A APPENDIX

We organize our appendix as follows. We explain the details of using CafeNet for training and searching in appendix A.1. In appendix A.2, we give details of our experiments w.r.t. different datasets. We report the implementation details of our CafeNet with Algorithm 1 in appendix A.3. Then, we introduce the corresponding relationship between names and references in appendix A.4. And we explore the efficiency of CafeNet in training one-shot supernet $\mathcal{N}$ A.5. We show the performance gain of each part of CafeNet in appendix A.6. We examine the effect of FLOPs-sensitive bins in appendix A.7. We explore more detailed experiments of ImageNet, CIFAR-10, and CelebA dataset in appendix A.8 and A.9. Then we show the visualization of searched network width in appendix A.11. In appendix A.12, we visualize the configuration of bins with EfficientNet-B0 and MobileNetV2. In addition, we investigate the effect of min-min optimization with MobileNetV2 and ResNet50 in appendix A.13. We study the impact of the multi-stage search with evolving bins on CIFAR-10 dataset in appendix A.14. We provide the estimation of the size of the search space in the multi-stage search in appendix A.15. Finally, the effect of the smallest bin size β in FLOPs-sensitive bins is shown in appendix A.16.

A.1 TRAINING AND SEARCHING WITH CAFENET

Training with min-min optimization. With the bin-based search space, we can train our CafeNet in a stochastic setting as Eq.(1), which simply samples a network width $\boldsymbol{c}$ first, and then optimizes the corresponding sub-network. Nevertheless, in CafeNet, a width is specified more freely by several sub-networks, and the performance of these sub-networks can be different to a great extent. According to this, we propose to indicate the performance of width by examining its sub-network with the best performance. Concretely, during the training of CafeNet, instead of randomly optimizing a sub-network, we optimize the sub-network with the smallest training loss. For a network width $\boldsymbol{c}$, we denote its corresponding sub-network set as $\mathbb{S}_{\boldsymbol{c}}$, thus for the CafeNet $\mathcal{N}$ with weights W, the training target of min-min optimization can be written as,

$$W^* = \arg\min_{W} \mathop{\mathbb{E}}_{\boldsymbol{c}\in U(\mathcal{C})} \left[\mathcal{L}_{train}(\boldsymbol{w}_{s^*_{\boldsymbol{c}}}; \mathcal{N}, \boldsymbol{c}, \mathcal{D}_{tr})\right] \tag{11}$$

$$\text{s.t. } s^*_{\boldsymbol{c}} = \arg\min_{s\in\mathbb{S}_{\boldsymbol{c}}} \mathcal{L}_{train}(\boldsymbol{w}_s; \mathcal{N}, \boldsymbol{c}, \mathcal{D}_{tr}), \tag{12}$$

where $w_s \subset W$, denotes the weights of subnet s. To find the optimal sub-network $s^*_{\boldsymbol{c}}$, it needs to traverse all sub-networks and results in additional computational cost. However, this can be efficiently implemented by calculating the loss of all networks without backpropagation (*i.e.*, *no_grad* mode in PyTorch). And then, we only need to backward once using the target sub-network $s^*_{\boldsymbol{c}}$.

Searching with max-max selection. After the CafeNet $\mathcal{N}$ is trained, we can evaluate each network width by examining its performance (*e.g.* classification accuracy) on the validation dataset $\mathcal{D}_{val}$. Similar to the training of CafeNet, for a network width $\boldsymbol{c}$, we use the sub-network with the highest performance to indicate its performance. Then the searching amounts to a max-max selection problem

$$\boldsymbol{c}^* = \arg\max_{\boldsymbol{c}\in\mathcal{C}} \max_{s\in\mathbb{S}_{\boldsymbol{c}}} \text{Accuracy}(\boldsymbol{w}^*_s, W^*; \mathcal{N}^*, \mathcal{D}_{val}), \text{ s.t. FLOPs}(\boldsymbol{c}) \le F_b. \tag{13}$$

Note that the inner $\max$ also needs to calculate the validation accuracy of all sub-networks. Nevertheless, since searching itself is much faster than the supernet training, the increased computational cost is subtle and acceptable in real practice.

A.2 DETAILS OF EXPERIMENTAL SETTINGS

In this section, we present the implementation details of our CafeNet w.r.t. experiments on various datasets. In general, for most networks except EfficientNet-B0, we use a SGD optimizer with momentum 0.9. While for EfficientNet-B0, we use RMSprop optimizer with epsilon set to 0.001. The parameters of β and α are initialized to 1 and 2, respectively, to determine the search space. For evolutionary search, we set the population and iteration size to 40 and 50, respectively. Besides, 2k network width are sampled for a random search. After searching, the optimized network width is trained from scratch for evaluation.

ImageNet and CIFAR-10 dataset. we set weight decay to 10^{-4} for ResNet50 (same setting is also adopted for ResNet34 and ResNet18) and VGGNet while 5×10^{-5} for MobileNetV2. The learning

rate is decayed with cosine strategy from 0.1 to 10^{-5} for MobileNetV2 and ResNet50 with 300 training epochs and a mini-batch size of 256. Besides, for ResNet50, we adopt the strategy of random erasing with a probability of 0.4. For VGGNet, we train for 400 epochs with batch size 128, and the learning rate is decayed by step strategy from 0.1 and divided by 10 at 150-th, 225-th, 300-th epoch. For EfficientNet-B0, the learning rate is initialized to 0.13 and decayed by 0.96 for every 3 epochs. We train EfficientNet-B0 for 300 epochs, of which the first 5 are warm-up epochs, and the weight decay is set to 1×10^{-5}.

