Lightweight Alpha Matting Network Using Distillation-Based Channel Pruning

Donggeun Yoon[1][0000-0001-6153-8056], Jinsun Park[2][0000-0002-2296-819X], and
Donghyeon Cho*[1][0000-0002-2184-921X]

[1] Chungnam National University, Daejeon, Republic of Korea
202250187@o.cnu.ac.kr, cdh12242@cnu.ac.kr
[2] Pusan National University, Busan, Republic of Korea
jspark@pusan.ac.kr

Abstract. Recently, alpha matting has received a lot of attention because of its usefulness in mobile applications such as selfies. Therefore, there has been a demand for a lightweight alpha matting model due to the limited computational resources of commercial portable devices. To this end, we suggest a distillation-based channel pruning method for the alpha matting networks. In the pruning step, we remove channels of a student network having fewer impacts on mimicking the knowledge of a teacher network. Then, the pruned lightweight student network is trained by the same distillation loss. A lightweight alpha matting model from the proposed method outperforms existing lightweight methods. To show superiority of our algorithm, we provide various quantitative and qualitative experiments with in-depth analyses. Furthermore, we demonstrate the versatility of the proposed distillation-based channel pruning method by applying it to semantic segmentation.

Keywords: Matting · Channel Pruning · Knowledge Distillation.

1 Introduction

The purpose of a natural image matting (*i.e.*, alpha matting) is to estimate the transparency of the user-specified foreground in an image. The alpha matting is formally defined as follows [12]:

$$I = \alpha F + (1-\alpha)B, \tag{1}$$

where I, F and B are the observed color image, foreground, background, respectively. Also, α is transparency (*i.e.*, alpha matte). The natural image matting is a highly ill-posed problem because it needs to estimate F, B, and α

* Corresponding author.
Project page is at https://github.com/DongGeun-Yoon/DCP

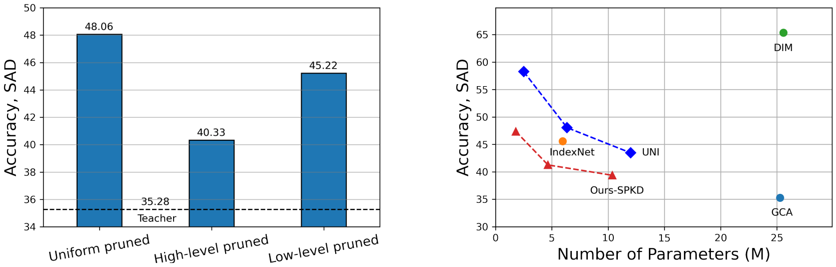

Fig. 1. (Left) Comparison of arbitrarily pruned models. The high-level pruned model removes channels from the high-level layers, and the low-level pruned model removes the channels from the low-level layers. (Right) Trade-offs between accuracy and model size. Circle dots show results of original models. Squares and triangles denote results of pruned GCA models using uniform pruning and our method, respectively.

simultaneously from an input color image I and a trimap providing known foreground and background pixels. Traditional approaches for natural image matting are categorized into affinity-based and sampling-based methods. Affinity-based methods [43, 28, 55, 20, 27, 4, 11, 5, 1] propagate alpha values from known regions to pixels in unknown regions by analyzing statistical correlation among pixels. Meanwhile, sampling-based methods [48, 16, 19, 41, 40, 7, 25, 8] construct foreground and background color sample sets using pixels in known areas, then estimates alpha values in unknown regions. However, these algorithms often rely on strong assumptions such as local smoothness [28] or sparsity of foreground and background colors [48].

Since the advent of large-scale image matting datasets such as Adobe-1k [50], deep learning-based matting algorithms been actively studied [10, 42, 9, 34, 35, 46, 2, 30, 37, 44]. These methods outperform conventional ones remarkably. Usually, the alpha matting networks are based on U-Net [38] or fully-convolutional networks (FCN) [33]. For better performance, the number of layers or channels can be increased and also auxiliary modules can be added to baseline networks. However, this leads to the increased computational costs and memory requirements that can be problematic for mobile applications. Recently, a lightweight alpha matting network based on similarity-preserving knowledge distillation (SPKD) [52] was introduced to resolve these issues. It successfully transfers similarities of features from the teacher network to the student network, which make the student network achieves much better performance than the baseline student network trained from scratch.

