Inception Convolution with Efficient Dilation Search

Jie Liu[1*], Chuming Li[2*], Feng Liang[2,5], Chen Lin[3], Ming Sun[2,5], Junjie Yan[2], Wanli Ouyang[4], Dong Xu[4]
[1]Beihang University, [2]SenseTime Research
[3]University of Oxford, [4]The University of Sydney, [5]Shanghai AI Laboratory
ljie@buaa.edu.cn, chen.lin@eng.ox.ac.uk, {wanli.ouyang, dong.xu}@sydney.edu.au
{lichuming, liangfeng, sunming1, yanjunjie}@sensetime.com

Abstract

As a variant of standard convolution, a dilated convolution can control effective receptive fields and handle large scale variance of objects without introducing additional computational costs. To fully explore the potential of dilated convolution, we proposed a new type of dilated convolution (referred to as inception convolution), where the convolution operations have independent dilation patterns among different axes, channels and layers. To develop a practical method for learning complex inception convolution based on the data, a simple but effective search algorithm, referred to as efficient dilation optimization (EDO), is developed. Based on statistical optimization, the EDO method operates in a low-cost manner and is extremely fast when it is applied on large scale datasets. Empirical results validate that our method achieves consistent performance gains for image recognition, object detection, instance segmentation, human detection, and human pose estimation. For instance, by simply replacing the 3 × 3 standard convolution in the ResNet-50 backbone with inception convolution, we significantly improve the AP of Faster R-CNN from 36.4% to 39.2% on MS COCO.

1. Introduction

As an important concept of convolution neural network, the receptive field has been extensively studied. In [29], Luo *et al*. showed that the intensity in each receptive field roughly obeys a Gaussian distribution and only few pixels around the central part of the receptive field can effectively contribute to the response of the output neuron. Furthermore, in the previous works [23, 32], a more carefully defined optimal receptive field (ORF) has been evaluated for different tasks.

The requirement of the optimal receptive field is due to the variance of the input image sizes or the scales of objects of interest. For instance, for image classification, the input sizes tend to be small (e.g, 224 × 224), while for object detection, the input sizes are much larger and the objects can be from a large range of scales. Correspondingly, different tasks would require different ERFs. The different requirements of ERFs from different tasks makes it necessary to develop a general and practical optimization algorithm to learn the optimal ERF for a specific task.

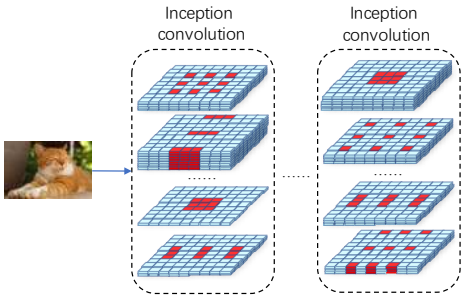

Figure 1. Inception convolution contains rich dilation patterns along both spatial axes and channels in each convolution layer.

As discussed in [29], the dilation value of dilated convolution kernels is a highly effective hyper-parameter to control the receptive field for different tasks. The work in [23] proposed to assign different dilation values at different stages of a CNN, which achieves consistent performance improvements. Subsequently, NATS [32] divided a convolution operation into different groups with each having independent dilation values. However, they apply the standard network architecture search methods in the relatively coarse search spaces, which neglects the fine-grained inner structure of dilated convolution. Therefore, in this work, we focus on exploring the search problem in the dilation domain to efficiently learn the optimal receptive field.

First of all, we would like to have a more flexible search space when compared with [23]. Flexibility can make us learn the optimal receptive field in order to better fit to different datasets. As shown in Figure 1, we propose a new type of dilated convolution, called Inception Convolution, which contains as much as possible dilation patterns. In the

*Equal contribution

space of inception convolution, the dilation pattern along each axis, each channel, and each convolution layer is independently defined. As a result, a dense range of possible receptive fields are considered in our inception convolution.

For optimization, a direct solution is to use the existing works in neural architecture search (NAS), which enables automatic search for the optimal combination of various network operations. DARTS [27] and single path one-shot [15] (SPOS) are two main families of efficient NAS methods. DARTS trained a supernet, where discrete operation selection was relaxed to a continuous weighted sum of the output from all candidate operations. After training, in each block, the operation with the largest architecture weight was chosen. SPOS randomly selected an operation sequence (subnet) from a pre-trained supernet and the same operation in different sequences share the same weights. After training, SPOS selected the best operation sequence via sampling and evaluation of multiple sequences with the shared-weights.

However, both DARTS and SPOS are not suitable for our huge search space. In DARTS, during training all operations in a block are applied to the input to make the architecture weights aware of each operation's importance, but the number of dilation patterns for a convolution layer (block) is large, *i.e.*, 16 if each of the two axes has 4 choices. It means DARTS requires 16 sequential calculations, thus it has low GPU utility and huge computational costs. SPOS samples the operation sequences during training. However, in our search space, the number of dilation patterns even in a single convolution layer is huge, *i.e.*, $(d_{max})^{2C}$, where C is the number of channels and d_{max} is the maximum dilation value. Due to the huge number of dilation patterns (paths), it is an extremely difficult task for SPOS as well.

In this work, we propose a simple and efficient dilation optimization algorithm (EDO). In EDO, each layer of the supernet is a standard convolution operation whose kernel covers all possible dilation patterns. After pre-training of the supernet, we select the dilation pattern for each channel in each convolution layer by solving a statistical optimization problem. Specifically, for each layer, according to the pre-trained weights, we minimize the L_1 error between the output of the original convolution layer and the output of the learned dilated convolution layer with the selected dilation pattern, based on which we can learn the optimal dilation pattern for this layer.

