Evolving Search Space for Neural Architecture Search
Supplementary Material

Shape	Block	c	n	s
$224^2 \times 3$	3x3 conv	16	1	2
$112^2 \times 16$	MBL	16	1	1
$112^2 \times 16$	MBL	24	4	2
$56^2 \times 24$	MBL	40	4	2
$28^2 \times 40$	MBL	80	4	2
$14^2 \times 80$	MBL	96	4	1
$14^2 \times 96$	MBL	192	4	2
$7^2 \times 192$	MBL	320	1	1
$7^2 \times 320$	1x1 conv	1024	1	1
$7^2 \times 1024$	7x7 avgpool	-	1	1
1024	fc	1024	1	-

Table 1. Macro-architecture for FLOPs constraint setting. "MBL" denotes the learnable Multi-Branch layer, c, n, s refer to the number of backbone filters, number of layers and the stride, respectively.

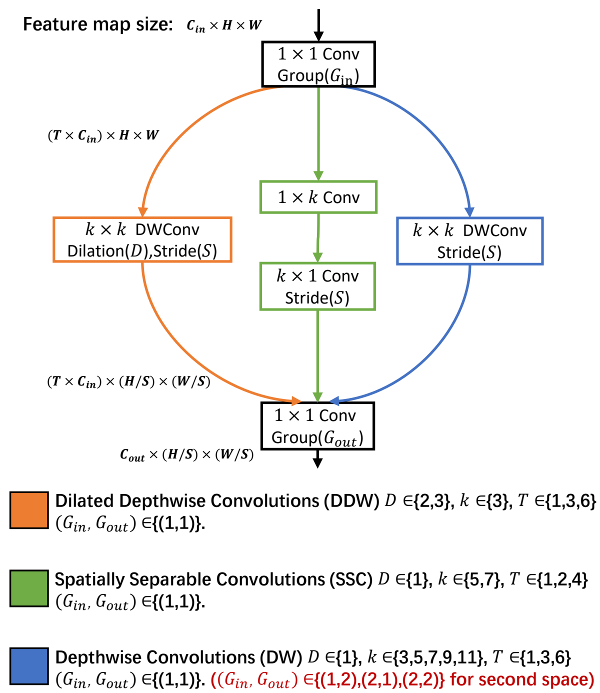


Figure 1. FLOPs Constraint search space details for all possible operations except Identity Mapping. The corresponding layer is recognized as reduction layer when $C_{in} \neq C_{out}$. Each type of operation has its corresponding kernel size k, dilation D, and expand ratio T. We do not search the group number (G_{in}, G_{out}) for 1×1 projections in the primitive 27 OPs space, for the second space, we search G in DW convolutions with varies k and T.

A. Search Space Details

A.1. FLOPs Constraint Search Space

The 27 OPs space for FLOPs constraint, as shown in Figure 1, is derived from multiple groups of operation designs. The first group of operations is depthwise (DW) convolution with kernel size $\{3, 5, 7, 9, 11\}$ and expand ratio $\{1, 3, 6\}$. The second group is 3×3 dilated convolution with dilation $\{2, 3\}$ and expand ratio $\{1, 3, 6\}$, this kind of operation, according to the study in MixNet [13], is not efficient under FLOPs constrained scenarios. However, we still include them in our search space to test the robustness of the proposed method and see if it can find competitive architectures in a noised large search space. We also include the $1 \times k - k \times 1$ convolutions with $k \in \{5, 7\}$ and expand ratio $\{1, 2, 4\}$, this operation is derived from the Inception-ResNet [11] and is a rarely included operation in NAS literature as well. Our major experiments are conducted in this setting.

The second space as shown in Figure 1 consists of DW convolutions with grouped 1×1 projections, a special variant of standard DW convolution that is included in FBNet [15] and MixNet [13]. The options of kernel size and expand ratio for this variant are $\{3, 5, 7, 9, 11\}$ and $\{1, 3, 6\}$ respectively, which is identical with standard DW convolutions in 27 OPs space. For both search space, we use identical macro-architecture as shown in Table 1.

A.2. Latency Constraint Search Space

Our search space for Latency constraint as shown in Figure 2 and Table 2 is identical with the extended search space used by Li *et al.* [4].