CelebA dataset. We set the training epochs to 15 (10) for MobileNetV2 and ResNet18 with learning rate annealed from 0.1 to 10^{-5} by cosine strategy, and the batch size of these two networks are set to 64.

MS COCO dataset. We conduct object detection experiments using two popular frameworks Faster R-CNN with Feature Pyramid Networks (FPN) (Lin et al., 2017a) and RetinaNet (Lin et al., 2017b) on MS COCO dataset (Everingham et al., 2010). The *trainval35k* split is used for training and we report the mean Average Precision (mAP) on *minival* split. We train all the models using SGD for 12 epochs from ImageNet pretrained weights, and the initial learning rate is set to 0.08 with batch size 256, which decays 0.1 at 8-th and 11-th epoch.

A.3 Algorithm of CafeNet

The details about CafeNet are presented in Algorithm 1. In specific, we search network width on CIFAR-10 dataset with 3 stages while others with 1 stage.

Algorithm 1 Locally free weight sharing for network width search

Input: The number of multi-stage $\mathcal{M}$. The training epochs of supernet $\mathcal{N}$. The smallest bin size β and the bin evloving speed α. Searching methods $\mathcal{A}$. Training dataset $\mathcal{D}_{tr}$. Validation dataset $\mathcal{D}_{val}$.

while stage $< \mathcal{M}$ **do**
  initialize the supernet $\mathcal{N}$ and bins groups.
  while epochs $< \mathcal{E}$ **do**
    randomly assign the training and backwards path
    updating the corresponding path of bin groups with dataset $\mathcal{D}_{tr}$.
  end while
  update $\beta = \beta \ / \ \alpha$
end while
search the optimized network width within supernet $\mathcal{N}$ by searching methods $\mathcal{A}$ (as illustrated in section 3.3).
train the optimized network width from scratch for evaluation.

Output: the optimized network width with evaluation results

A.4 Index and literature

In order to simplify the correspondence between names and references in all tables, we have listed the names used and their corresponding papers as follows:

Table 7: The names used in tables and their corresponding papers.

Names	papers	Names	papers	Names	papers
ResNet	(He et al., 2016)	AutoSlim	(Yu & Huang, 2019)	Rethinking	(Liu et al., 2019b)
MetaPruning	(Liu et al., 2019a)	MIL	(Dong et al., 2017)	LEGR	(Chin et al., 2019)
PF	(Li et al., 2016)	GBN	(You et al., 2019)	CNN-FCF	(Li et al., 2019)
SFP	(He et al., 2018a)	LEGR	(Chin et al., 2019)	FPGM	(He et al., 2019a)
TAS	(Dong & Yang, 2019)	GS	(Ye et al., 2020)	AutoPruner	(Luo & Wu, 2018)
CGNet	(Hua et al., 2019)	MobileNetV2	(Sandler et al., 2018)	NA	(Chen et al., 2020)
MFP	(He et al., 2019b)	AMC	(He et al., 2018b)	DMCP	(Guo et al., 2020)
MuffNet	(Chen et al., 2019)	DCP	(Zhuang et al., 2018)	FBS	(Gao et al., 2018)
VGGNet	(Simonyan & Zisserman, 2014)	GAL	(Lin et al., 2019)	Sliming	(Liu et al., 2017)
PS	(Wang et al., 2019)	AOFP	(Ding et al., 2019)		

A.5 THE EFFICIENCY OF CAFENET IN TRAINING ONE-SHOT SUPERNET $\mathcal{N}$

To investigate the searching efficiency of CafeNet, we examine the time cost of training 1 epoch with different radius r. As shown in Table 8, the training time of CafeNet raised with an increase of r, which is because r represents the preset allowed offset for free channels, thus increasing the times of forwarding for each batch of information. However, since CafeNet involves multi-times forward and once backward and the time cost of forwarding is much shorter than the backward, the overall training efficiency will not be greatly affected. To balance the performance and search efficiency, we set $r = 1$ for all experiments with CafeNet.

Table 8: The efficiency of CafeNet in training one-shot supernet $\mathcal{N}$.

CIFAR-10			ImageNet		
Local free	MobileNetV2	VGGNet	Local free	MobileNetV2	ResNet50
r=0	51.3s	61.6s	r=0	428.4s	748.1s
r=1	61.1s	72.4s	r=1	520.6s	954.3s
r=2	95.2s	109.7s	r=2	832.5s	1569.6s
r=3	203.6s	258.4s	r=3	1872.3s	3584.9s

A.6 PERFORMANCE GAIN OF EACH PART IN CAFENET

To investigate the performance gain of each part in CafeNet, we conduct experiments to search for 0.5× FLOPs MobileNetV2 and 0.5× FLOPs VGGNet on CIFAR-10 dataset and report their Top-1 accuracy. Since min-min optimization and max-max selection are always used simultaneously, for clarity, we abbreviate them as min-min optimization. As shown in Table 9, with evolutionary search, CafeNet can enjoy a performance gain of 0.91% (0.74%) accuracy on MobileNetV2 (VGGNet). Besides, similar results can also be obtained through the greedy search, which exactly shows that our CafeNet can boost to search for a decent network width.