EDO supports efficient channel-wise dilation pattern selection over our complete dilation pattern search space. When compared with the search based method in [15], the search cost of our methods is very low. When compared with the differentiable NAS methods [27, 3], EDO converts sequential calculation related to different dilation patterns into a parallel way, thus it has lower computation cost and higher GPU utilization. Further, when compared with SPOS, we do not need to design further mechanism to handle the extremely large number of dilation patterns (paths).

Our contributions are three folds: 1) We propose a new type of dilated convolution, referred to as inception convolution, which can be readily combined with a large number of backbones for various visual recognition tasks; 2) We also propose a low-cost statistical optimization based network architecture search algorithm EDO, which can efficiently learn the optimal receptive field based on the data; 3) Comprehensive experiments demonstrate that our learnt inception convolution in combination with various backbones generally achieves performance improvement for various visual tasks without introducing any additional computational costs.

2. Related Work

2.1. Receptive Field

The Receptive Field is a crucial concept in Convolutional Neural Networks (CNNs). Traditional CNNs [18, 37, 17] stacked multiple convolutional layers to enhance the receptive field. The inception networks [39, 40, 38] introduced different size of filters, which are operated on the same level to aggregate different receptive fields. Deformable convolution [10] predicted the sampling offsets with respect to the preceding feature maps to adjust the receptive fields automatically. Scale-Adaptive Convolutions [45] predicted local flexible-sized dilations at each position to cover objects of various sizes. However, these methods are unfriendly to hardware optimization and thus cannot be used for real-time applications.

Dilated (Atrous) convolution [43, 8] changes the receptive field by performing convolution at sparsely sampled locations, which have been widely used in semantic segmentation [7, 46] and object detection [22, 21, 31]. PSConv [19] manually mixed up a spectrum of dilation rates in one convolution, which is shown to be sub-optimal in our experiments. In this work, we aim at searching for efficient inception convolution, which can be treated as a mixed dilated convolution and it is also friendly to hardware optimization.

2.2. Neural Architecture Search

Neural architecture search has attracted increasing attention. Early NAS approaches [48, 49] are computationally expensive due to the evaluation of each candidate. To reduce the searching costs, EcoNAS [47] proposed an EA-based algorithm to improve searching efficiency. Recently, One-shot NAS methods [1, 27, 3, 15, 2, 20, 14, 28] built a direct acyclic graph G (a.k.a., supernet) to subsume all architectures in the search space to further reduce the costs. More relevant to our work, some NAS works were proposed to search for dilated convolutions. NATS [32] employed the DARTS method [27] to search for the dilated rate at the group level in the CNN backbone. CRNAS [23] searched

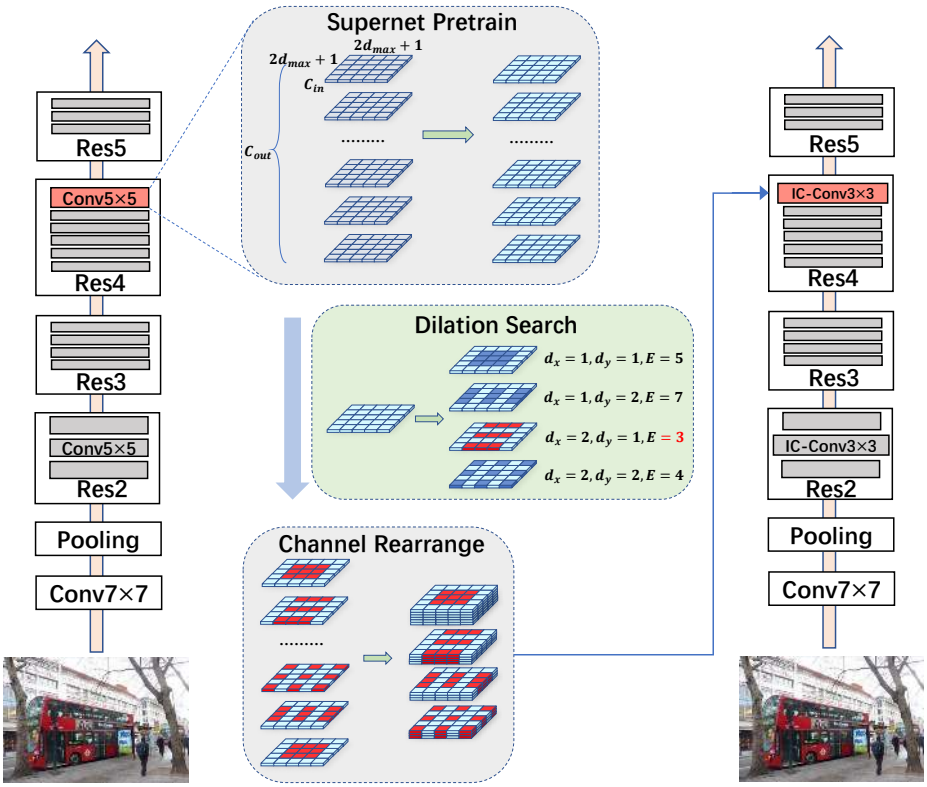

Figure 2. An overview of our EDO algorithm. ResNet50 (R50) is taken as an example. Firstly, for the convolution layers in R50 with the kernel size of 3×3, we change the kernel size to $(2d_{max}+1) \times (2d_{max}+1)$, where we set d_{max} = 2 in this figure, namely the kernel size of the supernet is 5×5. Secondly, we select the optimal dilation pattern with the best representation ability, which leads to the minimum representation error E (see Section 3.2 for more details). For one channel, dilation (2,1) with E = 3 is selected in this example. Finally, we rearrange the filters so that the filters with the same dilation patterns are arranged together, which produce our Inception Convolution.

for different dilation rates for different building blocks by using the SPOS method [15]. However, the search spaces of the aforementioned methods are limited. Moreover, simply adopting existing searching method, like DARTS [27] or SPOS [15] cannot work well for our search task as these methods cannot handle the extremely large number of possible operations in a single convolution layer.