A.3. Identity Mapping Path

Inspired by the Inception-Resnet [11], our search space has a residual structure, which means that all normal layers in the network have an identity mapping path (identity op-

Shape	Block	c	n	s
$224^2 \times 3$	3x3 conv	16	1	2
$112^2 \times 16$	MBL	16	1	1
$112^2 \times 16$	MBL	32	4	2
$56^2 \times 32$	MBL	64	4	2
$28^2 \times 64$	MBL	128	8	2
$14^2 \times 128$	MBL	256	4	2
$7^2 \times 256$	1x1 conv	1024	1	1
$7^2 \times 1024$	7x7 avgpool	-	1	1
1024	fc	1024	1	-

Table 2. Macro-architecture for Latency constraint setting. "MBL" denotes the learnable Multi-Branch layer, c, n, s refer to the number of backbone filters, number of layers and the stride, respectively.

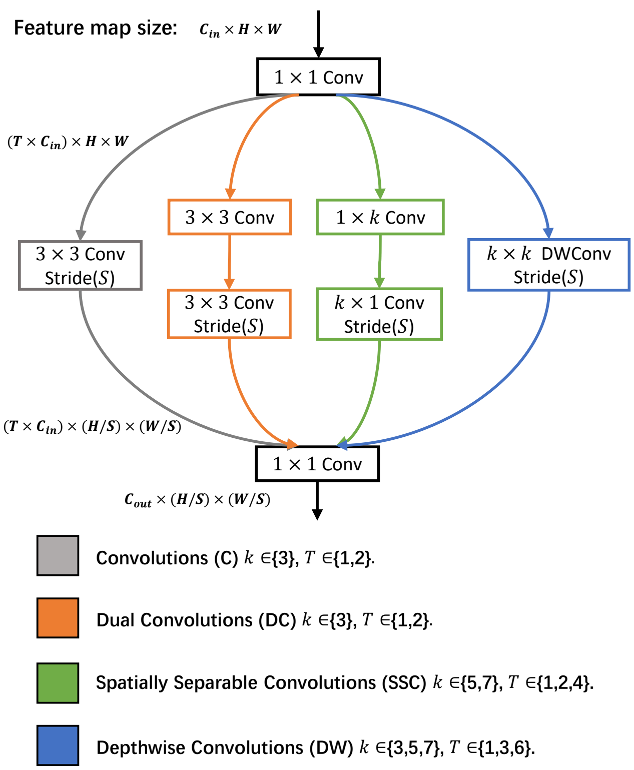


Figure 2. Latency Constraint search space details for all possible operations except Identity. These details are identical with the extended search space used by Li *et al*. [4]. The corresponding layer is recognized as a reduction layer when $C_{in} \neq C_{out}$. Each type of operation has different kernel size k or expand ratio T.

eration). The identity mapping path will always be sampled during supernet training and its path probability p is fixed to be 1 during fitness indicator updates.

A.4. Search Space Size Computation

For the case when we use 27 OPs space and layer-wise space size $K = 5$, the number of possible architectures

	Structure	Size
NASNet [16]	cell-based	7.1×10^{16}
Amoeba [9]	cell-based	5.6×10^{14}
ENAS [8]	cell-based	5.0×10^{12}
DARTS [7]	cell-based	2.4×10^{11}
Proxyless [1]	single-branch	3.0×10^{17}
SPOS [2]	single-branch	1.1×10^{12}
NSE $*$	multi-branch	1.4×10^{110}

Table 3. ImageNet NAS search space size compared. $*$ when we use 27 OPs space and $K = 5$.

$Comb_{arch}$ is computed as follows:

We denote the number of k-combinations given n elements as $C_k^n = \frac{n!}{k!(n-k)!}$. The number of possible combinations is $Comb_{norm} = \sum_{k=0}^{5} C_k^{27}$ for the normal layer and $Comb_{redu} = \sum_{k=1}^{5} C_k^{27}$ for the reduction layer. There are in total 16 normal layers and 6 reduction layers in FLOPs constrained macro architecture. Each layer has its own selected candidate operations. Thus the total number of possible architectures is $Comb_{arch} = (Comb_{norm})^{16} \times (Comb_{redu})^{6} \approx 1.4 \times 10^{110}$.