Table 9: Performance of searched 0.5× FLOPs MobileNetV2 and VGGNet on CIFAR-10 dataset with different supernets and searching methods.

supernet			searching			models	
FLOPs-sensitive bins	Locally free weight sharing pattern	min-min optimization	multi-stage search	greedy search	evolutionary search	MobileNetV2	VGGNet
				✓		94.27%	93.49%
✓				✓		94.51%	93.65%
✓	✓	✓		✓		94.76%	93.89%
					✓	94.53%	93.62%
✓					✓	94.71%	93.74%
✓	✓	✓			✓	95.03%	93.95%
✓	✓	✓	✓		✓	95.44%	94.36%

A.7 EFFECT OF FLOPS-SENSITIVE BINS

Our proposed FLOPs-sensitive bins aim to reduce the search space by allowing the FLOPs to distribute more evenly over layers for the search unit. To explore the effect of FLOPs-sensitive bins as a search unit compared with the uniform unit (uniform channel group), we search for network width under these two search units on CIFAR-10 dataset with MobileNetV2 and VGGNet, as shown in Table 10. In detail, compared with the uniform unit, higher accuracy under various FLOPs budgets can be achieved by leveraging FLOPs-sensitive bins.

Table 10: Performance comparison of two forming search unit methods, *i.e.*, baseline uniform groups, and FLOPs-sensitive bins.

VGGNet			MobileNetV2		
FLOPs	Uniform groups	**FLOPs-sensitive bins**	FLOPs	Uniform groups	**FLOPs-sensitive bins**
189M	94.12%	**94.36%**	188M	95.37%	**95.56%**
154M	93.87%	**94.23%**	144M	95.12%	**95.44%**
115M	93.64%	**94.01%**	44M	94.99%	**95.31%**
76M	93.25%	**93.67%**	28M	93.62%	**94.11%**

A.8 MORE DETAILED RESULTS OF IMAGENET DATASET FOR TABLE 1

To explore the effectiveness of CafeNet, we also implement our algorithm on ResNet34 and ResNet18 with the same training strategy as ResNet50. The original ResNet34 and ResNet18 has 21.8M, 11.7M and 3.6G, 1.8G FLOPs with 74.9%, 71.5% Top-1 accuracy, respectively. The detailed results are reported in Table 11, our 0.75× FLOPs ResNet34 and ResNet18 can achieve close performance to the origin model. Besides, under the tiny FLOPs budget (*i.e.*, 10% FLOPs), CafeNet can outperform the unform baseline by a large margin.

Table 11: Performance comparison of ResNet50, ResNet34, ResNet18 and MobileNetV2 on ImageNet. References of baseline methods are summarized in appendix A.4.

ResNet50					
	Methods	FLOPs	Param	Top-1	Top-5
3G	AutoSlim	3.0G	23.1M	76.0%	-
	MetaPruning	3.0G	-	76.2%	-
	LEGR	3.0G	-	76.2%	-
	Uniform	3.0G	19.1M	75.9%	93.0%
	Random	3.0G	-	75.2%	92.5%
	CafeNet-R	3.0G	22.6M	**77.1%**	94.3%
	CafeNet-E	3.0G	23.8M	**77.4%**	94.5%
2G	GBN	2.4G	31.8M	76.2%	92.8%
	SFP	2.4G	-	74.6%	92.1%
	LEGR	2.4G	-	75.7%	92.7%
	FPGM	2.4G	-	75.6%	92.6%
	TAS	2.3G	-	76.2%	93.1%
	MetaPruning	2.0G	-	75.4%	-
	AutoSlim	2.0G	20.6M	75.6%	-
	Uniform	2.0G	13.3M	75.1%	92.7%
	Random	2.0G	-	74.6%	92.2%
	CafeNet-R	2.0G	19.1M	**76.5%**	93.1%
	CafeNet-E	2.0G	18.4M	**76.9%**	93.3%
1G	AutoPruner	1.4G	-	73.1%	91.3%
	MetaPruning	1.0G	-	73.4%	-
	AutoSlim	1.0G	-	74.0%	-
	Uniform	1.0G	6.6M	73.1%	91.8%
	Random	1.0G	-	72.2%	91.4%
	CafeNet-R	1.0G	11.2M	**74.9%**	92.3%
	CafeNet-E	1.0G	12M	**75.3%**	92.6%
570M	AutoSlim	570M	-	72.2%	-
	Uniform	570M	4.0M	71.6%	90.6%
	Random	570M	-	69.4%	90.3%
	CafeNet-R	570M	11.3M	**72.7%**	90.9%
	CafeNet-E	570M	12.0M	**73.3%**	91.2%

