizing features with different resolutions, the multi-scale of our proposed method refers to the multiple available receptive fields at a more granular level. To achieve this goal, we replace the  $3 \times 3$  filters<sup>1</sup> of  $n$  channels, with a set of smaller filter groups, each with  $w$  channels (without loss of generality we use  $n = s \times w$ ). As shown in Fig. 2, these smaller filter groups are connected in a hierarchical residual-like style to increase the number of scales that the output features can represent. Specifically, we divide input feature maps into several groups. A group of filters first extracts features from a group of input feature maps. Output features of the previous group are then sent to the next group of filters along with another group of input feature maps. This process repeats several times until all input feature maps are processed. Finally, feature maps from all groups are concatenated and sent to another group of  $1 \times 1$  filters to fuse information altogether. Along with any possible path in which input features are transformed to output features, the equivalent receptive field increases whenever it passes a  $3 \times 3$  filter, resulting in many equivalent feature scales due to combination effects.

The Res2Net strategy exposes a **new dimension**, namely *scale* (the number of feature groups in the Res2Net block), as an essential factor in addition to existing dimensions of depth [47], width<sup>2</sup>, and cardinality [56]. We state in Sec. 4.4 that increasing *scale* is more effective than increasing other dimensions.

Note that the proposed approach exploits the multi-scale potential at a more granular level, which is orthogonal to existing methods that utilize layer-wise operations. Thus, the proposed building block, namely *Res2Net module*, can be easily plugged into many existing CNN architectures. Extensive experimental results show that the *Res2Net module* can further improve the performance of state-of-the-art CNNs, *e.g.*, ResNet [23], ResNeXt [56], and DLA [60].

## 2 RELATED WORK

### 2.1 Backbone Networks

Recent years have witnessed numerous backbone networks [15], [23], [26], [28], [47], [51], [56], [60], achieving state-of-the-art performance in various vision tasks with stronger multi-scale representations. As designed, CNNs are equipped with basic multi-scale feature representation ability since the input information follows a fine-to-coarse fashion. The AlexNet [28] stacks filters sequentially and achieves significant performance gain over traditional methods for visual recognition. However, due to the limited network depth and kernel size of filters, the AlexNet has only a relatively small receptive field. The VGGNet [47] increases the network depth and uses filters with smaller kernel size. A deeper structure can expand the receptive fields, which is useful for extracting features from a larger scale. It is more efficient to enlarge the receptive field by stacking more layers than using large kernels. As such, the VGGNet provides a stronger multi-scale representation model than AlexNet, with fewer parameters. However, both AlexNet and VGGNet stack filters directly, which means each feature layer has a relatively fixed receptive field.

Network in Network (NIN) [31] inserts multi-layer perceptrons as micro-networks into the large network to enhance model discriminability for local patches within the receptive field. The  $1 \times 1$  convolution introduced in NIN has been a popular module to fuse features. The GoogLeNet [51] utilizes parallel filters with different kernel sizes to enhance the multi-scale representation capability. However, such capability is often limited by the computational constraints due to its limited parameter efficiency. The Inception Nets [50], [52] stack more filters in each path of the parallel paths in the GoogLeNet to further expand the receptive field. On the other hand, the ResNet [23] introduces short connections to neural networks, thereby alleviating the gradient vanishing problem while obtaining much deeper network structures. During the feature extraction procedure, short connections allow different combinations of convolutional operators, resulting in a large number of equivalent feature scales. Similarly, densely connected layers in the DenseNet [26] enable the network to process objects in a very wide range of scales. DPN [10] combines the ResNet with DenseNet to enable feature re-usage ability of ResNet and the feature exploration ability of DenseNet. The recently proposed DLA [60] method combines layers in a tree structure. The hierarchical tree structure enables the network to obtain even stronger layer-wise multi-scale representation capability.

### 2.2 Multi-scale Representations for Vision Tasks

Multi-scale feature representations of CNNs are of great importance to a number of vision tasks including object detection [43], face analysis [4], [41], edge detection [37], semantic segmentation [6], salient object detection [34], [65], and skeleton detection [67], boosting the model performance of those fields.

#### 2.2.1 Object detection.

Effective CNN models need to locate objects of different scales in a scene. Earlier works such as the R-CNN [18] mainly rely on the backbone network, *i.e.*, VGGNet [47], to extract features of multiple scales. He *et al.* propose an SPP-Net approach [22] that utilizes spatial pyramid pooling after the backbone network to enhance the multi-scale ability. The Faster R-CNN method [43] further proposes the region proposal networks to generate bounding boxes with various scales. Based on the Faster R-CNN, the

1. Convolutional operators and filters are used interchangeably.  
2. Width refers to the number of channels in a layer as in [61].

FPN [32] approach introduces feature pyramid to extract features with different scales from a single image. The SSD method [36] utilizes feature maps from different stages to process visual information at different scales.

### 2.2.2 Semantic segmentation.

Extracting essential contextual information of objects requires CNN models to process features at various scales for effective semantic segmentation. Long *et al.* [38] propose one of the earliest methods that enables multi-scale representations of the fully convolutional network (FCN) for semantic segmentation task. In DeepLab, Chen *et al.* [6], [7] introduces cascaded atrous convolutional module to expand the receptive field further while preserving spatial resolutions. More recently, global context information is aggregated from region-based features via the pyramid pooling scheme in the PSPNet [64].

### 2.2.3 Salient object detection.

Precisely locating the salient object regions in an image requires an understanding of both large-scale context information for the determination of object saliency, and small-scale features to localize object boundaries accurately [66]. Early approaches [3] utilize handcrafted representations of global contrast [13] or multi-scale region features [53]. Li *et al.* [29] propose one of the earliest methods that enables multi-scale deep features for salient object detection. Later, multi-context deep learning [68] and multi-level convolutional features [62] are proposed for improving salient object detection. More recently, Hou *et al.* [24] introduce dense short connections among stages to provide rich multi-scale feature maps at each layer for salient object detection.