However, it is still an open problem that which architecture is the best one for the lightweight student network for natural image matting. It can be seen from the left of Figure 1 that the performance varies greatly depending on which layer the channels are removed from. Note that the high-level pruned model has fewer parameters than the low-level pruned model. Also, as shown in right of Figure 1, there is a trade-off between performance and model size, thus it is important to find an proper network architecture. To find the optimal lightweight network architecture, various network pruning techniques can be applied. Although it has been actively studied in the field of classification, it has not been dealt with much in the reconstruction problem including alpha matting. Recently, channel pruning methods for semantic segmentation task were introduced in [6, 21], but they mainly focus on preserving high-level semantic information rather than low-level fine structures that are crucial for the natural image matting problems.

To focus on low-level fine details during the channel pruning, we borrow the power of a pre-trained high-performance matting network which well preserves fine details. In other words, in this paper, we present a distillation-based channel pruning method that removes the channels having a low impact in mimicking the pre-trained teacher network. In the pruning phase, we induce the sparsity of the scaling factor of the batch normalization (BN) layer as in [31, 6, 21] and additionally apply distillation loss with a powerful pre-trained teacher model that is capable of precisely guiding a student network to preserve fine structural details in its prediction. In the training phase, we train the pruned lightweight network by the same distillation loss used in the pruning stage. Note that proposed method can make existing lightweight model (*i.e.*, IndexNet) even lighter.

Our contributions can be summarized as follows. (i) We introduce a novel channel pruning method for the natural image matting problem. To our best knowledge, this is the first attempt to apply the network pruning technique for the alpha matting problem. (ii) By utilizing a distillation loss within the channel pruning step, we succeed in finding a lightweight alpha matting network that can recover fine details. (iii) Our pruned network outperforms other baseline pruning approaches on two publicly available alpha matting datasets (Adobe-1K, Distinctions-646) while having a comparable number of parameters. In addition, we provide various ablation studies for a deeper understanding.

2 Related Works

2.1 Natural Image Matting

Most image matting techniques can be categorized into affinity-based and color sampling-based methods. In affinity-based methods, statistical affinity is analyzed among the local and non-local neighbors to propagate values of alpha to the unknown areas from the known regions. Levin *et al.* [28] introduced the closed-form solution based on matting Laplacian using the linear color model. For handling high resolution images, He *et al.* [20] proposed efficient method to solve a large kernel matting Laplacian. Furthermore, Lee and Wu [27] introduced non-local matting propagating alpha values across non-local neighboring pixels. Chen *et al.* [4] suggested the KNN matting which uses only k-nearest non-local neighbors to propagate alpha values. In addition, Chen *et al.* [5] utilized both local and non-local smoothness prior and Aksoy *et al.* [1] proposed multiple definitions of affinity for natural image matting.

The color sampling-based methods find foreground and background colors from constructed color sampler sets, then estimate alpha values in unknown regions. Bayesian matting [12] utilizes statistical models to analyze pixels in unknown regions. Robust matting [48], shared matting [16] and weighted color and texture matting [41] select the best color samples based on their own designed cost functions that take into account spatial, photometric, or texture information. He *et al.* [19] proposed a randomized searching method to use global samples in the known areas to find the best combination of foreground and background colors. Shahrian *et al.* [40] constructed comprehensive color sample sets to cover broad color variations using Gaussian Mixture Model (GMM). Karacan *et al.* [25] choose colors of foreground and background based on sparse representation.

After large-scale alpha matting datasets were published [50, 37], a lot of deep learning-based works have been introduced. Xu *et al.* [50] proposed a simple two-stage network for natural image matting. Lutz *et al.* [35] applied adversarial training for obtaining visually appealing alpha matte results. To preserve details of alpha mattes, Hao *et al.* [34] introduced IndexNet including indices-guided unpooling operation. In addition, contextual attention [30] and hierarchical attention [37] mechanisms were proposed for the matting problem. Yu *et al.* [53] proposed mask-guided matting leveraging a progressive refinement network with a general coarse mask as guidance. Although the performance of alpha matting has been substantially improved, there are still not many studies on lightening alpha matting networks. Recently, Yoon *et al.* [52] succeeded to utilize knowledge distillation (KD) to obtain the lightweight deep-CNN model for alpha matting. They reduce the number of channels with a fixed ratio, therefore, the optimal channel reduction ratio should be determined empirically.