B. Details of Training Configs

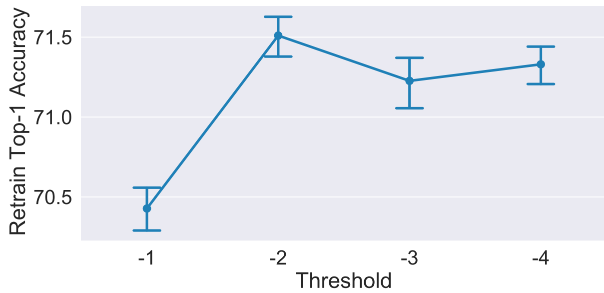
For every supernet training, we use Nesterov SGD with 0.9 momentum, weight decay $4e^{-5}$, batch size 1024 with 100 epochs. The initial learning rate is 0.4 and gradually reaches 0 through cosine learning rate decay with warm-up for 2 epochs . We use Adam optimizer with an initial learning rate of 0.1 to update fitness indicators, and we perform such updates every two supernet updates. Fitness indicators Θ are initialized to 0 and the corresponding pruning threshold is set to -2. While a larger random sample number D helps to find better Pareto front, limited by its computational cost, we set sample sizes as $D = 2000, D_e = 100$.

For the hyperparameters of the resource constraint regularization, we set $\alpha = 1e^{-5}, \beta = 2, \tau = 300$ for FLOPs (M) constraint and $\alpha = 2e^{-2}, \beta = 2, \tau = 7$ for Latency (ms) constraint. The α parameter for Latency constraint is set higher so that two constraints are of similar magnitude.

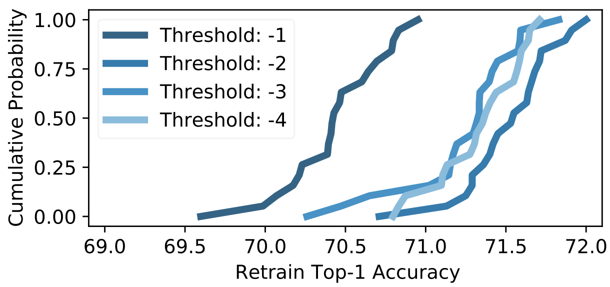
For model retraining, we increase the number of epochs to 350, with batch size 2048, learning rate 0.8, weight decay $4e^{-5}$ [4] for FLOPs constraint and $1e^{-4}$ [6] for Latency constraint, together with exponential moving average with decay 0.9999. For a fair comparison, swish activation, SE module together with identical training configs from EfficientNet [12] are optionally used subject to the specific settings.

C. Ablation on Pruning Threshold

To show how the trade-off between early and accurate search space simplification affects the optimized search



(a)



(b)

Figure 3. Comparison of 1-st round "aggregated" search space with respect to different pruning thresholds. "aggregated" denote the search space achieved by NSE after Pareto front aggregation. All results are based on identical search space initialization with layer-wise space size $K = 5$. For each "aggregated" search space, we randomly sample 20 architectures that have FLOPs within the interval of [323M, 327M]. Each model is then trained from scratch for 50 epochs to retrieve the retrain Top-1 accuracy illustrated above. (a) accuracies are shown in mean with 95% confidence intervals.

space to be inherited, we evaluate the quality of aggregated search space achieved by different pruning thresholds in Figure 3. As the threshold -1 is too close to 0 (the initialized value of fitness indicators Θ), its result is significantly worse when compared to lower thresholds. However, as the threshold is set lower than -2, the result seems saturated, and a lower threshold could even harm the quality of optimized search space.

D. Detection Result for NSENet

We have also evaluated our NSENet on object detection task. We take the pretrained NSENet as a drop-in replacement for the backbone feature extractor in EfficientDet-D0 [14]. Table 4 shows the performance of our NSENet, comparing with MobileNetV2 and the original backbone network EfficientNet-B0. We trained the network with identical configs as used by EfficientDet-D0. As shown in Table 4, our model significantly improves mAP score over MobileNetV2 and EfficientNet-B0 with fewer FLOPs.

Backbone	FLOPs	mAP
EfficientNet-B0 [12]	2.50B	33.8 [14]
MobileNetV2 1.0 [10]	2.24B	32.7
NSENet	**2.18B**	**34.5**

Table 4. NSENet object detection performance on COCO [5] dataset. All experiments adopt identical configs as used by EfficientDet-D0 [14] except backbone network.