## 2.3 Concurrent Works

Recently, there are some concurrent works aiming at improving the performance by utilizing the multi-scale features [5], [9], [11], [49]. Big-Little Net [5] is a multi-branch network composed of branches with different computational complexity. Octave Conv [9] decomposes the standard convolution into two resolutions to process features at different frequencies. MSNet [11] utilizes a high-resolution network to learn high-frequency residuals by using the up-sampled low-resolution features learned by a low-resolution network. Other than the low-resolution representations in current works, the HRNet [48], [49] introduces high-resolution representations in the network and repeatedly performs multi-scale fusions to strengthen high-resolution representations. One common operation in [5], [9], [11], [48], [49] is that they all use pooling or up-sample to re-size the feature map to  $2^n$  times of the original scale to save the computational budget while maintaining or even improving performance. While in the Res2Net block, the hierarchical residual-like connections within a single residual block module enable the variation of receptive fields at a more granular level to capture details and global features. Experimental results show that Res2Net module can be integrated with those novel network designs to further boost the performance.

## 3 RES2NET

### 3.1 Res2Net Module

The bottleneck structure shown in Fig. 2(a) is a basic building block in many modern backbone CNNs architectures,

e.g., ResNet [23], ResNeXt [56], and DLA [60]. Instead of extracting features using a group of  $3 \times 3$  filters as in the bottleneck block, we seek alternative architectures with stronger multi-scale feature extraction ability, while maintaining a similar computational load. Specifically, we replace a group of  $3 \times 3$  filters with smaller groups of filters, while connecting different filter groups in a hierarchical residual-like style. Since our proposed neural network module involves residual-like connections within a single residual block, we name it *Res2Net*.

algorithm details

Fig. 2 shows the differences between the bottleneck block and the proposed Res2Net module. After the  $1 \times 1$  convolution, we evenly split the feature maps into  $s$  feature map subsets, denoted by  $\mathbf{x}_i$ , where  $i \in \{1, 2, \dots, s\}$ . Each feature subset  $\mathbf{x}_i$  has the same spatial size but  $1/s$  number of channels compared with the input feature map. Except for  $\mathbf{x}_1$ , each  $\mathbf{x}_i$  has a corresponding  $3 \times 3$  convolution, denoted by  $\mathbf{K}_i()$ . We denote by  $\mathbf{y}_i$  the output of  $\mathbf{K}_i()$ . The feature subset  $\mathbf{x}_i$  is added with the output of  $\mathbf{K}_{i-1}()$ , and then fed into  $\mathbf{K}_i()$ . To reduce parameters while increasing  $s$ , we omit the  $3 \times 3$  convolution for  $\mathbf{x}_1$ . Thus,  $\mathbf{y}_i$  can be written as:

$$\mathbf{y}_i = \begin{cases} \mathbf{x}_i & i = 1; \\ \mathbf{K}_i(\mathbf{x}_i) & i = 2; \\ \mathbf{K}_i(\mathbf{x}_i + \mathbf{y}_{i-1}) & 2 < i \leq s. \end{cases} \quad (1)$$

Notice that each  $3 \times 3$  convolutional operator  $\mathbf{K}_i()$  could potentially receive feature information from all feature splits  $\{\mathbf{x}_j, j \leq i\}$ . Each time a feature split  $\mathbf{x}_j$  goes through a  $3 \times 3$  convolutional operator, the output result can have a larger receptive field than  $\mathbf{x}_j$ . Due to the combinatorial explosion effect, the output of the Res2Net module contains a different number and different combination of receptive field sizes/scales.

In the Res2Net module, splits are processed in a multi-scale fashion, which is conducive to the extraction of both global and local information. To better fuse information at different scales, we concatenate all splits and pass them through a  $1 \times 1$  convolution. The split and concatenation strategy can enforce convolutions to process features more effectively. To reduce the number of parameters, we omit the convolution for the first split, which can also be regarded as a form of feature reuse.

In this work, we use  $s$  as a control parameter of the *scale* dimension. Larger  $s$  potentially allows features with richer receptive field sizes to be learnt, with negligible computational/memory overheads introduced by concatenation.

### 3.2 Integration with Modern Modules

Numerous neural network modules have been proposed in recent years, including cardinality dimension introduced by Xie *et al.* [56], as well as squeeze and excitation (SE) block presented by Hu *et al.* [25]. The proposed Res2Net module introduces the scale dimension that is orthogonal to these improvements. As shown in Fig. 3, we can easily integrate the cardinality dimension [56] and the SE block [25] with the proposed Res2Net module.

#### 3.2.1 Dimension cardinality.

The dimension cardinality indicates the number of groups within a filter [56]. This dimension changes filters from single-branch to multi-branch and improves the representation ability of a CNN model. In our design, we can replace the  $3 \times 3$  convolution with the  $3 \times 3$  group convolution, where  $c$  indicates the number of groups. Experimental comparisons between the scale dimension and cardinality are presented in Sec. 4.2 and Sec. 4.4.



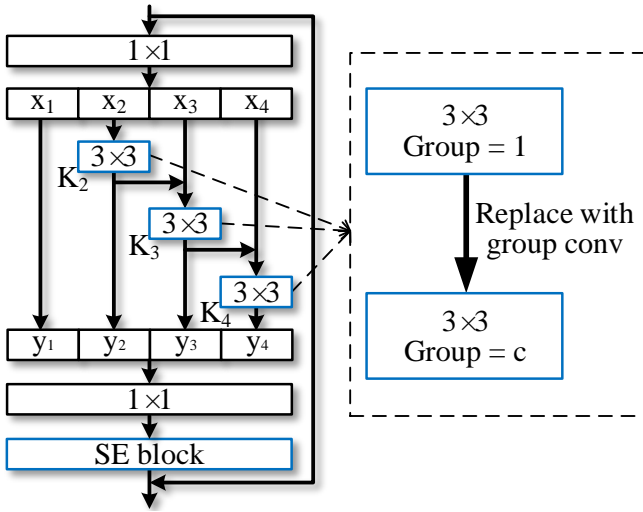


Fig. 3: The Res2Net module can be integrated with the dimension cardinality [56] (replace conv with group conv) and SE [25] blocks.