2.2 Network Pruning

The purpose of the network pruning is to reduce redundancies in the overparameterized deep CNN models for fast run-time while maintaining performance. In general, network pruning is divided into unstructured pruning [18, 36, 14, 45] which requires special libraries or hardware, and structured pruning [49, 29, 3, 31] which is relatively easy to implement. In this subsection, we focus on structured pruning that is more relevant to our work. Wen *et al.* [49] proposed a Structured Sparsity Learning (SSL) method to sparsify structures including filters, channels, or layers by using group sparsity regularization. Li *et al.* [29] introduced a method to remove channels having small incoming weights in a trained deep CNN model. Changpinyo *et al.* [3] deactivate connections between filters in convolutional layers to obtain smaller networks. Liu *et al.* [31] proposed the network slimming method to explicitly impose channel-wise sparsity in the deep CNN model using scaling factors in batch normalization. Gao *et al.* [15] proposed a feature boosting and suppression (FBS) method to dynamically remove and boost channels according to the inputs using auxiliary connections. Despite many pruning studies, most of them focus on the classification task. Fortunately, pruning techniques for semantic segmentation have begun to be introduced recently. Chen *et al.* [6] suggested a channel pruning method for semantic segmentation based on multi-task learning. Furthermore, He *et al.* [21] proposed

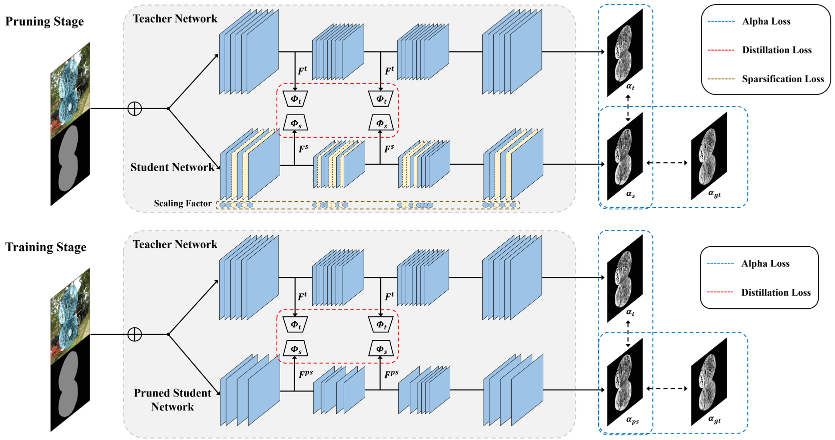

Fig. 2. In the pruning stage, the student network is lightweighted using scaling factor sparsification loss and distillation loss with pre-trained teacher model. In the training stage, the same distillation loss is used to train the pruned student network.

context-aware channel pruning method by leveraging the layer-wise channels interdependency. However, pruning researches for matting network have not been addressed yet.

Since the estimation of the fine structures in alpha mattes are the most important objective of the matting network, a powerful pruning technique suitable for this purpose is strongly required. To this end, we present a distillation-based channel pruning technique that exploits a powerful pre-trained alpha matting model suitable for recovering low-level fine details.

3 Proposed Approach

In this section, we briefly describe the basics of KD and motivation of using KD for network pruning. Then, we introduce the distillation-based channel pruning method for sparsifying alpha matting network, and explain a method for training pruned lightweight model with KD. We use the same distillation method for both pruning and training stages, even though it is also possible to utilize different methods. Related experiments are provided by the ablation studies. Overview of our distillation-based channel pruning and training is illustrated in Figure 2.

3.1 Knowledge Distillation

Knowledge distillation (KD) [23] is a technique supervising a small student model by a larger teacher model. The main purpose of KD is to transfer rich feature representations of the large model trained by the huge amount of data into the small model. Therefore, it is very useful when there is a lack of training data or limited computational resources and memory of the devices.

Table 1. The first row: model with channels removed from low-level layers. The second row: model with channels removed from high-level layers.

Methods	MSE	SAD	Grad	Conn	#Param
Low-level pruned	0.012	45.22	24.85	39.77	22.63M
High-level pruned	**0.011**	**40.33**	**20.34**	**35.48**	**8.83M**

Mathematically, feature maps of the teacher and student networks in i-th layer are denoted as $F_i^t \in \mathbb{R}^{C_i^t \times H_i^t \times W_i^t}$ and $F_i^s \in \mathbb{R}^{C_i^s \times H_i^s \times W_i^s}$, respectively. Note that $\{C_i^t, C_i^s\}$ are the number of channels, $\{H_i^t, H_i^s\}$ and $\{W_i^t, W_i^s\}$ represent the spatial size. Generally, the distillation loss for each layer is formulated as

$$\mathcal{L}_{KD}(F_i^t, F_i^s) = \mathcal{L}_F(\varPhi_t(F_i^t), \varPhi_s(F_i^s)), \tag{2}$$

where, $\mathcal{L}_F(\cdot)$ is a similarity function, $\varPhi_t(\cdot)$ and $\varPhi_s(\cdot)$ are feature transform functions for the teacher and student networks. According to the purpose of distillation, appropriate $\mathcal{L}_F(\cdot)$, $\varPhi_t(\cdot)$ and $\varPhi_s(\cdot)$ should be designed.