E. Omitted Figures

Below we show the omitted figures. Figure 4 shows architecture details for final results. Figure 5 shows intermediate results of aggregated search space subset on 27 OPs space. Figure 6 illustrates the edging effect on Pareto frontier.

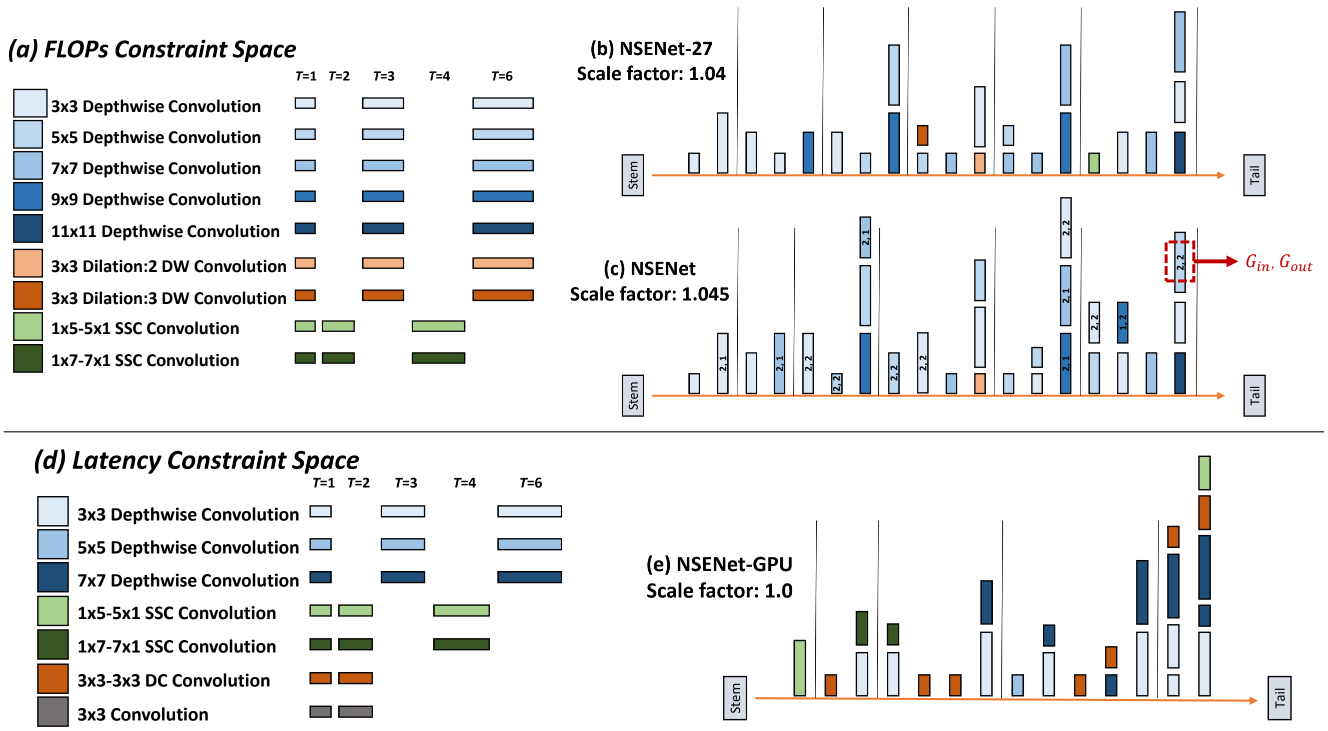


Figure 4. The detailed operations (a)(d) and structure (b)(c)(e) of our final results for FLOPs constriant and latency constraint, notice that (b) is the final result derived from 27 OPs space while (c) inherits the final search space subset derived from the 27 OPs space, then search on the second space as shown in Figure 1. The two numbers within the operation blocks shown in (c) represents the group number (G_{in}, G_{out}) of 1x1 projections. The width of the blocks correspond to the T in (a)(d) for candidate operation, which denotes the expand ratio of the corresponding operation, with details in Figure 1 and Figure 2. A straight line is put after every reduction layer in (b)(c) and (e). A "Scale Factor" [3] is used to adjust the amount of resource (*e.g.* FLOPs) consumed by the architecture by changing the number of channels uniformly. We can see that architectures searched under FLOPs constraint tend to go deeper while both constraints prefer efficient operations such as DW convolutions over less commonly used operations such as SSC convolutions.