### 3.2.2 SE block.

A SE block adaptively re-calibrates channel-wise feature responses by explicitly modelling inter-dependencies among channels [25]. Similar to [25], we add the SE block right before the residual connections of the Res2Net module. Our Res2Net module can benefit from the integration of the SE block, which we have experimentally demonstrated in Sec. 4.2 and Sec. 4.3.

### 3.3 Integrated Models

Since the proposed Res2Net module does not have specific requirements of the overall network structure and the multi-scale representation ability of the Res2Net module is orthogonal to the layer-wise feature aggregation models of CNNs, we can easily integrate the proposed Res2Net module into the state-of-the-art models, such as ResNet [23], ResNeXt [56], DLA [60] and Big-Little Net [5]. The corresponding models are referred to as Res2Net, Res2NeXt, Res2Net-DLA, and bLRes2Net-50, respectively.

The proposed scale dimension is orthogonal to the cardinality [56] dimension and width [23] dimension of prior work. Thus, after the *scale* is set, we adjust the value of cardinality and width to maintain the overall model complexity similar to its counterparts. We do not focus on reducing the model size in this work since it requires more meticulous designs such as depth-wise separable convolution [40], model pruning [19], and model compression [14].

For experiments on the ImageNet [44] dataset, we mainly use the ResNet-50 [23], ResNeXt-50 [56], DLA-60 [60], and bLResNet-50 [5] as our baseline models. The complexity of the proposed model is approximately equal to those of the baseline models, whose number of parameters is around  $25M$  and the number of FLOPs for an image of  $224 \times 224$  pixels is around  $4.2G$  for 50-layer networks. For experiments on the CIFAR [27] dataset, we use the ResNeXt-29,  $8c \times 64w$  [56] as our baseline model. Empirical evaluations and discussions of the proposed models with respect to model complexity are presented in Sec. 4.4.

TABLE 1: Top-1 and Top-5 test error on the ImageNet dataset.

|                  | top-1 err. (%) | top-5 err. (%) |
|------------------|----------------|----------------|
| ResNet-50 [23]   | 23.85          | 7.13           |
| Res2Net-50       | <b>22.01</b>   | <b>6.15</b>    |
| InceptionV3 [52] | 22.55          | 6.44           |
| Res2Net-50-299   | <b>21.41</b>   | <b>5.88</b>    |
| ResNeXt-50 [56]  | 22.61          | 6.50           |
| Res2NeXt-50      | <b>21.76</b>   | <b>6.09</b>    |
| DLA-60 [60]      | 23.32          | 6.60           |
| Res2Net-DLA-60   | <b>21.53</b>   | <b>5.80</b>    |
| DLA-X-60 [60]    | 22.19          | 6.13           |
| Res2NeXt-DLA-60  | <b>21.55</b>   | <b>5.86</b>    |
| SENet-50 [25]    | 23.24          | 6.69           |
| SE-Res2Net-50    | <b>21.56</b>   | <b>5.94</b>    |
| bLResNet-50 [5]  | 22.41          | -              |
| bLRes2Net-50     | <b>21.68</b>   | 6.00           |

## 4 EXPERIMENTS

### 4.1 Implementation Details

We implement the proposed models using the Pytorch framework. For fair comparisons, we use the Pytorch implementation of ResNet [23], ResNeXt [56], DLA [60] as well as bLResNet-50 [5], and only replace the original bottleneck block with the proposed Res2Net module. Similar to prior work, on the ImageNet dataset [44], each image is of  $224 \times 224$  pixels randomly cropped from a re-sized image. We use the same data argumentation strategy as [23], [52]. Similar to [23], we train the network using SGD with weight decay 0.0001, momentum 0.9, and a mini-batch of 256 on 4 Titan Xp GPUs. The learning rate is initially set to 0.1 and divided by 10 every 30 epochs.

All models for the ImageNet, including the baseline and proposed models, are trained for 100 epochs with the same training and data argumentation strategy. For testing, we use the same image cropping method as [23]. On the CIFAR dataset, we use the implementation of ResNeXt-29 [56]. For all tasks, we use the original implementations of baselines and only replace the backbone model with the proposed Res2Net.

### 4.2 ImageNet

We conduct experiments on the ImageNet dataset [44], which contains 1.28 million training images and 50k validation images from 1000 classes. We construct the models with approximate 50 layers for performance evaluation against the state-of-the-art methods. More ablation studies are conducted on the CIFAR dataset.

#### 4.2.1 Performance gain.

Table 1 shows the top-1 and top-5 test error on the ImageNet dataset. For simplicity, all Res2Net models in Table 1 have the scale  $s = 4$ . The Res2Net-50 has an improvement of 1.84% on top-1 error over the ResNet-50. The Res2NeXt-50 achieves a 0.85% improvement in terms of top-1 error over the ResNeXt-50. Also, the Res2Net-DLA-60 outperforms the DLA-60 by 1.27% in terms of top-1 error. The Res2NeXt-DLA-60 outperforms the DLA-X-60 by 0.64% in terms of top-1 error. The SE-Res2Net-50 has an improvement of 1.68% over the SENet-50. bLRes2Net-50 has an improvement of 0.73% in terms of top-1 error over the

TABLE 2: Top-1 and Top-5 test error (%) of deeper networks on the ImageNet dataset.

|                   | top-1 err.   | top-5 err.  |
|-------------------|--------------|-------------|
| DenseNet-161 [26] | 22.35        | 6.20        |
| ResNet-101 [23]   | 22.63        | 6.44        |
| Res2Net-101       | <b>20.81</b> | <b>5.57</b> |

bLResNet-50. The Res2Net module further enhances the multi-scale ability of bLResNet at a granular level even bLResNet is designed to utilize features with different scales as discussed in Sec. 2.3. Note that the ResNet [23], ResNeXt [56], SE-Net [25], bLResNet [5], and DLA [60] are the state-of-the-art CNN models. Compared with these strong baselines, models integrated with the Res2Net module still have consistent performance gains.

We also compare our method against the InceptionV3 [52] model, which utilizes parallel filters with different kernel combinations. For fair comparisons, we use the ResNet-50 [23] as the baseline model and train our model with the input image size of  $299 \times 299$  pixels, as used in the InceptionV3 model. The proposed Res2Net-50-299 outperforms InceptionV3 by 1.14% on top-1 error. We conclude that the hierarchical residual-like connection of the Res2Net module is more effective than the parallel filters of InceptionV3 when processing multi-scale information. While the combination pattern of filters in InceptionV3 is dedicatedly designed, the Res2Net module presents a simple but effective combination pattern.