3. Methodology

3.1. Problem Formulation of Inception Convolution

To fully explore the flexibility of dilation, in our inception convolution we consider a complete dilation space. An Inception Convolution has independent dilation values for the two axes in each channel and is formally represented as follows:

$$\begin{aligned} \boldsymbol{d} = &\{(d_x^i, d_y^i) | d_x^i \in \{1, 2, .., d_{max}\}, d_y^i \in \{1, 2, .., d_{max}\}, \\ &i \in \{1, 2, ..., C^{out}\}\}, \end{aligned} \tag{1}$$

where d_x^i and d_y^i are the dilation values in x axis and y axis of the filter at the i-th output channel, ranging from 1 to d_{max}, and C^{out} denotes the number of output channels. The total number of all possible dilation patterns in a single inception convolution is $(d_{max})^{2C_{out}}$. In this work, we aim to develop an algorithm to efficiently learn the optimal receptive field for different tasks by selecting the optimal $\boldsymbol{d}$.

3.2. Solution

Recently, NAS evolves as an effective way to learn high-performance network architectures in the specified search spaces. DARTS and SPOS are two main-stream families of NAS methods. However, as inception convolution contains $(d_{max})^2$ dilation patterns and $(d_{max})^{2C_{out}}$ candidates, both DARTS and SPOS cannot be used for efficient search in the huge search space.

Both works SPOS and DARTS show that the weights in a pre-trained supernet are informative to guide the selection operation. In this work, we follow this idea and formulate a statistical optimization problem to select the optimal dilation patterns based on the corresponding weights in a pre-trained supernet. Our optimization algorithm, EDO, is simple, effective, and efficient. The pipeline of our proposed method is shown in Figure 2.

Supernet. Given a network architecture, we construct a supernet and then we keep the architecture unchanged but

change the kernel sizes to cover all candidate dilation patterns. Formally, for a convolution layer in the supernet with the kernel size of $2k+1$, we replace it with the convolution kernel with the size of $2kd_{max}+1$, which is the maximum width and height for all candidate dilation patterns. Note that the supernet is pre-trained for each given task.

Statistical Optimization. For each convolution layer with the weights $\boldsymbol{W}$, we define $\boldsymbol{W}^i$ as the weights of the i-th convolutional filter in our supernet, and denote $\boldsymbol{d}$ as the set of $\{(d_x^i, d_y^i)|_{i=1}^{C_{out}}\}$ where d_x^i and d_y^i represent the sampled positions from the i-th channel. Based on d_x^i and d_y^i, we can generate $\widetilde{\boldsymbol{W}}^i_{d_x^i,d_y^i}$, in which the sampled positions are filled with the corresponding values from $\boldsymbol{W^i}$ while the unsampled positions are filled with zeros. We stack $\widetilde{\boldsymbol{W}}^i_{d_x^i,d_y^i}$ along the output channel dimension and produce $\widetilde{\boldsymbol{W}}_{\boldsymbol{d}}$. Note the dimensions of $\boldsymbol{W}$ and $\widetilde{\boldsymbol{W}}_{\boldsymbol{d}}$ are the same (*i.e.*, $\boldsymbol{W}, \widetilde{\boldsymbol{W}}_{\boldsymbol{d}} \in \mathbb{R}^{C_{out}\times C_{in}\times(2kd_{max}+1)\times(2kd_{max}+1)}$) and the dimensions of $\boldsymbol{W}^i$ and $\widetilde{\boldsymbol{W}}^i_{d_x^i,d_y^i}$ are also the same (*i.e.*, $\boldsymbol{W}^i, \widetilde{\boldsymbol{W}}^i_{d_x^i,d_y^i} \in \mathbb{R}^{C_{in}\times(2kd_{max}+1)\times(2kd_{max}+1)}$), where C_{in} and C_{out} are the numbers of input and output channels. We formulate the dilation pattern selection task as an optimization problem, where the L_1 error between the expectation of the output from the pre-trained weights $\boldsymbol{W}$ and the expectation of the output from the sampled dilation weights $\widetilde{\boldsymbol{W}}_{\boldsymbol{d}}$ is minimized. Formally, we arrive at:

$$\begin{aligned} &\min_{\boldsymbol{d}} \ \|E[\boldsymbol{W} * \boldsymbol{X}] - E[\widetilde{\boldsymbol{W}}_{\boldsymbol{d}} * \boldsymbol{X}]\|_1, \\ &s.t. \ \ d_x^i \in \{1, ..., d_{max}\}, d_y^i \in \{1, ..., d_{max}\}, \end{aligned} \tag{2}$$