3.2 Motivation

Over the recent years, various KD methods have been introduced [23, 51, 47, 22, 17], but most of them arbitrarily set the architecture of the student network. Therefore, they do not ensure whether the student network is optimal for both distillation and the given tasks. For example, the importance of each channel in the layers of a deep CNN model may be different, therefore, reducing the number of channels uniformly for all the layers is sub-optimal obviously. We believe that it is also important for the alpha matting task to find the optimal student model.

To confirm this, we perform a preliminary experiment using GCA [30] as a baseline matting model. First, we divide the encoder of GCA model into two groups: low-level layers (`conv1-conv3`) and high-level layers (`conv4-conv5`). Then, we apply uniform 50% channel pruning to low-level and high-level layers separately, and then obtain two different pruned networks. The ratios of the removed channel parameters to the whole encoder are 12.75% in low-level and 37.25% in high-level layers, respectively. In other words, more parameters are eliminated from the high-level layers rather than the low-level ones. Using these two uniformly pruned GCA models, we verified the alpha matting performance on the Adobe-1k dataset [50]. As reported in Table 1, the model removing channels of high-level layers (the second row) prunes more channels than the model removing channels of low-level layers (the first row) but achieves better performance. The number of network parameters is also much less. As a result, it can be seen that the channels of low-level layers have more influence on the alpha matting performance. Regarding that the high-level pruned network has achieved better performance, this result implies that the channel distributions of the original and optimal pruned models might be different significantly and the low-level layers are highly important in the alpha matting problem.

Therefore, in this paper, we propose a method to find a student network model that can well receive low-level knowledge from a large-capacity teacher network. To this end, we introduce a distillation-aware pruning loss in the pruning stage to create an optimal lightweight network for alpha matting.

3.3 Pruning with KD

Inspired by [31], we adopt a channel pruning method based on the sparsification of scaling factors in batch normalization (BN) layers. BN layer is used in most deep CNN models for better generalization and fast convergence. Formally, the BN layer is defined as follows:

$$y = \gamma \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta, \tag{3}$$

where x and y are input and output of BN layer, and μ and σ are mean and standard deviation of the input mini-batch features, and ϵ is a small constant. γ and β are the learnable scaling and shifting factors. In [31], the scaling factor γ in the BN layer is considered as the measure for the importance of each channel. In other words, a channel with a very small γ is regarded as the layer which does not contribute significantly to the final prediction. Therefore, enforcing sparsification on the scaling factors eases the identification of prunable layers.

Similarly, our pruning method trains a target student network with sparsification loss and distillation loss, then remove channels with small scaling factors in BN layers. We adopt the same alpha matting model for both teacher and student networks. For the network pruning, only parameters of the student network are updated while those of the teacher network are fixed. The final loss includes alpha prediction, channel sparsification, and distillation losses as follows:

$$\mathcal{L}_P = \lambda_1 \mathcal{L}_\alpha(\alpha_{\mathrm{s}}, \alpha_{\mathrm{gt}}) + \lambda_2 \mathcal{L}_\alpha(\alpha_{\mathrm{s}}, \alpha_{\mathrm{t}}) + \lambda_3 \sum_{\gamma \in \zeta} |\gamma| + \lambda_4 \sum_{i \in \eta} \mathcal{L}_{KD}(F_i^t, F_i^s), \tag{4}$$

where $\mathcal{L}_\alpha(\cdot)$ is the vanilla alpha prediction loss introduced in [50], and α_{s}, α_{t}, α_{gt} are alpha matte prediction results from the student network and the teacher network, and ground truth, respectively. ζ is the set of scaling factors over all the BN layers and η is the index set of layers utilized for distillation loss. λ_1, λ_2, λ_3, and λ_4 are balancing factors for each term. Note that the distillation loss is used only for the encoder part.