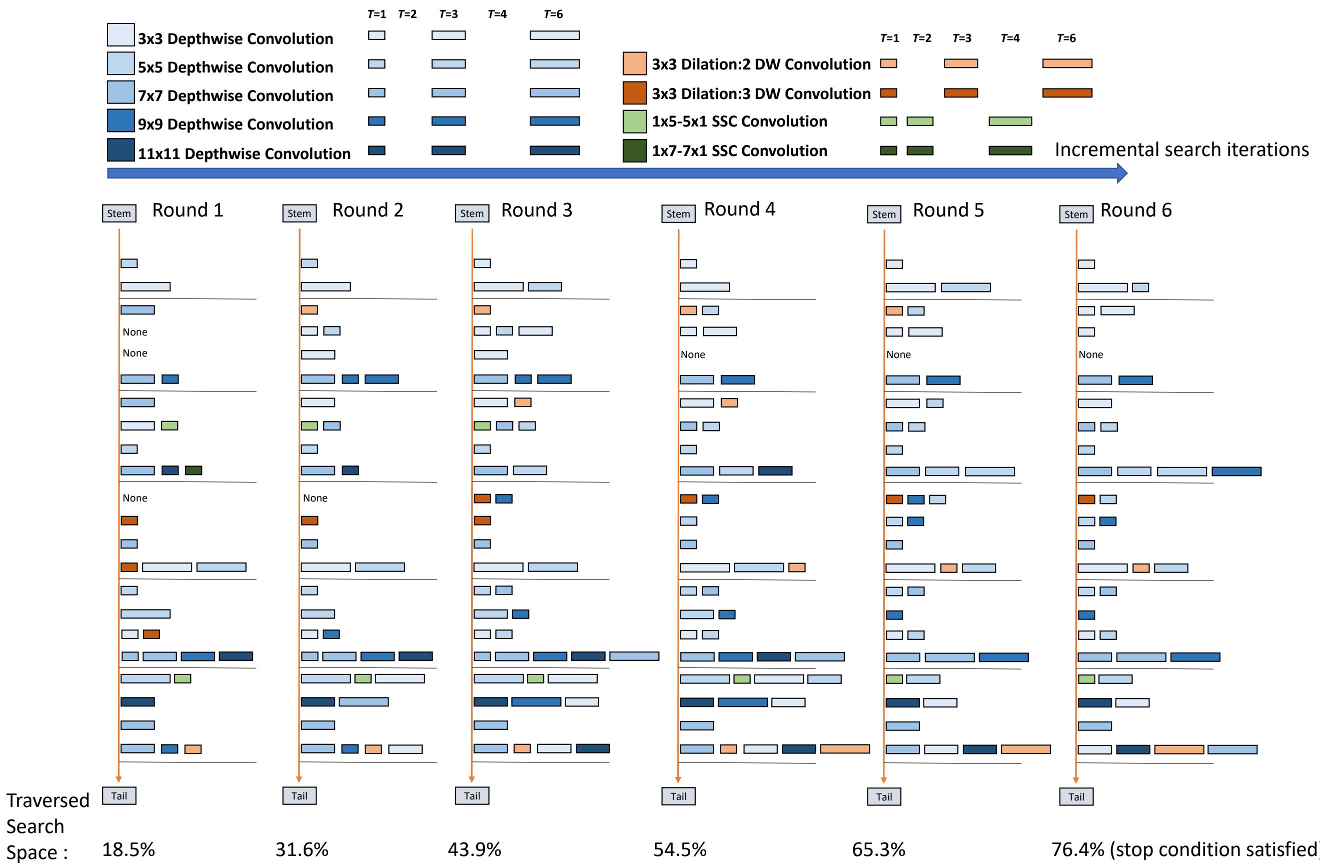


Figure 5. Intermediate results of the search space subset derived from Pareto front architecture aggregation. The results are based on the 27 OPs space and are from the same experiment where we get the NSENet-27 architecture. We can see that less commonly used operations such as SSC convolutions and dilated DW convolutions are seldom in the search space subset. On the other hand, most of the operations being included in the search space subset would last for multiple rounds or even till the final round, demonstrating the effectiveness of the proposed pipeline in terms of knowledge extraction and preservation.