where $\boldsymbol{X} \in \mathbb{R}^{B\times C_{in}\times H\times W}$ is the input of this convolution layer with the batch size of B, the number of input channels C_{in}, the height H and the width W. We use $*$ to denote the convolution operator, and E is the expectation operator. As $\boldsymbol{W}$ and $\widetilde{\boldsymbol{W}}_{\boldsymbol{d}}$ are independent of $\boldsymbol{X}$, the objective function in Eq. (2) is further expressed as follows:

$$\begin{aligned} &\|E[\boldsymbol{W} * \boldsymbol{X}] - E[\widetilde{\boldsymbol{W}}_{\boldsymbol{d}} * \boldsymbol{X}]\|_1 \\ &= \|\boldsymbol{W} * E[\boldsymbol{X}] - \widetilde{\boldsymbol{W}}_{\boldsymbol{d}} * E[\boldsymbol{X}]\|_1, \\ &= \|(\boldsymbol{W} - \widetilde{\boldsymbol{W}}_{\boldsymbol{d}}) * E[\boldsymbol{X}]\|_1. \end{aligned} \tag{3}$$

In the above optimization problem, we assume that, after batch normalization, the mean feature value does not vary too much across different positions of $\boldsymbol{X}$, which means that $E[\boldsymbol{X}]$ has almost the same value among all positions. Assuming the mean value is α, then we propose to solve an alternative optimization problem as follows:

$$\min_{\boldsymbol{d}} \sum_{i=1}^{C^{out}} \alpha \|(\boldsymbol{W}^i - \widetilde{\boldsymbol{W}}^i_{d_x^i,d_y^i}) * \boldsymbol{1}\|_1, \tag{4}$$

where $\boldsymbol{1}$ is an all-ones matrix with the same dimension as $\boldsymbol{X}$, and α is a constant scalar which can be omitted in optimization. Based on Eq. (4), the optimal $\boldsymbol{d}$ of each convolution layer can be easily solved, by independently traversing all dilation patterns (d_x^i, d_y^i) for each filter $\boldsymbol{W}^i$ with little cost.

3.3. Discussion

Relationship with DARTS. An intuitive application of DARTS to our inception convolution is introduced in [32]. In DARTS, $(d_{max})^2$ operations are calculated sequentially, while our EDO algorithm can calculate the $(d_{max})^2$ operations in parallel. Besides, the total cost of DARTS is $C_{in} \times C_{out} \times (2k+1)^2 \times (d_{max})^2$, but the total cost of our EDO is $C_{in} \times C_{out} \times (2kd_{max}+1)^2$. For most CNNs, where k is usually 1, $C_{in} \times C_{out} \times (2kd_{max}+1)^2$ is only 56% of $C_{in} \times C_{out} \times (2k+1)^2 \times (d_{max})^2$ when d_{max} equals to 4. Therefore, EDO is more efficient than DARTS.

Additionally, as shown in [44], DARTS degenerates into random sampling in some cases because the principal eigenvalue of the Hessian matrix of the architecture parameters appears large. However, we directly define the statistical optimization problem over the pre-trained network weights rather than introducing the architecture parameters, which is more robust than DARTS.

Relationship with NATS and CRNAS. While NATS and CRNAS also considers the search space for searching flexible dilation patterns, it is less complete when compared with our inception convolution. CRNAS searches the dilation patterns independently at each stage, thus it is based on SPOS. NATS divides a convolution into the groups and search among a few dilation patterns (usually 5 patterns) for each group with DARTS. In contrast, our inception convolution is channel-wise and contains all dilation patterns within the max dilation value d_{max}.

4. Experiments

4.1. Image Recognition

4.1.1 Dataset and implementation details

For image recognition, we evaluate our method on the ImageNet dataset [34] with 1.28M training images and 50k validation images. We first train our supernet with the largest kernel size (*i.e.*, 9) and follow the standard training procedure in [17]. More specifically, we use stochastic gradient descent (SGD) as the optimizer with the momentum of 0.9 and the weight decay of 0.0001. The supernet is trained for 100 epochs with the batch size of 1024 without any trick. We adopt the cosine learning rate scheduler with the initial learning rate of 0.4. Then we perform EDO to obtain the best inception convolutions, as described in Section 3. The resultant IC-Net is retrained by using the same procedure as

the supernet training process. Below, each method in combination with our inception convolution is also referred to as "IC-X" (e.g. "IC-ResNet18").

4.1.2 Experimental results

We search inception convolution based on various types of networks, from MobileNetV2 [35] to ResNeXt [42]. As shown in Table 1, our inception convolution consistently boosts the performance on ImageNet. In terms of top-1 accuracy, IC-ResNet18 and IC-ResNet50 outperform the baselines by 1.07% and 1.11%, respectively. Our inception convolution is also compatible with other networks consisting of depth-wise convolution or group convolution. For instance, in terms of top-1 accuracy, inception convolution leads to 1.21% and 0.62% gain on MobileNetV2 and ResNeXt101, respectively.