In (4), the gamma value γ corresponding to the importance score can be estimated significantly differently depending on the distillation loss, which means that different pruned networks can be created. Therefore, we adopt several recent KD methods to be utilized in the proposed channel pruning method as follows:

- **Neuron Selectivity Transfer (NST)**: Huang and Wang proposed NST [24] that aligns the distribution of spatial neuron activations between teacher and student networks. To this end, NST minimizes maximum mean discrepancy (MMD) distance between activations of teacher and student networks. Thus,

$\Phi(\cdot)$ in (2) is a certain function for kernel trick which projects samples into a higher dimensional feature space. Also, $\mathcal{L}_F(\cdot)$ is distance (*i.e.*, L_2 distance) between means of projected features of teacher and student networks.
- **Overhaul of Feature Distillation (OFD)**: Heo *et al.* [22] investigated various aspects of the existing feature distillation methods and suggested OFD that is a simple but effective distillation method. In particular, $\Phi_t(\cdot)$ in (2) is a margin ReLU function while $\Phi_s(\cdot)$ is a regressor consisting of a 1×1 convolution layer. Also, $\mathcal{L}_F(\cdot)$ in (2) is a partial L_2 distance.
- **Similarity-Preserving Knowledge Distillation (SPKD)**: The SPKD-based distillation method makes the pairwise similarity of the student network similar to that of the teacher network. In [47], batch similarity for the classification task is used while spatial and channel similarity for the regression task is utilized in [52, 26]. Thus, $\Phi(\cdot)$ in (2) is a function of making pairwise similarities and $\mathcal{L}_F(\cdot)$ is the L_2 distance.

After training with distillation loss as in (4), we prune the target student network based on scaling factors of BN layers. The smaller the scaling factor is, the less impact it has on the output of the layer, thus we remove the channels with a lower scaling factor than a threshold. To eliminate M channels, we adopt the M-th smallest scaling factor as the threshold. At this point, thresholds of the encoder and decoder are obtained separately since distillation loss is only used in the encoder. After pruning, we can get the compact lightweight alpha matting network, which is suitable to get fine details by KD.

3.4 Training with KD

By the aforementioned our distillation-based channel pruning, the architecture of a lightweight student network can be obtained. In [32], the network structure itself is considered more important than the remaining parameters after pruning. In other words, the fine-tuned model and the trained from scratch model achieve similar result, or even the trained from scratch model performs better. Thus, we train the pruned network from scratch again by applying KD using the teacher network based on the loss function defined as follows:

$$\mathcal{L}_T = w_1\mathcal{L}_\alpha(\alpha_{\mathrm{ps}}, \alpha_{\mathrm{gt}}) + w_2\mathcal{L}_\alpha(\alpha_{\mathrm{ps}}, \alpha_{\mathrm{t}}) + w_3\sum_{i\in\eta}\mathcal{L}_{KD}(F_i^t, F_i^{ps}), \tag{5}$$

where α_{ps} is a prediction of the pruned student network and F_i^{ps} is feature maps in the i-th layer of the pruned student network. w_1, w_2, and w_3 are balancing factors for each term in (5). Unlike (4), sparsification loss is not included, and the pruned student network is used. We use the same distillation loss as the pruning step, but other distillation losses can be used.

4 Experimental Results

In this section, we evaluate the proposed distillation-based channel pruning method both quantitatively and qualitatively. We validate our method on vari-

Table 2. Quantitative evaluation by GCA-50% model on the benchmark.

Dataset	Methods		MSE	SAD	Grad	Conn	#Param	FLOPs
	KD	Prune						
Adobe-1k	Unpruned(teacher)		0.009	35.28	16.92	35.53	25.27M	11.19G
	-	UNI	0.017	52.61	34.27	46.24	6.35M	**2.90G**
	-	NS	0.017	51.57	28.70	45.66	**4.30M**	6.84G
	-	FBS	0.013	45.09	22.87	39.39	7.11M	**2.90G**
	-	CAP	0.014	48.55	26.01	42.76	5.13M	6.50G
	NST	NS	0.020	58.10	36.78	51.80	**4.30M**	6.84G
	NST	Ours	0.017	53.47	30.59	47.37	5.80M	7.02G
	OFD	NS	0.014	46.96	24.16	41.93	**4.30M**	6.84G
	OFD	Ours	0.012	43.15	21.79	37.66	5.13M	6.48G
	SPKD	NS	0.012	42.69	21.88	37.54	**4.30M**	6.84G
	SPKD	Ours	**0.011**	**41.26**	**21.42**	**35.87**	4.66M	6.74G
Disticntions-646	Unpruned(teacher)		0.025	27.60	15.82	22.03	25.27M	11.19G
	-	UNI	0.037	37.83	26.35	28.36	6.35M	**2.90G**
	-	NS	0.028	31.45	20.47	25.69	**3.99M**	6.41G
	-	FBS	**0.024**	28.61	**17.36**	23.52	7.11M	**2.90G**
	-	CAP	0.026	29.23	18.59	23.85	5.30M	6.36G
	NST	NS	0.038	36.13	24.33	28.20	**3.99M**	6.41G
	NST	Ours	0.035	36.58	23.69	27.42	4.93M	6.16G
	OFD	NS	0.026	28.45	19.86	23.22	**3.99M**	6.41G
	OFD	Ours	**0.024**	**27.12**	**17.36**	22.03	5.03M	6.22G
	SPKD	NS	**0.024**	27.88	18.33	22.46	**3.99M**	6.41G
	SPKD	Ours	**0.024**	27.31	17.81	**21.90**	4.24M	6.72G