ResNet34					
	Methods	FLOPs	Param	Top-1	Top-5
2.7G	Rethinking	2.79G	-	72.9%	-
	MIL	2.75G	-	73.0%	-
	PF	2.79G	-	72.1%	-
	Uniform	2.7G	-	72.2%	90.9%
	Random	2.7G	-	71.6%	90.8%
	CafeNet-R	2.7G	18.6M	**74.4%**	92.0%
	CafeNet-E	2.7G	20.4M	**74.8%**	92.3%
	CNN-FCF	2.7G	15.9M	73.6%	91.5%
	CafeNet-R	2.5G	19.3M	**74.0%**	91.9%
	CafeNet-E	2.5G	20.2M	**74.5%**	92.1%
1.8G	FPGM	2.2G	-	72.5%	-
	SFP	2.2G	-	71.8%	90.3%
	CNN-FCF	2.2G	12.6M	72.8%	91.0%
	GS	2.1G	-	72.9%	-
	Uniform	1.8G	-	71.6%	90.3%
	Random	1.8G	-	71.1%	89.9%
	CafeNet-R	1.8G	17.2M	**73.1%**	91.4%
	CafeNet-E	1.8G	16.9M	**73.4%**	91.5%
	CGNet	1.8G	-	71.3%	-
	CNN-FCF	1.7G	9.6M	71.3%	90.2%
1G-	CNN-FCF	1.2G	7.1M	69.7%	89.3%
	CGNet	1.2G	-	70.2%	-
	Uniform	0.9G	-	69.5%	89.4%
	Random	0.9G	-	69.1%	88.9%
	CafeNet-R	0.9G	10.1M	**71.8%**	89.5%
	CafeNet-E	0.9G	9.8M	**72.1%**	89.8%
	Uniform	0.36G	-	59.9%	82.3%
	Random	0.36G	-	56.2%	80.6%
	CafeNet-R	0.36G	3.5M	**63.3%**	85.2%
	CafeNet-E	0.36G	3.6M	**64.0%**	85.4%

MobileNetV2					
	Methods	FLOPs	Param	Top-1	Top-5
200M	MetaPruning	217M	-	71.2%	-
	LEGR	210M	-	71.4%	-
	AutoSlim	207M	4.1M	73.0%	-
	Uniform	217M	2.7M	70.9%	89.4%
	Random	217M	-	70.3%	89.1%
	CafeNet-R	217M	3.0M	**73.3%**	91.1%
	CafeNet-E	217M	3.3M	**73.4%**	91.2%
150M	LEGR	150M	-	70.8%	-
	TAS	150M	-	70.9%	-
	AMC	150M	-	70.8%	-
	LEGR	150M	-	69.4%	-
	MuffNet	149M	-	63.7%	-
	Uniform	150M	2.0M	69.3%	88.9%
	Random	150M	-	68.8%	88.7%
	CafeNet-R	150M	2.7M	**71.9%**	90.0%
	CafeNet-E	150M	3.0M	**72.4%**	90.4%
100M	MetaPruning	105M	-	65.0%	-
	Uniform	105M	1.5M	65.1%	89.6%
	Random	105M	-	63.9%	89.2%
	CafeNet-R	106M	2.2M	**68.2%**	88.2%
	CafeNet-E	106M	2.1M	**68.7%**	88.5%
	MuffNet	50M	-	50.3%	-
	MetaPruning	43M	-	58.3%	-
	Uniform	50M	0.9M	59.7%	82.0%
	Random	50M	-	57.4%	81.2%
	CafeNet-R	50M	1.7M	**64.3%**	85.2%
	CafeNet-E	50M	1.6M	**64.9%**	85.4%

ResNet18					
	Methods	FLOPs	Param	Top-1	Top-5
1.2G	TAS	1.2G	-	69.2%	89.2%
	MIL	1.2G	-	66.3%	86.9%
	Uniform	1.2G	8.5M	68.8%	88.5%
	Random	1.2G	-	68.4%	88.1%
	CafeNet-R	1.2G	11.5M	**70.8%**	89.8%
	CafeNet-E	1.2G	11.3M	**71.2%**	89.9%
1G	NA	1.17G	-	69.4%	88.71%
	SFP	1.05G	-	67.1%	87.8%
	MFP	1.05G	-	68.3%	88.3%
	DMCP	1.04G	-	69.2%	-
	FPGM	1.04G	-	68.4%	88.5%
	DCP	0.96G	-	67.4%	87.6%
	CGNet	0.94G	-	68.8%	-
	MFP	0.9G	-	67.1%	87.5%
	FBS	0.9G	-	68.2%	88.2%
	Uniform	0.9G	6.0M	67.1%	87.5%
	Random	0.9G	-	66.7%	87.1%
	CafeNet-R	0.9G	9.7M	**69.6%**	88.8%
	CafeNet-E	0.9G	10.2M	**69.8%**	89.0%
0.45G-	Uniform	450M	2.9M	61.6%	83.6%
	Random	450M	-	59.8%	82.3%
	CafeNet-R	450M	5.0M	**65.2%**	86.0%
	CafeNet-E	450M	5.8M	**65.6%**	86.2%
	Uniform	180M	1.1M	53.7%	77.5%
	Random	180M	-	51.6%	76.9%
	CafeNet-R	180M	1.9M	**57.8%**	81.7%
	CafeNet-E	180M	2.0M	**58.4%**	81.9%

A.9 MORE DETAILED RESULTS OF CIFAR-10 AND CELEBA DATASET FOR TABLE 4 AND 5

To examine the hyperparameters setting with CIFAR-10 and CelebA dataset, we also explore the searching results from two vanilla baselines named Random and Uniform, as described in Section 4. The detailed results on CIFAR-10 dataset and CelebA dataset are reported in Table 12 and Table 13, respectively. Besides, for CelebA dataset, the layer widths of fully-connection(FC) layers are not implemented to be searched and keep the same as (Sener & Koltun, 2018), since they only takes a tiny amount of FLOPs to the network.