#### 4.2.2 Going deeper with Res2Net.

Deeper networks have been shown to have stronger representation capability [23], [56] for vision tasks. To validate our model with greater depth, we compare the classification performance of the Res2Net and the ResNet, both with 101 layers. As shown in Table 2, the Res2Net-101 achieves significant performance gains over the ResNet-101 with 1.82% in terms of top-1 error. Note that the Res2Net-50 has the performance gain of 1.84% in terms of top-1 error over the ResNet-50. These results show that the proposed module with additional dimension scale can be integrated with deeper models to achieve better performance. We also compare our method with the DenseNet [26]. Compared with the DenseNet-161, the best performing model of the officially provided DenseNet family, the Res2Net-101 has an improvement of 1.54% in terms of top-1 error.

#### 4.2.3 Effectiveness of scale dimension.

To validate our proposed dimension scale, we experimentally analyze the effect of different scales. As shown in Table 3, the performance increases with the increase of scale. With the increase of scale, the Res2Net-50 with  $14w \times 8s$  achieves performance gains over the ResNet-50 with 1.99% in terms of top-1 error. Note that with the preserved complexity, the width of  $\mathbf{K}_i()$  decreases with the increase of scale. We further evaluate the performance gain of increasing scale with increased model complexity. The Res2Net-50 with  $26w \times 8s$  achieves significant performance gains over the ResNet-50 with 3.05% in terms of top-1 error. A Res2Net-50 with  $18w \times 4s$  also outperforms the ResNet-50 by 0.93% in terms of top-1 error with only 69% FLOPs. Table 3 shows the Runtime under different scales, which is the average time to infer the ImageNet validation set with the size of  $224 \times 224$ . Although the feature splits  $\{\mathbf{y}_i\}$  need to be computed

TABLE 3: Top-1 and Top-5 test error (%) of Res2Net-50 with different scales on the ImageNet dataset. Parameter  $w$  is the width of filters, and  $s$  is the number of scale, as described in Equation (1).

|  | Setting         | FLOPs | Runtime | top-1 err. | top-5 err. |
|--|-----------------|-------|---------|------------|------------|
| ResNet-50                                | 64w             | 4.2G  | 149ms   | 23.85      | 7.13       |
| Res2Net-50<br>( Preserved<br>complexity) | $48w \times 2s$ | 4.2G  | 148ms   | 22.68      | 6.47       |
|  | $26w \times 4s$ | 4.2G  | 153ms   | 22.01      | 6.15       |
|  | $14w \times 8s$ | 4.2G  | 172ms   | 21.86      | 6.14       |
| Res2Net-50<br>( Increased<br>complexity) | $26w \times 4s$ | 4.2G  | -       | 22.01      | 6.15       |
|  | $26w \times 6s$ | 6.3G  | -       | 21.42      | 5.87       |
|  | $26w \times 8s$ | 8.3G  | -       | 20.80      | 5.63       |
| Res2Net-50-L                             | $18w \times 4s$ | 2.9G  | 106ms   | 22.92      | 6.67       |

TABLE 4: Top-1 test error (%) and model size on the CIFAR-100 dataset. Parameter  $c$  indicates the value of cardinality, and  $w$  is the width of filters.

|  | Params | top-1 err.   |
|--|--------|--------------|
| Wide ResNet [61]                           | 36.5M  | 20.50        |
| ResNeXt-29, $8c \times 64w$ [56] (base)    | 34.4M  | 17.90        |
| ResNeXt-29, $16c \times 64w$ [56]          | 68.1M  | 17.31        |
| DenseNet-BC ( $k = 40$ ) [26]              | 25.6M  | 17.18        |
| Res2NeXt-29, $6c \times 24w \times 4s$     | 24.3M  | 16.98        |
| Res2NeXt-29, $8c \times 25w \times 4s$     | 33.8M  | 16.93        |
| Res2NeXt-29, $6c \times 24w \times 6s$     | 36.7M  | <b>16.79</b> |
| ResNeXt-29, $8c \times 64w$ -SE [25]       | 35.1M  | 16.77        |
| Res2NeXt-29, $6c \times 24w \times 4s$ -SE | 26.0M  | 16.68        |
| Res2NeXt-29, $8c \times 25w \times 4s$ -SE | 34.0M  | 16.64        |
| Res2NeXt-29, $6c \times 24w \times 6s$ -SE | 36.9M  | <b>16.56</b> |

sequentially due to hierarchical connections, the extra run-time introduced by Res2Net module can often be ignored. Since the number of available tensors in a GPU is limited, there are typically sufficient parallel computations within a single GPU clock period for the typical setting of Res2Net, *i.e.*,  $s = 4$ .

### 4.3 CIFAR

We also conduct some experiments on the CIFAR-100 dataset [27], which contains 50k training images and 10k testing images from 100 classes. The *ResNeXt-29,  $8c \times 64w$*  [56] is used as the baseline model. We only replace the original basic block with our proposed Res2Net module while keeping other configurations unchanged. Table 4 shows the top-1 test error and model size on the CIFAR-100 dataset. Experimental results show that our method surpasses the baseline and other methods with fewer parameters. Our proposed *Res2NeXt-29,  $6c \times 24w \times 6s$*  outperforms the baseline by 1.11%. *Res2NeXt-29,  $6c \times 24w \times 4s$*  even outperforms the *ResNeXt-29,  $16c \times 64w$*  with only 35% parameters. We also achieve better performance with fewer parameters, compared with DenseNet-BC ( $k = 40$ ). Compared with *Res2NeXt-29,  $6c \times 24w \times 4s$* , *Res2NeXt-29,  $8c \times 25w \times 4s$*  achieves a better result with more width and cardinality, indicating that the dimension scale is orthogonal to dimension width and cardinality. We also integrate the recently proposed SE block into our structure. With fewer parameters, our method still outperforms the *ResNeXt-29,  $8c \times 64w$ -SE* baseline.