ous teacher models including GCA [30], DIM [50], and IndexNet [34], and also provide various ablation studies. Finally, we show that the proposed algorithm can be utilized for the other task such as semantic segmentation.

4.1 Implementation Details

In most experiments, we adopt GCA matting as a baseline alpha matting network. In order to evaluate our distillation-based channel pruning method, we use two public benchmark datasets: Adobe-1k [50] and Distinctions-646 [37]. Since Distinctions-646 test set does not provide official trimaps, we generate trimaps from ground truth alpha matte using dilation with kernel size 10. The evaluation metrics for all quantitative experiments are mean squared error (MSE), sum of absolute difference (SAD), gradient error (Grad), connectivity (Conn), the number of network parameters (#Param) and floating-point-opertions (FLOPs). We use activations of the last four layers in the encoder for computing distillation loss as in [52] for a fair comparison.

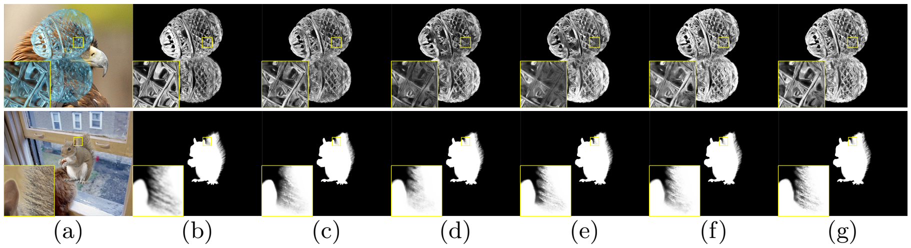

Fig. 3. Qualitative image matting results by GCA. (a) Input images. (b) Ground truths. (c) UNI. (d) NS [31]. (e) CAP [21]. (f) NS-SPKD. (g) Ours-SPKD.

Table 3. Quantitative results by DIM-50% model on the Adobe-1k.

Methods KD	 Prune	MSE	SAD	Grad	Conn	#Param	FLOPs
Unpruned(teacher)		0.021	65.37	33.20	67.58	25.58M	24.51G
-	UNI	0.049	114.02	78.89	122.00	6.40M	6.19G
-	NS	0.052	120.82	83.10	129.70	6.32M	7.65G
-	FBS	0.052	110.22	83.14	118.60	6.92M	6.19G
-	CAP	0.040	101.77	63.93	108.61	**4.08M**	**4.26G**
NST	NS	0.033	94.31	55.48	100.67	6.32M	7.65G
NST	Ours	0.032	89.83	49.04	95.20	5.98M	11.23G
OFD	NS	0.029	76.74	42.45	79.86	6.32M	7.65G
OFD	Ours	0.032	76.71	43.86	80.61	7.97M	17.61G
SPKD	NS	0.029	76.73	42.70	80.77	6.32M	7.65G
SPKD	Ours	**0.027**	**73.67**	**40.78**	**76.58**	7.81M	12.53G

4.2 Quantitative Comparisons

We quantitatively verify our distillation-based channel pruning and training methods on Adobe-1k and Distinctions-646 datasets. We adopt the aforementioned NST [24], OFD [22], and SPKD [52] as KD methods for both pruning and training stages. For comparison, uniform channel pruning (UNI), network slimming (NS) [31], feature boosting and suppression (FBS) [15], and context-aware pruning (CAP) [21] are chosen. As reported in Table 2, the number of parameters in all student networks and FLOPs are much smaller than that of a teacher network (about 16-25% parameters and 60% FLOPs). Although our pruned model sometimes has more parameters or more FLOPs than other pruned models (UNI, NS, FBS and CAP), the alpha matting performance is far superior to their performances. Also, our distillation-based channel pruning method achieves better performance than NS regardless of distillation types. Note that we utilize the same KD method in the training step for both our pruning method and NS. Usually, performance is slightly higher when SPKD is used than OFD.