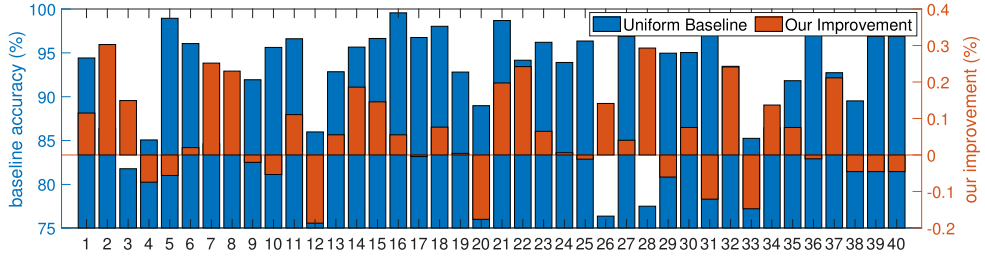
Table 12: Performance comparison of MobileNetV2 and VGGNet on CIFAR-10 dataset.

MobileNetV2					VGGNet				
Groups	Methods	FLOPs	Parameters	Accuracy	Groups	Methods	FLOPs	Parameters	Accuracy
200M	DCP	218M	1.7M	94.75%	200M	GAL	190M	-	93.80%
	Uniform	188M	1.4M	94.57%		DCP	199M	10.4M	94.16%
	Random	188M	-	94.20%		Sliming	199M	10.4M	93.80%
	CafeNet-R	188M	1.4M	**95.44%**		Uniform	189M	9.5M	93.37%
	CafeNet-E	188M	1.5M	**95.56%**		Random	189M	-	93.06%
144M	MuffNet	175M	-	94.71%		**CafeNet-R**	189M	8.3M	**94.27%**
	Uniform	144M	1.1M	94.28%		**CafeNet-E**	189M	8.0M	**94.36%**
	Random	144M	-	93.76%	100M+	PS	156M	-	93.63%
	CafeNet-R	144M	1.2M	**95.28%**		Uniform	154M	7.7M	93.11%
	CafeNet-E	144M	1.1M	**95.44%**		Random	154M	-	92.86%
44M	AutoSlim	88M	1.5M	93.20%		**CafeNet-R**	154M	3.4M	**94.09%**
	AutoSlim	59M	0.7M	93.00%		**CafeNet-E**	154M	3.1M	**94.23%**
	MuffNet	45M	-	93.12%		AOFP	124M	-	93.84%
	Uniform	44M	0.3M	92.72%		Uniform	115M	5.9M	92.95%
	Random	44M	-	92.24%		Random	115M	-	92.77%
	CafeNet-R	44M	0.4M	**95.16%**		**CafeNet-R**	115M	2.4M	**93.87%**
	CafeNet-E	44M	0.4M	**95.31%**		**CafeNet-E**	115M	2.1M	**94.01%**
28M	AutoSlim	28M	0.3M	92.00%	76M	CGNets	92M	-	92.88%
	Uniform	28M	0.3M	91.87%		Uniform	76M	3.9M	92.32%
	Random	28M	-	91.36%		Random	76M	-	91.67%
	CafeNet-R	28M	0.2M	**93.87%**		**CafeNet-R**	76M	2.2M	**93.36%**
	CafeNet-E	28M	0.2M	**94.11%**		**CafeNet-E**	76M	1.4M	**93.67%**

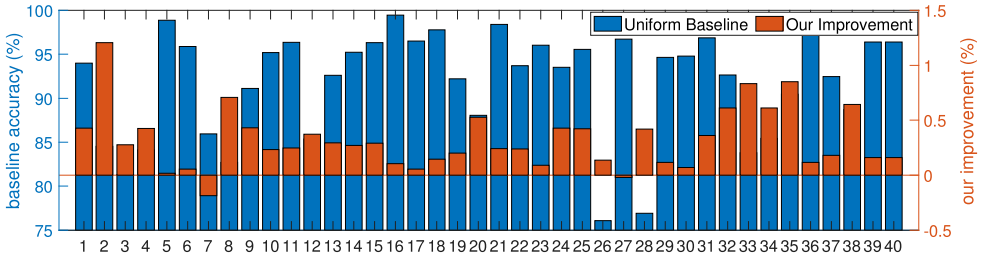
Table 13: Performance comparison of MobileNetV2 and ResNet18 on CelebA dataset.

MobileNetV2				ResNet18			
Methods	FLOPs	Parameters	Accuracy	Methods	FLOPs	Parameters	Accuracy
Uniform	162M	-	91.97%	Uniform	1G	-	92.03%
Random	162M	-	91.76%	Random	1G	-	91.84%
CafeNet-R	162M	1.7M	**92.12%**	**CafeNet-R**	1G	1.7M	**92.17%**
CafeNet-E	162M	1.8M	**92.19%**	**CafeNet-E**	1G	1.8M	**92.25%**
Uniform	106M	-	91.92%	Uniform	619M	-	91.93%
Random	106M	-	91.63%	Random	619M	-	91.67%
CafeNet-R	106M	1.2M	**92.09%**	**CafeNet-R**	619M	0.4M	**92.13%**
CafeNet-E	106M	1.2M	**92.16%**	**CafeNet-E**	619M	0.5M	**92.18%**
Uniform	51M	-	91.73%	Uniform	316M	-	91.79%
Random	51M	-	91.52%	Random	316M	-	91.62%
CafeNet-R	51M	0.6M	**92.03%**	**CafeNet-R**	316M	3.1M	**92.07%**
CafeNet-E	51M	0.5M	**92.13%**	**CafeNet-E**	316M	3.4M	**92.16%**
Uniform	21M	-	91.63%	Uniform	130M	-	91.66%
Random	21M	-	91.42%	Random	130M	-	91.51%
CafeNet-R	21M	0.2M	**91.71%**	**CafeNet-R**	130M	1.2M	**91.83%**
CafeNet-E	21M	0.2M	**91.85%**	**CafeNet-E**	130M	1.0M	**91.92%**

Classification results for all 40 labels on CelebA dataset. As shown in Fig.3, our searched network width has better performance in comparing to the uniform baseline, especially for small FLOPs budget and labels that are difficult to be classified.