Table 1. Top-1/top-5 accuracy (%) comparison on the ImageNet validation set. All the methods are compared under the single center crop evaluation setting. **The baseline results are re-implemented by ourselves based on the same code as our IC-Net.**

Network Architectures	Conv Types	Top-1/5 Acc.(%)
MobileNetV2 [35]	Standard	70.71 / 89.81
	IC-Conv	**71.92 / 90.54**
ResNet18 [17]	Standard	70.67 / 89.74
	IC-Conv	**71.74 / 90.91**
ResNet50 [17]	Standard	76.19 / 92.93
	IC-Conv	**77.30 / 93.58**
ResNeXt101 (32x4d) [42]	Standard	78.71 / 94.20
	IC-Conv	**79.33 / 94.74**

4.2. Object Detection

4.2.1 Dataset and implementation details

In the following experiments, unless otherwise stated, we will only replace the standard convolutions with our searched inception convolutions in the backbone. For object detection, we use MS COCO 2017 [26] for the experiments. The dataset is challenging due to the huge variation of object scales and a large number of objects per image. The supernet with the largest kernel size (*i.e.*, 9) is used as the pre-trained model to generate the inception convolutions. For detector training, we use stochastic gradient descent (SGD) as the optimizer with the momentum of 0.9 and the weight decay of 0.0001. The model is trained for 13 epochs, known as 1× schedule [13]. We use the multi-GPU training strategy over eight 1080TI GPUs with a total batch size of 16. The initial learning rate is 0.00125 per image and is divided by 10 at the 8th and the 11th epochs. Warm-up is adopted for both baselines and our method.

4.2.2 Experimental results

Our searched inception convolution has great potential to be combined with various detectors, such as Faster R-CNN [33] and Cascade R-CNN [4]. The same type of detectors is trained by using exactly the same 1× training procedure [13]. We replace all the 3×3 convolutions in the pre-trained backbone network by our inception convolution, while the convolutions in the FPN neck are kept as the standard convolutions. As shown in Table 3, our searched inception convolution boosts the performance of an extensive range of backbones on COCO. For Faster R-CNN [33, 24] with FPN, our backbones of IC-ResNet50, IC-ResNet101, and IC-ResNeXt101-32x4d outperform the baseline backbones of ResNet50, ResNet101, and ResNeXt101-32x4d by large margins of 2.5%, 3.1% and 1.6% respectively. Our inception convolution is especially effective for large objects (4.1%, 4.1% and 2.4% improvement in terms of AP_L), possibly due to the large receptive field provided by dilation pattern search. For a more powerful method, Cascade R-CNN [4] with FPN, our method is also effective. Our backbone IC-ResNeXt101-32x4d achieves the AP of 45.7%, which is 1.3% higher than the baseline backbone.

4.2.3 Results from/on different detectors/datasets

Table 2. Results (AP%) of different detectors when using the standard convolution and our inception convolution on the COCO 2017 validation set. For all detectors, we achieve consistent performance improvements. For DETR, due to limited computational resources, we use the officially released training scripts with 150 epochs rather than the one with 500 epochs in the original work.

Detector	Backbone	Conv Type	AP
Faster R-CNN (C4) [33]	ResNet50	Standard	35.0
		IC-Conv	38.5 (+3.5)
RetinaNet [25]	ResNet50	Standard	36.0
		IC-Conv	37.9 (+1.9)
DETR [6]	ResNet50	Standard	39.7
		IC-Conv	40.7 (+1.0)
FCOS [41]	ResNet50	Standard	37.2
		IC-Conv	38.8 (+1.6)
Faster R-CNN (NAS-FPN) [12]	ResNet50	Standard	40.2
		IC-Conv	41.1 (+0.9)

Different detectors. We transfer our searched inception convolution to more types of detectors, including the one-stage detector RetinaNet [25], the anchor box free detector FCOS [41], NAS-FPN [12], and the transformer based detector DETR [6]. The experimental results on COCO are reported in Table 2. Equipped with our inception convolution, the average AP gain for the five detectors is 1.8% on COCO without additional FLOPs cost. In particular, our inception convolution is compatible with the new transformer based detector DETR. In combination with our inception convolution, DETR can achieve the AP of 40.7% on COCO, which is 1.0% higher than the vanilla DETR method.

Table 3. Detection performance (AP%) of different backbones when using the standard convolution and our inception convolution on the COCO 2017 validation set.

Detector	Backbone	Conv Type	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster R-CNN [33]	ResNet50	Standard	36.4	58.6	39.2	21.7	40.2	46.4
		IC-Conv	38.9 (+2.5)	61.6	41.8	22.9	42.3	50.5
	ResNet101	Standard	38.8	60.9	42.1	22.6	42.9	50.5
		IC-Conv	41.9 (+3.1)	64.2	45.3	25.5	45.8	54.6
	ResNeXt101-32x4d	Standard	40.5	63.1	44.4	24.9	44.8	52.0
		IC-Conv	42.1 (+1.6)	64.7	45.7	25.5	46.1	54.4
Cascade R-CNN [4]	ResNet50	Standard	40.5	59.2	44.0	22.5	43.9	53.6
		IC-Conv	42.4 (+1.9)	62.0	46.0	25.2	45.9	56.0
	ResNet101	Standard	42.6	60.9	46.2	23.8	46.2	56.9
		IC-Conv	45.0 (+2.4)	64.8	48.7	26.9	49.0	59.6
	ResNeXt101-32x4d	Standard	44.4	63.6	48.5	25.8	48.5	58.1
		IC-Conv	45.7 (+1.3)	65.5	49.7	27.1	49.8	59.8

Different datasets. We also directly evaluate our searched inception convolution on another object detection dataset VOC [11] by using the RFCN framework [9]. As shown in Table 4, in terms of AP_{50}, our backbones IC-ResNet50 and IC-ResNet101 achieve the improvements by 1.94% and 1.59%, when compared with the baseline backbones.