Table 4. Quantitative results by IndexNet-25% model on the Adobe-1k.

Methods		MSE	SAD	Grad	Conn	#Param	FLOPs
KD	Prune						
Unpruned(teacher)		0.013	45.61	27.06	43.79	8.15M	5.64G
-	UNI	0.027	65.21	42.56	66.83	3.49M	**3.22G**
-	NS	0.026	65.45	40.79	67.11	4.21M	5.34G
-	FBS	0.020	56.43	34.41	56.78	4.05M	**3.22G**
-	CAP	0.018	53.32	35.17	53.41	4.07M	5.38G
NST	NS	0.020	57.04	33.99	57.49	4.21M	5.34G
NST	Ours	0.019	54.21	28.14	53.57	**4.02M**	4.95G
OFD	NS	0.017	52.13	29.99	51.76	4.21M	5.34G
OFD	Ours	0.015	47.24	**24.42**	46.26	4.41M	5.13G
SPKD	NS	0.016	50.35	27.21	49.86	4.21M	5.34G
SPKD	Ours	**0.014**	**47.06**	25.98	**45.77**	5.09M	5.21G

However, when NST is used, the performance is lower than the existing pruning that does not include KD in the training step. It indicates that the type of distillation loss is also an important factor for both the pruning and training.

To verify the generality of our method, the same experiments are performed using DIM and IndexNet as backbone models instead of GCA matting. Similar to the case of GCA matting, the best performance is achieved with SPKD as reported in Table 3 and Table 4. A different point from the case of GCA matting is that comparable performance was achieved even when using NST. Note that the original IndexNet is already a lightweight model because it is based on MobileNetv2 [39], but it can be even lighter by applying our channel pruning.

4.3 Qualitative Comparisons

Figure 3 shows the qualitative performance of our method. We compare our results obtained using SPKD with results from existing pruning algorithms. Examples contain various object structures: short hair, overlapped color distribution (*squirrel*) and transparency (*glass*). As expected, the results of the existing pruning methods are over-smoothed as shown in the (*glass*) example of Figure 3-(d,e). In this example, UNI produces a better result than NS and CAP. Overall, results using distillation loss in the pruning step (Figure 3-(f)) show stable and visually pleasing predictions. Moreover, our final results using the SPKD in both pruning and training step (Figure 3-(g)) provide the best predictions with fine details preserved.

4.4 Ablation Studies

Different Distillation for Pruning and Training. Since it is possible to utilize different distillation losses for pruning and training stages, it is meaningful

Table 5. Ablation study on various combinations of methods for distillation and pruning method. All evaluations are conducted by GCA-50% model on the Adobe-1k.

Methods KD	 Prune	MSE	SAD	Grad	Conn
NST	+NST	0.017	53.47	30.59	47.37
	+OFD	0.018	53.70	30.85	46.78
	+SPKD	0.019	56.85	35.94	49.30
OFD	+NST	0.012	44.09	21.91	39.24
	+OFD	0.012	43.15	21.79	37.66
	+SPKD	0.013	44.14	23.51	39.82
SPKD	+NST	0.013	43.98	23.65	38.82
	+OFD	**0.011**	42.07	21.51	36.58
	+SPKD	**0.011**	**41.26**	**21.42**	**35.87**

Table 6. Results according to various pruning ratios. All evaluations are conducted by GCA-model with SPKD on the Adobe-1k.

Methods	MSE	SAD	Grad	Conn	#Param
UNI-30%	0.012	43.46	23.78	38.61	12.00M
UNI-50%	0.015	48.06	30.28	42.81	6.35M
UNI-70%	0.019	58.26	33.53	50.77	2.50M
Ours-30%	**0.010**	**39.38**	**19.45**	**34.80**	10.37M
Ours-50%	0.011	41.26	21.42	35.87	4.66M
Ours-70%	0.014	47.36	25.87	42.02	**1.80M**

to explore whether it is better to use different distillation losses in pruning and training steps or to use the same distillation loss. To this end, we performed experiments on all combinations of NST, OFD, and SPKD in the pruning and training phases. As reported in Table 5, we can achieve better performance when the same distillation loss is used in the pruning and training phases. Even more, in the student model pruned with NST, it is better to use NST in the training stage than OFD and SPKD, which are more advanced distillation techniques. These results are reasonable because the student network architecture obtained by a specific KD method will have a high chance to be more effective for the same distillation method than the other ones.