(a) Performance of 50% FLOPs MobileNetV2 w.r.t.40 labels on CelebA dataset.



(b) Performance of 10% FLOPs MobileNetV2 w.r.t.40 labels on CelebA dataset.

Figure 3: Performance of 50% MobileNetV2 w.r.t.40 labels on CelebA dataset. The blue bar refers to the performance of network width searched by CafeNet, while the red bar indicates the performance gap in comparison to the uniform baseline.

A.10 MORE EXPERIMENTAL RESULTS WITH ALIGNED TRAINING RECIPES OF AUTOSLIM YU & HUANG (2019)

To examine the performance of our searched network width, we retrain the searched width of MobileNetV2 and ResNet50 with aligned training recipes of AutoSlim Yu & Huang (2019). In detail, we report the retraining results of width with similar FLOPs to AutoSlim in Table 14.

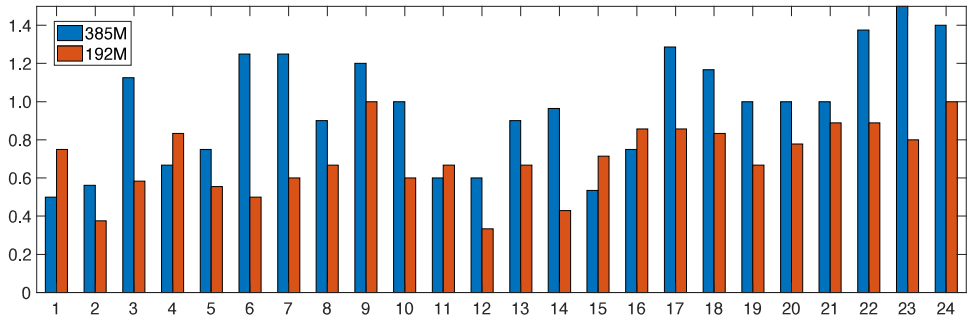
Table 14: Performance of MobileNetV2 and ResNet50 with aligned training recipes of AutoSlim Yu & Huang (2019).

ResNet50				
Methods	FLOPs	Parameters	Top-1	Top-5
AutoSlim	3.0G	23.1M	76.0%	-
CafeNet-R	3.0G	22.6M	76.2%	93.1%
CafeNet-E	3.0G	23.8M	76.4%	93.3%
AutoSlim	1G	-	74.0%	-
CafeNet-R	1G	11.2M	74.2%	91.9%
CafeNet-E	1G	12M	74.5%	92.1%
AutoSlim	570M	-	72.2%	-
CafeNet-R	570M	11.3M	72.1%	90.7%
CafeNet-E	570M	12M	72.6%	91.0%
MobileNetV2				
AutoSlim	207M	4.1M	73.0%	-
CafeNet-R	217M	3.0M	73.1%	90.9%
CafeNet-E	217M	3.3M	73.2%	91.1%

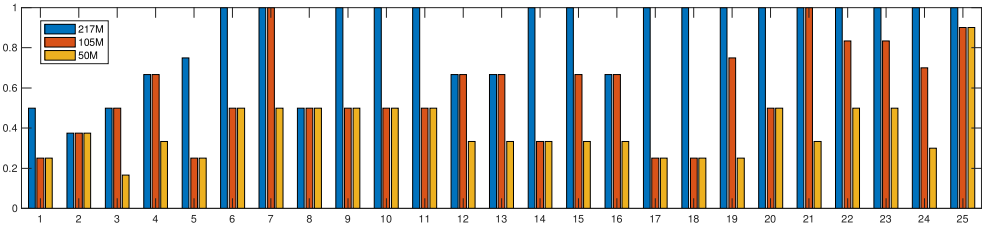
As Table 14 shows, with similar FLOPs and the same training recipes, CafeNet can search the width with higher accuracy, which proves the effectiveness of our method.

A.11 VISUALIZATION OF SEARCHED NETWORK WIDTH

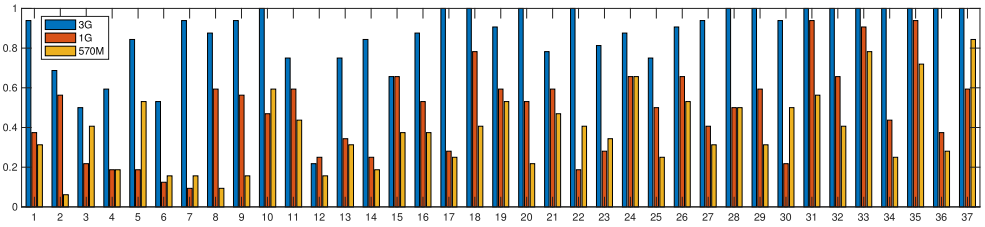
For intuitively understanding, we visualize our searched networks with ImageNet dataset, as shown in Fig.4. For clarity, we show the retained ratio of network width from the original model. Note that, EfficientNet-B0, ResNet50, and MobileNetV2 have SE block, skipping, or depthwise layers; we merged these layers, which are required to have the same network width for visual clarity. For EfficientNet-B0, we also did not show the channel of the SE block because these layers are fixed during the network width search.