Table 4. Detection performance (AP_{50}%) on VOC test2007 by using the RFCN framework with the standard convolution and our inception convolution in the backbones ResNet50 and ResNet101.

ResNet50	IC-ResNet50	ResNet101	IC-ResNet101
79.66	**81.60**	81.36	**82.95**

4.3. Instance Segmentation

4.3.1 Dataset and implementation details

We further search for the dilation patterns for the instance segmentation task by using the Mask R-CNN [16] framework on the MS COCO [26] dataset. The supernet with the kernel size of 9 is also used as the pre-trained model to generate the inception convolutions. For Mask R-CNN training, we use the same training configurations as in Section 4.2.1.

4.3.2 Experimental results

We search for the dilation patterns based on Mask R-CNN [16] and transfer it to Cascade Mask R-CNN [5]. As shown in Table 6, our searched inception convolution consistently boosts the instance segmentation accuracy. For Mask R-CNN with FPN, in terms of box AP, our backbones of IC-ResNet50, IC-ResNet101 and IC-ResNeXt101-32x4d outperform the baseline backbones by 2.8%, 2.8% and 2.0% respectively. In particular, for the stronger Cascade Mask R-CNN with FPN, our backbone of IC-ResNeXt101-32x4d can further improve the box AP of the baseline backbone by 1.5%.

4.4. Comparison with other dilation search methods

According to the results available in other works, we choose Faster R-CNN and Mask R-CNN as two examples for comparison. In Table 5, we compare our method with POD [31], NATS [32], PSConv[2] [19] and CRNAS [23]. Remarkably, our method achieves better performance than all the other baseline methods when using different backbones and detectors. We have a larger channel-wise search space and we also propose a more efficient search algorithm, which leads to the improved detection accuracy.

Table 5. Comparison (AP%) between our inception convolution search method with other dilation search methods on the COCO 2017 validation set. The results in the column "Standard Conv" are copied from their original works. These results are comparable, which shows that we do not use any implementation trick. R50, R101 and X101-32x4d denote ResNet50, ResNet101 and ResNeXt101-32x4d, respectively.

Method	Backbone	Conv type	Standard Conv	Searched Conv
Faster R-CNN [33]	R50	POD	36.2	37.9
		NATS	36.4	38.4
		PSConv	36.4	38.4
		CRNAS	36.4	38.3
		Ours	36.4	**38.9**
	R101	POD	38.6	40.1
		NATS	38.6	40.4
		PSConv	38.5	40.9
		CRNAS	38.6	40.2
		Ours	38.8	**41.9**
	X101-32x4d	NATS	40.5	41.6
		CRNAS	40.6	41.5
		PSConv	40.1	41.3
		Ours	40.5	**42.1**
Mask R-CNN [16]	R50	NATS	37.5	39.3
		PSConv	37.3	39.4
		CRNAS	37.6	39.1
		Ours	37.2	**40.0**
	R101	PSConv	39.4	41.6
		CRNAS	39.7	41.5
		Ours	39.8	**42.6**
	X101-32x4d	PSConv	41.1	42.4
		Ours	41.4	**43.4**

[2]Dilation patterns in PSConv are not searched but specifically designed.

Table 6. Detection and instance segmentation results of Mask R-CNN and Cascade Mask R-CNN on the COCO 2017 validation set. Box AP (%) and mask AP (%) of the bounding boxes and the segmentation results are reported, respectively. R50, R101 and X101-32x4d denote ResNet50, ResNet101 and ResNeXt101-32x4d, respectively.

Detector	Backbone	Conv Type	Box AP						Mask AP					
			AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN [16]	R50	Standard	37.2	59.0	40.1	22.3	41.2	47.7	33.8	55.7	35.8	17.9	37.6	45.9
		IC-Conv	$40.0_{(+2.8)}$	62.1	43.1	23.5	43.7	52.1	$35.9_{(+1.9)}$	58.4	37.9	18.9	39.5	49.5
	R101	Standard	39.8	61.8	43.5	23.2	43.9	51.9	35.6	58.1	38.1	18.4	39.5	49.0
		IC-Conv	$42.6_{(+2.8)}$	64.6	46.4	25.6	46.6	55.4	$37.9_{(+2.3)}$	61.2	40.3	20.5	41.7	52.0
	X101-32x4d	Standard	41.4	63.6	45.2	24.6	45.9	53.2	37.1	60.1	39.6	19.4	41.3	50.9
		IC-Conv	$43.4_{(+2.0)}$	65.7	47.4	27.2	47.2	56.3	$38.4_{(+1.3)}$	62.1	40.4	21.3	42.0	53.0
Cascade Mask R-CNN [5]	R50	Standard	41.2	59.7	44.8	23.4	44.6	54.7	35.0	56.5	37.4	18.0	38.3	48.5
		IC-Conv	$43.4_{(+2.2)}$	62.5	47.0	25.3	47.1	57.1	$36.8_{(+1.8)}$	59.2	39.2	19.4	40.1	50.8
	R101	Standard	43.1	61.9	46.8	24.8	47.0	56.7	37.0	59.0	39.6	19.0	40.7	50.8
		IC-Conv	$45.7_{(+2.6)}$	65.2	49.8	26.8	49.6	61.0	$38.7_{(+1.7)}$	61.9	41.3	20.5	42.2	53.9
	X101-32x4d	Standard	44.9	64.1	48.9	26.1	48.3	59.4	37.9	60.6	40.6	20.0	41.2	52.6
		IC-Conv	$46.4_{(+1.5)}$	66.0	50.5	27.1	50.3	61.0	$39.1_{(+1.2)}$	62.6	41.6	20.6	42.7	53.7

4.5. Crowd Human Detection

In crowd scenarios, different people have large variations in terms of poses and scales. An optimal receptive field need to be learned for correctly perceiving neighboring features.