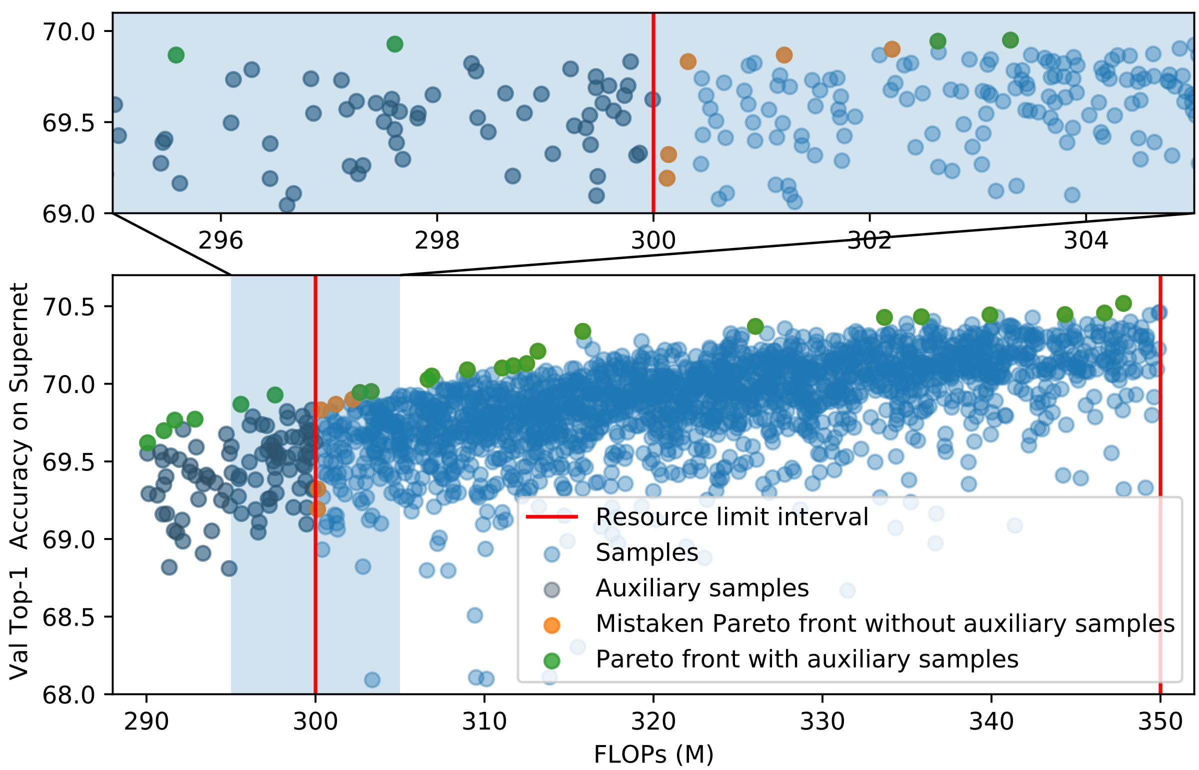


Figure 6. Edging effect in constrained Pareto frontier retrieval. When trying to get Pareto-optimal architectures only with the samples within the constraint interval, some of the samples (orange points in this figure) located close to the limit boundary (300M FLOPs) could be mistakenly considered as Pareto-optimal architectures. By considering auxiliary samples outside the limit interval, we can alleviate this issue. The data used in this figure is derived from the final round of search over 27 OPs space.

References

[1] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. *arXiv preprint arXiv:1812.00332*, 2018. 2

[2] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 544–560, 2020. 2

[3] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 4

[4] 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. 1, 2

[5] 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. 3

[6] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018. 2

[7] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018. 2

[8] Hieu Pham, Melody Y Guan, Barret Zoph, Quoc V Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. *arXiv preprint arXiv:1802.03268*, 2018. 2

[9] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. *arXiv preprint arXiv:1802.01548*, 2018. 2

[10] 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. 3

[11] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 1

[12] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019. 2, 3

[13] Mingxing Tan and Quoc V Le. Mixconv: Mixed depthwise convolutional kernels. *CoRR, abs/1907.09595*, 2019. 1

[14] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10781–10790, 2020. 3

[15] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. *arXiv preprint arXiv:1812.03443*, 2018. 1

[16] 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. 2