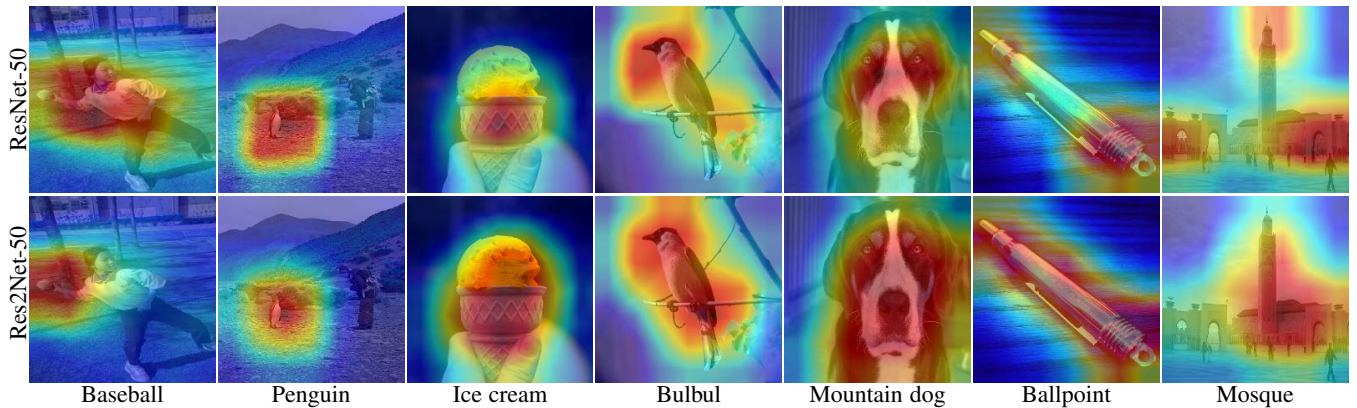


Fig. 4: Visualization of class activation mapping [45], using ResNet-50 and Res2Net-50 as backbone networks.

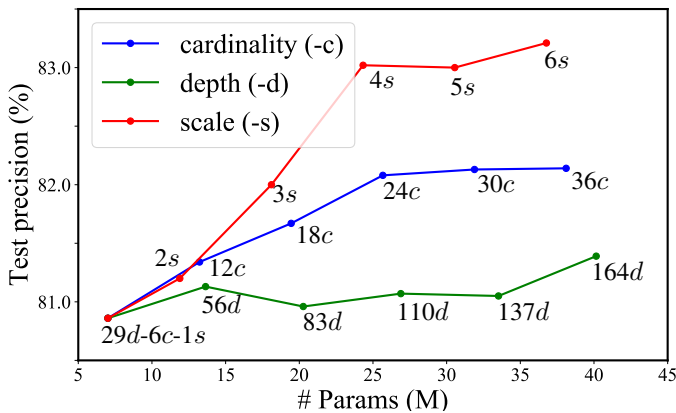


Fig. 5: Test precision on the CIFAR-100 dataset with regard to the model size, by changing cardinality (ResNeXt-29), depth (ResNeXt), and scale (Res2Net-29).

#### 4.4 Scale Variation

Similar to Xie *et al.* [56], we evaluate the test performance of the baseline model by increasing different CNN dimensions, including scale (Equation (1)), cardinality [56], and depth [47]. While increasing model capacity using one dimension, we fix all other dimensions. A series of networks are trained and evaluated under these changes. Since [56] has already shown that increasing cardinality is more effective than increasing width, we only compare the proposed dimension scale with cardinality and depth.

Fig. 5 shows the test precision on the CIFAR-100 dataset with regard to the model size. The depth, cardinality, and scale of the baseline model are 29, 6 and 1, respectively. Experimental results suggest that scale is an effective dimension to improve model performance, which is consistent with what we have observed on the ImageNet dataset in Sec. 4.2. Moreover, increasing scale is more effective than other dimensions, resulting in quicker performance gains. As described in Equation (1) and Fig. 2, for the case of scale  $s = 2$ , we only increase the model capacity by adding more parameters of  $1 \times 1$  filters. Thus, the model performance of  $s = 2$  is slightly worse than that of increasing cardinality. For  $s = 3, 4$ , the combination effects of our hierarchical residual-like structure produce a rich set of equivalent scales, resulting in significant performance gains. However, the models with scale 5 and 6 have limited performance gains, about which we assume that the image in the CIFAR dataset is too small ( $32 \times 32$ ) to have many scales.

TABLE 5: Object detection results on the PASCAL VOC07 and COCO datasets, measured using AP (%) and AP@IoU=0.5 (%). The Res2Net has similar complexity compared with its counterparts.

| Dataset | Backbone   | AP          | AP@IoU=0.5  |
|---------|------------|-------------|-------------|
| VOC07   | ResNet-50  | 72.1        | -           |
|         | Res2Net-50 | <b>74.4</b> | -           |
| COCO    | ResNet-50  | 31.1        | 51.4        |
|         | Res2Net-50 | <b>33.7</b> | <b>53.6</b> |

TABLE 6: Average Precision (AP) and Average Recall (AR) of object detection with different sizes on the COCO dataset.

|            |        | Object size |        |       |      |
|------------|--------|-------------|--------|-------|------|
|            |        | Small       | Medium | Large | All  |
| ResNet-50  | AP (%) | 13.5        | 35.4   | 46.2  | 31.1 |
| Res2Net-50 |        | 14.0        | 38.3   | 51.1  | 33.7 |
| Improve.   |        | +0.5        | +2.9   | +4.9  | +2.6 |
| ResNet-50  | AR (%) | 21.8        | 48.6   | 61.6  | 42.8 |
| Res2Net-50 |        | 23.2        | 51.1   | 65.3  | 45.0 |
| Improve.   |        | +1.4        | +2.5   | +3.7  | +2.2 |

#### 4.5 Class Activation Mapping

To understand the multi-scale ability of the Res2Net, we visualize the class activation mapping (CAM) using Grad-CAM [45], which is commonly used to localize the discriminative regions for image classification. In the visualization examples shown in Fig. 4, stronger CAM areas are covered with lighter colors. Compared with ResNet, the Res2Net based CAM results have more concentrated activation maps on small objects such as ‘baseball’ and ‘penguin’. Both methods have similar activation maps on the middle size objects, such as ‘ice cream’. Due to stronger multi-scale ability, the Res2Net has activation maps that tend to cover the whole object on big objects such as ‘bulbul’, ‘mountain dog’, ‘ballpoint’, and ‘mosque’, while activation maps of ResNet only cover parts of objects. Such ability of precisely localizing CAM region makes the Res2Net potentially valuable for object region mining in weakly supervised semantic segmentation tasks [54].