(a) Network width of searched EfficientNet-B0 on ImageNet dataset.



(b) Network width of searched MobileNetV2 on ImageNet dataset.



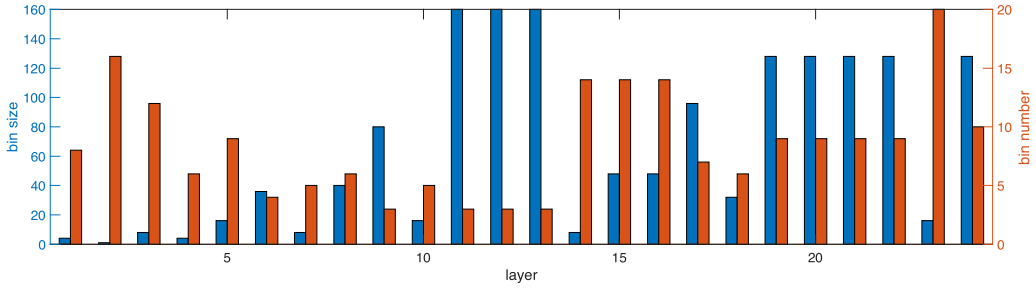
(c) Network width of searched ResNet50 on ImageNet dataset.

Figure 4: Visualization of Pruned models.

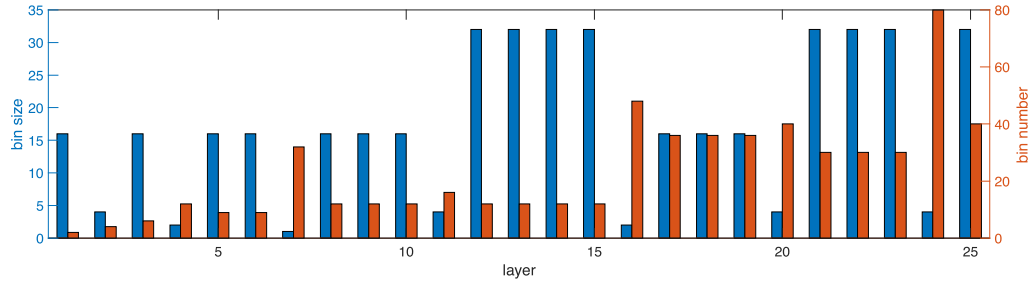
For ImageNet dataset, from Fig.4, we can see that layer width in skipping layers are more inclined to be preserved, which means skipping layers are very useful for classification. Besides, with a large FLOPs budget, width in the $1\times$ 1 convolution layers shrink more than in $3\times$ 3 convolution layers, which means $1\times$ 1 convolution may contribute less to classification performance. However, we observe an opposite phenomenon for width pruned from small FLOPs budgets, which implies the network is forced to use more $1\times$ 1 convolution layers to extract information from feature maps instead of $3\times$ 3 convolution layers.

A.12 VISUALIZATION OF BINS

For intuitively understanding of FLOPs-sensitive bins, as shown in Fig.5, we visualize the number of channels contained in each bin (bin size) and the number of bins in each layer (bin number) with MobileNetV2 and VGGNet. Similar to the visualization of the searched network width, we merged the SE blocks, skipping, and depthwise layers for clarity. To make the bins from all layers have a similar contribution to FLOPs, the number of bins varies significantly between layers. For EfficientNet-B0, the size of bin changes according to the different selected blocks (*i.e.*, blocks with 3×3 or 5×5 convolution kernel size), *i.e.*, the convolution of 3×3 or 5×5, and the size of bin becomes larger at the middle and the end of the network. For MobileNetV2, since only 3×3 convolution is involved, the bin size changes more regularly, and the number of bins gradually increases from the front to the end of the network.



(a) EfficientNet-B0



(b) MobileNetV2

Figure 5: Visualization of bin size and bin number of CafeNet.

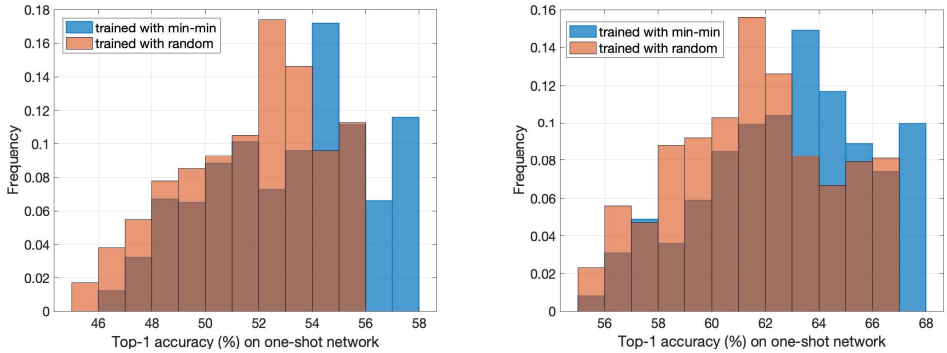
A.13 EFFECT OF MIN-MIN OPTIMIZATION IN EQ.(11)~(13)

To train the supernet, we investigate random optimization and min-min optimization with 50% FLOPs MobileNetV2 and ResNet50 on ImageNet dataset, for instance. In detail, based on the trained supernet, we use the evolution search to obtain the optimized network width and record 1000 network width for both random optimization and min-min optimization, and report the results by a statistical distribution histogram, as shown in Fig.6. During sampling, the average performance of network width with our min-min optimization achieved 62.4% (52.7%) Top-1 accuracy with ResNet50 (MobileNetV2), which is 1.4% (1.3%) better than results from random optimization. In addition, width with the best performance is trained from scratch for evolution, as shown in Table 15.