We use the CrowdHuman [36] dataset, which contains 15K images for training. Faster R-CNN with ResNet50 is adopted as the framework and the supernet kernel size is also 9. We use SGD as the optimizer with the momentum of 0.9 and the weight decay of 0.0001. The model is trained for 20 epochs and the learning rate is divided by 10 at the 10th and the 15th epochs. As shown in Table 7, on CrowdHuman, our backbone using inception convolution also outperforms the baseline backbone using the standard convolution by 0.8 % in terms of MR^{-2}. Note MR^{-2} is the most important metric on CrowdHuman and the results demonstrate the generalization capability of our inception convolution for different detection tasks.

Table 7. Detection performance of Faster R-CNN (C4) on *CrowdHuman* when using ResNet50 as the backbone together with the standard convolution and our inception convolution. MR^{-2} is the most critical indicator, in which lower value indicates better result.

Conv Type	Recall	AP	MR^{-2}
Standard	79.29	75.61	59.60
IC-Conv	**79.32**	**75.68**	**58.82**

4.6. Human Pose Estimation

Human pose estimation aims to locate keypoints or parts of human (e.g. elbow, wrist, etc). To fully validate the effectiveness of our inception convolution, we choose a more difficult multi-person pose estimation problem, where all the human parts in an image should be detected and the keypoints of the same person should be associated.

Dataset and implementation details. We train our model on the COCO train2017 dataset and evaluate our approach on the val2017 dataset. The supernet with the largest kernel size of 13 is used for searching the inception convolutions. Following the setting in MMPose (https://github.com/open-mmlab/mmpose), we adopt associative embedding [30] for bottom-up human pose estimation and use the Adam optimizer. The initial learning rate is set as $1e{-}3$, and then drops to $1e{-}4$ and $1e{-}5$ at the 200th and the 260th epochs, respectively. The training process is terminated within 300 epochs.

Experimental results. In Table 8, we report the results of our inception convolution and standard convolution for the same input size (*i.e.*, 640x640). Our backbones using inception convolution is more than **7%** higher than the baseline backbones using the standard convolution in terms of AP, which is a **significant** improvement.

4.7. Analysis of Our Design

4.7.1 Inception convolution and our EDO design

In this section, we take Faster R-CNN [33] with FPN as an example to validate the effectiveness of both inception convolution and EDO.

Enlarging the kernel size versus using inception convolution. ResNet50 is the vanilla ResNet method. In ResNet50_k9, we replace the 3×3 convolution operation in ResNet50 with the 9×9 convolution operation as our supernet also uses the kernel size of 9. In IC-ResNet50_1d, we use the same dilation rate for both horizontal and vertical directions, thus we only have 1-dimensional freedom of dilation values. As shown in Table 9, based on the simplified 1-dimensional choice of dilation values, IC-ResNet50_1d achieves a higher AP on COCO than the alternative method ResNet50_k9. It indicates that our inception convolution can well capture the optimal receptive field for each channel. On the other hand, our complete method IC-ResNet50 enables searching for the optimal dilation values along both horizontal and vertical directions and thus outperforms IC-ResNet50_1d as that searches for the optimal dilation patterns only along one direction.

Searching Inception Convolution per layer or per channel. In IC-ResNet50_sl, we keep the dilation values

Table 8. Comparison between inception convolution and standard convolution for pose estimation on the COCO 2017 validation set.

Method	Backbone	Input Size	Conv Type	AP	AP_{50}	AP_{75}	AP_M	AP_L	AR
Associative Embedding [30]	w/o multi-scale test setting								
	ResNet50	640x640	Standard	51.0	78.3	52.3	50.0	52.0	59.3
			IC-Conv	62.2 (+11.2)	84.6	67.8	55.2	72.2	67.5
	ResNet101	640x640	Standard	55.5	80.8	58.2	52.2	60.0	62.7
			IC-Conv	63.3 (+7.8)	85.8	69.4	55.9	74.0	68.3
	w/ multi-scale test setting								
	ResNet50	640x640	Standard	55.8	81.0	58.1	56.3	55.3	63.8
			IC-Conv	65.8 (+10.0)	85.9	71.3	60.3	73.9	70.7
	ResNet101	640x640	Standard	60.2	83.8	63.3	58.7	62.4	67.2
			IC-Conv	68.5 (+8.3)	87.9	74.2	63.6	75.7	73.0

Table 9. Detection performance (AP%) of ResNet50, our method and different variants of our method on the COCO 2017 validation set.

Method	AP (%)
ResNet50	36.4
ResNet50_k9	38.1
IC-ResNet50_1d	38.3
IC-ResNet50_sl	37.8
IC-ResNet50(Ours)	**38.9**

the same across all channels in each convolution layer. Our complete method IC-ResNet50 searches for the optimal dilation value for each channel, and it performs better than the alternative method IC-ResNet50_sl that only conducts searching per layer.

4.7.2 Convolution kernel sizes of the supernet

Larger kernel size introduces more dilation combination but brings more computation complexity in the retraining process of the supernet. In this section, we investigate the influence of the kernel size. We conduct all the detection experiments on COCO. As shown in Figure 3, when the kernel size grows from 3x3 to 13x13, the AP grows consistently. The result indicates that our method has great potential. By simply increasing the kernel size of the supernet from 9x9 to 13x13, higher gains can be achieved. By default, we set the kernel size as 9x9 because it is a good trade-off between accuracy and training time.