TABLE 7: Performance of semantic segmentation on PASCAL VOC12 val set using Res2Net-50 with different scales. The Res2Net has similar complexity compared with its counterparts.

| Backbone    | Setting | Mean IoU (%) |
|-------------|---------|--------------|
| ResNet-50   | 64w     | 77.7         |
| Res2Net-50  | 48w×2s  | 78.2         |
|             | 26w×4s  | <b>79.2</b>  |
|             | 18w×6s  | 79.1         |
|             | 14w×8s  | 79.0         |
| ResNet-101  | 64w     | 79.0         |
| Res2Net-101 | 26w×4s  | <b>80.2</b>  |

#### 4.6 Object Detection

For object detection task, we validate the Res2Net on the PASCAL VOC07 [17] and MS COCO [33] datasets, using Faster R-CNN [43] as the baseline method. We use the backbone network of ResNet-50 vs. Res2Net-50, and follow all other implementation details of [43] for a fair comparison. Table 5 shows the object detection results. On the PASCAL VOC07 dataset, the Res2Net-50 based model outperforms its counterparts by 2.3% on average precision (AP). On the COCO dataset, the Res2Net-50 based model outperforms its counterparts by 2.6% on AP, and 2.2% on AP@IoU=0.5.

We further test the AP and average recall (AR) scores for objects of different sizes as shown in Table 6. Objects are divided into three categories based on the size, according to [33]. The Res2Net based model has a large margin of improvement over its counterparts by 0.5%, 2.9%, and 4.9% on AP for small, medium, and large objects, respectively. The improvement of AR for small, medium, and large objects are 1.4%, 2.5%, and 3.7%, respectively. Due to the strong multi-scale ability, the Res2Net based models can cover a large range of receptive fields, boosting the performance on objects of different sizes.

#### 4.7 Semantic Segmentation

Semantic segmentation requires a strong multi-scale ability of CNNs to extract essential contextual information of objects. We thus evaluate the multi-scale ability of Res2Net on the semantic segmentation task using PASCAL VOC12 dataset [16]. We follow the previous work to use the augmented PASCAL VOC12 dataset [20] which contains 10582 training images and 1449 val images. We use the Deeplab v3+ [8] as our segmentation method. All implementations remain the same with Deeplab v3+ [8] except that the backbone network is replaced with ResNet and our proposed Res2Net. The output strides used in training and evaluation are both 16. As shown in Table 7, Res2Net-50 based method outperforms its counterpart by 1.5% on mean IoU. And Res2Net-101 based method outperforms its counterpart by 1.2% on mean IoU. Visual comparisons of semantic segmentation results on challenging examples are illustrated in Fig. 6. The Res2Net based method tends to segment all parts of objects regardless of object size.

#### 4.8 Instance Segmentation

Instance segmentation is the combination of object detection and semantic segmentation. It requires not only the correct detection of objects with various sizes in an image but also the precise segmentation of each object. As mentioned in Sec. 4.6 and Sec. 4.7,

TABLE 8: Performance of instance segmentation on the COCO dataset using Res2Net-50 with different scales. The Res2Net has similar complexity compared with its counterparts.

| Backbone   | Setting | AP          | AP <sub>50</sub> | AP <sub>75</sub> | AP <sub>S</sub> | AP <sub>M</sub> | AP <sub>L</sub> |
|------------|---------|-------------|------------------|------------------|-----------------|-----------------|-----------------|
| ResNet-50  | 64w     | 33.9        | 55.2             | 36.0             | 14.8            | 36.0            | 50.9            |
| Res2Net-50 | 48w×2s  | 34.2        | 55.6             | 36.3             | 14.9            | 36.8            | 50.9            |
|            | 26w×4s  | 35.6        | <b>57.6</b>      | 37.6             | <b>15.7</b>     | 37.9            | <b>53.7</b>     |
|            | 18w×6s  | <b>35.7</b> | 57.5             | <b>38.1</b>      | 15.4            | <b>38.1</b>     | 53.7            |
|            | 14w×8s  | 35.3        | 57.0             | 37.5             | 15.6            | 37.5            | 53.4            |

TABLE 9: Salient object detection results on different datasets, measured using F-measure and Mean Absolute Error (MAE). The Res2Net has similar complexity compared with its counterparts.

| Dataset   | Backbone   | F-measure↑   | MAE ↓        |
|-----------|------------|--------------|--------------|
| ECSSD     | ResNet-50  | 0.910        | 0.065        |
|           | Res2Net-50 | <b>0.926</b> | <b>0.056</b> |
| PASCAL-S  | ResNet-50  | 0.823        | 0.105        |
|           | Res2Net-50 | <b>0.841</b> | <b>0.099</b> |
| HKU-IS    | ResNet-50  | 0.894        | 0.058        |
|           | Res2Net-50 | <b>0.905</b> | <b>0.050</b> |
| DUT-OMRON | ResNet-50  | 0.748        | 0.092        |
|           | Res2Net-50 | <b>0.800</b> | <b>0.071</b> |

both object detection and semantic segmentation require a strong multi-scale ability of CNNs. Thus, the multi-scale representation is quite beneficial to instance segmentation. We use the Mask R-CNN [21] as the instance segmentation method, and replace the backbone network of ResNet-50 with our proposed Res2Net-50. The performance of instance segmentation on MS COCO [33] dataset is shown in Table 8. The Res2Net-26w×4s based method outperforms its counterparts by 1.7% on AP and 2.4% on AP<sub>50</sub>. The performance gains on objects with different sizes are also demonstrated. The improvement of AP for small, medium, and large objects are 0.9%, 1.9%, and 2.8%, respectively. Table 8 also shows the performance comparisons of Res2Net under the same complexity with different scales. The performance shows an overall upward trend with the increase of scale. Note that compared with the Res2Net-50-48w×2s, the Res2Net-50-26w×4s has an improvement of 2.8 % on AP<sub>L</sub>, while the Res2Net-50-48w×2s has the same AP<sub>L</sub> compared with ResNet-50. We assume that the performance gain on large objects is benefited from the extra scales. When the scale is relatively larger, the performance gain is not obvious. The Res2Net module is capable of learning a suitable range of receptive fields. The performance gain is limited when the scale of objects in the image is already covered by the available receptive fields in the Res2Net module. With fixed complexity, the increased scale results in fewer channels for each receptive field, which may reduce the ability to process features of a particular scale.