Table 15: Performance comparison of weights updating methods.

0.5×ResNet50			0.5×MobileNetV2		
Method	Sampling	CafeNet	Method	Sampling	CafeNet
Random	67.14%	76.34%	Random	56.22%	71.97%
min-min	**68.56%**	**76.92%**	**min-min**	**58.37%**	**72.43%**

In Table 15, we record the accuracy of Top-1 performance network width in supernet as Sampling, and than we training from scratch the width to report its Top-1 accuracy as CafeNet.



(a) Histogram of Top-1 accuracy by evolutionary with MobileNetV2.

(b) Histogram of Top-1 accuracy by evolutionary with ResNet50.

Figure 6: Histogram of Top-1 accuracy of network width trained from min-min optimization and random optimization by evolutionary w.r.t. MobileNetV2 and ResNet50.

A.14 Effect of multi-stage search with evolving bins

To examine the effect of multi-stage search with evolving bins, we report the Top-1 accuracy of MobileNetV2 on CIFAR-10 dataset with different training stages and the bin evolving speed α in Section 3.4, the results are summarized in Table 16. In detail, for CIFAR-10 dataset, our algorithm achieves superior performance with α set to 2 and searching in 3 stages, which indicates bin evolving strategy can lead to better results comparing with searching in one stage. However, when α is increased to 4, the performance of the searching results decreases a bit, which may be due to the larger search space that limits the ability of evolutionary search.

Table 16: Performance of 50% FLOPs MobileNetV2 and VGGNet on CIFAR-10 dataset with different training stages and bin evolving speed α.

Number of training stages	Bin evolving speed α	VGGNet	MobileNetV2
1	-	93.95%	95.03%
2	1	94.22%	95.31%
2	2	94.29%	95.41%
2	4	94.18%	95.29%
3	1	94.29%	95.36%
3	2	94.36%	95.44%
3	4	94.21%	95.31%

A.15 Estimation of the size of search space in multi-stage search

Note that by leveraging the multi-stage searching strategy, we can further reduce the search space by following the searching "from coarse-grained to fine-grained" with evolving bins "from large to small". In this way, we directly investigate how the search space is reduced by the multi-stage searching strategy. We compare the search space of supernet under different training stages, bin size β, and bin evolving speed α in Table 17. To fairly compare the results with different training stages, we set the size of the bin unit to be the same in Table 17. And the search space is defined as the product of the number of bins in all layers.

$$\mathcal{B}^{s}_{space} = \sum_{j=1}^{s} \prod_{i=1}^{L} \frac{Channel_i^j}{b_i^j} \tag{14}$$

Where $Channel_i^j$ represents the total channels at layer i in stage j, and b_i^j indicates the bin size of layer i in stage j. We examine the search space of 0.5$\times$ FLOPs MobileNetV2 on ImageNet dataset, as shown in Table 17.

Table 17: Search space of 0.5× FLOPs MobileNetV2 on ImageNet dataset.

0.5× MobileNetV2				
Stages	Bin size β	Evolve speed α	Search space	Percent
1	-	-	1.5×10^{27}	1
2	1	2	2.0×10^{19}	1.3×10^{-8}
2	2	4	4.8×10^{10}	3.0×10^{-17}
3	2	2	6.2×10^{10}	3.9×10^{-17}
4	4	2	7.3×10^{3}	4.6×10^{-24}

In detail, our search space reduced by more than $1/(1.3 \times 10^{-8})$ times from the original size (with one stage) by searching more than or equal to 2 stages. As a result, our proposed multi-stages search method can effectively reduce the search space.

A.16 Effect of smallest bin size β in FLOPs-sensitive bins

As formulated in Eq.(9), the total number of bins within each layer relies on the smallest bin size β. To examine the effect of β w.r.t. bin numbers and its corresponding performance, we perform experiments with MobileNetV2 on ImageNet dataset under 50% FLOPs budget, as shown in Table 18. In detail, the performance of searched network width first increases when β goes large but a little decreased after β surpasses 0.5, which may result from that a very small β induces a too huge search space thus is harmful to searching algorithms to find the optimized network width.

Table 18: Performance of ResNet50 and MobileNetV2 on ImageNet dataset w.r.t. different smallest bin size β for FLOPs-sensitive bins.

MobileNetV2			ResNet50		
Smallest bin size β	Top-1	Top-5	Smallest bin size β	Top-1	Top-5
4	72.22%	90.22%	4	76.43%	93.01%
2	72.32%	90.25%	2	76.73%	93.17%
1	72.40%	90.38%	1	76.91%	93.33%
0.5	72.34%	90.29%	0.5	76.82%	93.24%
0.25	72.28%	90.31%	0.25	76.67%	93.14%