4.7.3 Searching head/neck of the detector

By default, we only search the optimal dilation patterns for the backbone part of the detector in Section 4.2. Here we further extend our method to search the head and neck parts of the detector. Specifically, we search all the 3×3 convolution in FPN [24] and RPN [33]. The results in Table 10 show that searching the dilation patterns for the head or neck part of the detector also leads to some improvement, though the improvement is less significant when compared

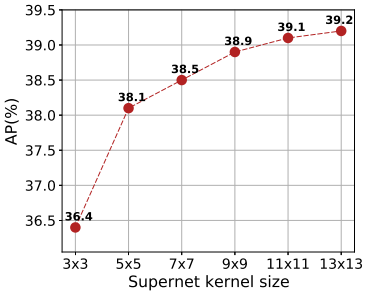

Figure 3. Detection performance (%) of our method when using different supernet kernel sizes on the COCO 2017 validation set.

Table 10. Searching different parts of the detector on the COCO 2017 validation set, in which we use Faster R-CNN with ResNet50 as the baseline.

Backbone	FPN [24]	RPN [33]	AP (%)
✓			38.9
✓	✓		39.0
✓	✓	✓	**39.2**

with the improvement for the backbone.

5. Conclusion

In this work, we have proposed inception convolution, which is obtained by searching channel-wise dilation patterns through Efficient Dilation Optimization (EDO). IC-Conv can effectively decide the optimal receptive field in a convolution operation and aggregate the information induced by the receptive field at multiple scales. Our IC-Conv generalizes well for a large range of tasks and can be plugged into arbitrary CNN architectures.

Acknowledgments This work was supported by the National Key Research and Development Project of China (No. 2018AAA0101900). Wanli Ouyang was supported by the Australian Research Council Grant DP200103223 and Australian Medical Research Future Fund MRFAI000085.

References

[1] Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. Understanding and simplifying one-shot architecture search. In *International Conference on Machine Learning*, pages 550–559, 2018.

[2] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019.

[3] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. *arXiv preprint arXiv:1812.00332*, 2018.

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: high quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020.

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan 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, 2017.

[8] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[9] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[10] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[11] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

[12] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7036–7045, 2019.

[13] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. https://github.com/facebookresearch/detectron, 2018.

[14] Ronghao Guo, Chen Lin, Chuming Li, Keyu Tian, Ming Sun, Lu Sheng, and Junjie Yan. Powering one-shot topological nas with stabilized share-parameter proxy. *arXiv preprint arXiv:2005.10511*, 2020.

[15] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. Single path one-shot neural architecture search with uniform sampling. In *European Conference on Computer Vision*, pages 544–560. Springer, 2020.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[19] Duo Li, Anbang Yao, and Qifeng Chen. Psconv: Squeezing feature pyramid into one compact poly-scale convolutional layer. *arXiv preprint arXiv:2007.06191*, 2020.

[20] Xiang Li, Chen Lin, Chuming Li, Ming Sun, Wei Wu, Junjie Yan, and Wanli Ouyang. Improving one-shot nas by suppressing the posterior fading. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13836–13845, 2020.

[21] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 6054–6063, 2019.

[22] Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. Detnet: A backbone network for object detection. *arXiv preprint arXiv:1804.06215*, 2018.

[23] Feng Liang, Chen Lin, Ronghao Guo, Ming Sun, Wei Wu, Junjie Yan, and Wanli Ouyang. Computation reallocation for object detection. *arXiv preprint arXiv:1912.11234*, 2019.

[24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[25] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[27] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[28] Jiaheng Liu, Shunfeng Zhou, Yichao Wu, Ken Chen, Wanli Ouyang, and Dong Xu. Block proposal neural architecture search. *IEEE Transactions on Image Processing*, 30:15–25, 2020.

[29] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in neural information processing systems*, pages 4898–4906, 2016.

[30] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in neural information processing systems*, pages 2277–2287, 2017.

[31] Junran Peng, Ming Sun, Zhaoxiang Zhang, Tieniu Tan, and Junjie Yan. Pod: practical object detection with scale-sensitive network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9607–9616, 2019.

[32] Junran Peng, Ming Sun, ZHAO-XIANG ZHANG, Tieniu Tan, and Junjie Yan. Efficient neural architecture transformation search in channel-level for object detection. In *Advances in Neural Information Processing Systems*, pages 14313–14322, 2019.

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[35] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[36] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018.

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[38] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016.

[39] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[40] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[41] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 9627–9636, 2019.

[42] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[43] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[44] Arber Zela, Thomas Elsken, Tonmoy Saikia, Yassine Marrakchi, Thomas Brox, and Frank Hutter. Understanding and robustifying differentiable architecture search. *arXiv preprint arXiv:1909.09656*, 2019.

[45] Rui Zhang, Sheng Tang, Yongdong Zhang, Jintao Li, and Shuicheng Yan. Scale-adaptive convolutions for scene parsing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2031–2039, 2017.

[46] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[47] Dongzhan Zhou, Xinchi Zhou, Wenwei Zhang, Chen Change Loy, Shuai Yi, Xuesen Zhang, and Wanli Ouyang. Econas: Finding proxies for economical neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11396–11404, 2020.

[48] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

[49] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.