#### 4.9 Salient Object Detection

Pixel level tasks such as salient object detection also require the strong multi-scale ability of CNNs to locate both the holistic objects as well as their region details. Here we use the latest method DSS [24] as our baseline. For a fair comparison, we only replace the backbone with ResNet-50 and our proposed Res2Net-50, while keeping other configurations unchanged. Following [24],





Fig. 6: Visualization of semantic segmentation results [8], using ResNet-101 and Res2Net-101 as backbone networks.

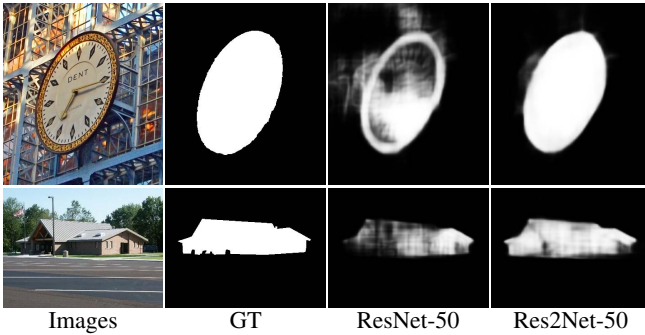


Fig. 7: Examples of salient object detection [24] results, using ResNet-50 and Res2Net-50 as backbone networks, respectively.

we train those two models using the MSRA-B dataset [35], and evaluate results on ECSSD [58], PASCAL-S [30], HKU-IS [29], and DUT-OMRON [59] datasets. The F-measure and Mean Absolute Error (MAE) are used for evaluation. As shown in Table 9, the Res2Net based model has a consistent improvement compared with its counterparts on all datasets. On the DUT-OMRON dataset (containing 5168 images), the Res2Net based model has a 5.2% improvement on F-measure and a 2.1% improvement on MAE, compared with ResNet based model. The Res2Net based approach achieves greatest performance gain on the DUT-OMRON dataset, since this dataset contains the most significant object size variation compared with the other three datasets. Some visual comparisons of salient object detection results on challenging examples are illustrated in Fig. 7.

#### 4.10 Key-points Estimation

Human parts are of different sizes, which requires the key-points estimation method to locate human key-points with different scales. To verify whether the multi-scale representation ability of Res2Net can benefit the task of key-points estimation, we use the SimpleBaseline [55] as the key-points estimation method and only replace the backbone with the proposed Res2Net. All implementations including the training and testing strategies remain the same with the SimpleBaseline [55]. We train the model using the COCO key-point detection dataset [33], and evaluate the model

TABLE 10: Performance of key-points estimation on the COCO validation set. The Res2Net has similar complexity compared with its counterparts.

| Backbone    | $AP$ | $AP_{50}$ | $AP_{75}$ | $AP_M$ | $AP_L$ |
|-------------|------|-----------|-----------|--------|--------|
| ResNet-50   | 70.4 | 88.6      | 78.3      | 67.1   | 77.2   |
| Res2Net-50  | 73.7 | 92.5      | 81.4      | 70.8   | 78.2   |
| ResNet-101  | 71.4 | 89.3      | 79.3      | 68.1   | 78.1   |
| Res2Net-101 | 74.4 | 92.6      | 82.6      | 72.0   | 78.5   |

using the COCO validation set. Following common settings, we use the same person detectors in SimpleBaseline [55] for evaluation. Table 10 shows the performance of key-points estimation on the COCO validation set using Res2Net. The Res2Net-50 and Res2Net-101 based models outperform baselines on  $AP$  by 3.3% and 3.0%, respectively. Also, Res2Net based models have considerable performance gains on human with different scales compared with baselines.

## 5 CONCLUSION AND FUTURE WORK

We present a simple yet efficient block, namely Res2Net, to further explore the multi-scale ability of CNNs at a more granular level. The Res2Net exposes a new dimension, namely “scale”, which is an essential and more effective factor in addition to existing dimensions of depth, width, and cardinality. Our Res2Net module can be integrated with existing state-of-the-art methods with no effort. Image classification results on CIFAR-100 and ImageNet benchmarks suggested that our new backbone network consistently performs favourably against its state-of-the-art competitors, including ResNet, ResNeXt, DLA, *etc.*

Although the superiority of the proposed backbone model has been demonstrated in the context of several representative computer vision tasks, including class activation mapping, object detection, and salient object detection, we believe multi-scale representation is essential for a much wider range of application areas. To encourage future works to leverage the strong multi-scale ability of the Res2Net, the source code is available on <https://mmcheng.net/res2net/>.



## ACKNOWLEDGMENTS

This research was supported by NSFC (NO. 61620106008, 61572264), the national youth talent support program, and Tianjin Natural Science Foundation (17JCJQJC43700, 18ZXZNGX00110).

## REFERENCES

- [1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.
- [2] A. Borji, M.-M. Cheng, Q. Hou, H. Jiang, and J. Li. Salient object detection: A survey. *Computational Visual Media*, 5(2):117–150, 2019.
- [3] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. *IEEE Transactions on Image Processing*, 24(12):5706–5722, 2015.
- [4] A. Bulat and G. Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1021–1030, 2017.
- [5] C.-F. R. Chen, Q. Fan, N. Mallinar, T. Sercu, and R. Feris. Big-Little Net: An Efficient Multi-Scale Feature Representation for Visual and Speech Recognition. In *International Conference on Learning Representations*, 2019.
- [6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018.
- [7] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.
- [8] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *The European Conference on Computer Vision (ECCV)*, September 2018.
- [9] Y. Chen, H. Fang, B. Xu, Z. Yan, Y. Kalantidis, M. Rohrbach, S. Yan, and J. Feng. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [10] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng. Dual path networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4467–4475, 2017.
- [11] B. Cheng, R. Xiao, J. Wang, T. Huang, and L. Zhang. High frequency residual learning for multi-scale image classification. In *British Machine Vision Conference (BMVC)*, 2019.
- [12] M.-M. Cheng, Y. Liu, W.-Y. Lin, Z. Zhang, P. L. Rosin, and P. H. S. Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. *Computational Visual Media*, 5(1):3–20, Mar 2019.
- [13] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, 2015.
- [14] Y. Cheng, D. Wang, P. Zhou, and T. Zhang. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:1710.09282*, 2017.
- [15] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [16] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.
- [17] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.
- [18] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.
- [19] S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1135–1143, 2015.
- [20] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2011.
- [21] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.
- [22] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015.
- [23] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [24] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr. Deeply supervised salient object detection with short connections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4):815–828, 2019.
- [25] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [26] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [27] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.
- [28] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.
- [29] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5455–5463, 2015.
- [30] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 280–287, 2014.
- [31] M. Lin, Q. Chen, and S. Yan. Network in network. In *Int. Conf. Learn. Represent.*, 2013.
- [32] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017.
- [33] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.
- [34] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang. A simple pooling-based design for real-time salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [35] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):353–367, 2011.
- [36] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, pages 21–37. Springer, 2016.
- [37] Y. Liu, M.-M. Cheng, X. Hu, J.-W. Bian, L. Zhang, X. Bai, and J. Tang. Richer convolutional features for edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1939–1946, 2019.
- [38] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.
- [39] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [40] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *European Conference on Computer Vision (ECCV)*, September 2018.
- [41] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis. Ssh: Single stage headless face detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4875–4884, 2017.
- [42] G.-Y. Nie, M.-M. Cheng, Y. Liu, Z. Liang, D.-P. Fan, Y. Liu, and Y. Wang. Multi-level context ultra-aggregation for stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3283–3291, 2019.
- [43] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 91–99, 2015.
- [44] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- [45] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.
- [46] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NIPS)*, pages 568–576, 2014.
- [47] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Int. Conf. Learn. Represent.*, 2014.
- [48] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In *IEEE Conference on*

- Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [49] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang. High-resolution representations for labeling pixels and regions. *CoRR*, abs/1904.04514, 2019.
  - [50] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *The National Conference on Artificial Intelligence (AAAI)*, volume 4, page 12, 2017.
  - [51] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.
  - [52] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.
  - [53] J. Wang, H. Jiang, Z. Yuan, M.-M. Cheng, X. Hu, and N. Zheng. Salient object detection: A discriminative regional feature integration approach. *International Journal of Computer Vision*, 123(2):251–268, 2017.
  - [54] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
  - [55] B. Xiao, H. Wu, and Y. Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision (ECCV)*, September 2018.
  - [56] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. **Aggregated residual transformations for deep neural networks**. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995. IEEE, 2017.
  - [57] S. Xie and Z. Tu. Holistically-nested edge detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1395–1403, 2015.
  - [58] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1155–1162, 2013.
  - [59] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3166–3173, 2013.
  - [60] F. Yu, D. Wang, E. Shelhamer, and T. Darrell. Deep layer aggregation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2403–2412, 2018.
  - [61] S. Zagoruyko and N. Komodakis. Wide residual networks. In *British Machine Vision Conference (BMVC)*, 2016.
  - [62] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 202–211, 2017.
  - [63] T. Zhang, C. Xu, and M.-H. Yang. Multi-task correlation particle filter for robust object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
  - [64] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
  - [65] J. Zhao, Y. Cao, D.-P. Fan, X.-Y. Li, L. Zhang, and M.-M. Cheng. Contrast prior and fluid pyramid integration for rgbd salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
  - [66] K. Zhao, S. Gao, W. Wang, and M.-M. Cheng. Optimizing the F-measure for threshold-free salient object detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.
  - [67] K. Zhao, W. Shen, S. Gao, D. Li, and M.-M. Cheng. Hi-Fi: Hierarchical feature integration for skeleton detection. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.
  - [68] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1265–1274, 2015.



**Shang-Hua Gao** is a master student in Media Computing Lab at Nankai University. He is supervised via Prof. Ming-Ming Cheng. His research interests include computer vision, machine learning, and radio vortex wireless communications.



**Ming-Ming Cheng** received his PhD degree from Tsinghua University in 2012, and then worked with Prof. Philip Torr in Oxford for 2 years. He is now a professor at Nankai University, leading the Media Computing Lab. His research interests include computer vision and computer graphics. He received awards including ACM China Rising Star Award, IBM Global SUR Award, etc. He is a senior member of the IEEE and on the editorial boards of IEEE TIP.



**Kai Zhao** Kai Zhao is currently a Ph.D candidate with college of computer science, Nankai University, under the supervision of Prof Ming-Ming Cheng. His research interests mainly focus on statistical learning and computer vision.



**Xin-Yu Zhang** is an undergraduate student from School of Mathematical Sciences at Nankai University. His research interests include computer vision and deep learning.



**Ming-Hsuan Yang** is a professor in Electrical Engineering and Computer Science at University of California, Merced. He received the PhD degree in Computer Science from the University of Illinois at Urbana-Champaign in 2000. Yang has served as an associate editor of the IEEE TPAMI, IJCV, CVIU, etc. He received the NSF CAREER award in 2012 and the Google Faculty Award in 2009.



**Philip Torr** received the PhD degree from Oxford University. After working for another three years at Oxford, he worked for six years for Microsoft Research, first in Redmond, then in Cambridge, founding the vision side of the Machine Learning and Perception Group. He is now a professor at Oxford University. He has won awards from top vision conferences, including ICCV, CVPR, ECCV, NIPS and BMVC. He is a senior member of the IEEE and a Royal Society Wolfson Research Merit Award holder.