Pruning Ratio. We analyze our distillation-based pruning according to pruning ratios. We compare the results of the model in which the number of channels is reduced by 30%, 50%, and 70% using the our method, and the model uniformly reduced in the same proportion. For all cases, we use the same SPKD as distillation loss for training step. As in Table 6, the pruned models whose channels are reduced by 70% and 50% using our method achieve better performance and fewer network parameters than the model uniformly pruned by 50%.

Table 7. Results of training our pruned model from scratch without distillation.

Prune	MSE	SAD	Grad	Conn
+NST	**0.017**	**54.02**	**31.50**	**47.45**
+OFD	**0.017**	56.44	33.10	49.39
+SPKD	**0.017**	56.88	32.62	50.09

Table 8. Comparisons of Running Time (RT) per image on DIM model.

Method	MSE	SAD	#Param	FLOPs	RT(ms)
Unpruned	0.021	65.37	25.58M	24.51G	11.33
UNI	0.049	114.02	6.40M	6.19G	3.80
NS	0.052	110.22	6.32M	7.65G	3.87
CAP	0.040	101.77	**4.08M**	**4.26G**	**3.50**
Ours(NST)	0.032	89.83	5.98M	11.23G	5.53
Ours(OFD)	0.032	76.71	7.97M	17.61G	5.67
Ours(SPKD)	**0.027**	**73.67**	7.81M	12.53G	5.64

Training from Scratch without KD. To analyze the effect of paired distillation loss for both the pruning and training stages, we train our pruned model from scratch without KD. As reported in Table 7, our pruned model achieves slightly worse performance than models pruned by UNI, NS, and CAP when trained without KD in the training phase. Therefore, we conclude that our distillation-based channel pruning is more beneficial when it is combined with the proper distillation method during the training.

Running Time We measure the running time of the each pruned model using Adobe-1k dataset. As reported in Table 8, the performance (MSE, SAD) of our method with SPKD is quite close to those of unpruned teacher model while it runs twice faster than the teacher model. The existing methods (UNI, NS, CAP) are faster than our method, but the performance (MSE, SAD) is very poor.

Application on Semantic Segmentation. Our distillation-based channel pruning technique is applicable not only to alpha matting but also to other tasks. Therefore, in this subsection, we verify whether the proposed method is effective for semantic segmentation. For experiments, we adopt PSPNet-50 [54] as a baseline model and test our method on Pascal VOC 2012 validation set [13]. We utilize the mean Intersection over Union (mIoU), and pixel accuracy (Acc.) as evaluation metrics.

As reported in Table 9, the proposed distillation-based channel pruning method achieves superior performance compared to the other existing channel pruning methods. Note that the performance of the pruned model by the NS is similar to our method when KD is applied, but the pruned model by our method has much fewer parameters. Also, as shown in Figure 4, our channel pruning method produces a visually more pleasing result compared to the other channel pruning techniques.

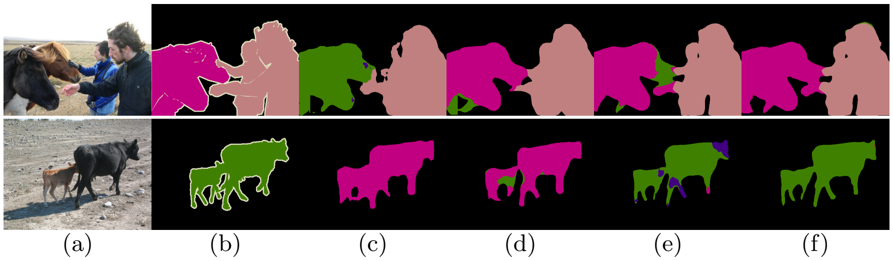

Fig. 4. Example of semantic segmentation results on PASCAL VOC2012 validation set. (a) Input images. (b) Ground truths. (c) NS. (d) CAP. (e) NS-OFD. (f) Ours-OFD.

Table 9. Semantic segmentation results by 50% pruned PSPNet-50.

Methods		mIoU	Acc.	#Param	Methods		mIoU	Acc.	#Param
KD	Prune				KD	Prune			
Unpruned (teacher)		78.02	95.13	49.1M	NST	NS	60.67	90.80	26.5M
					NST	Ours	60.81	90.69	22.0M
-	UNI	51.40	88.05	**12.3M**	OFD	NS	62.45	90.69	26.5M
-	NS	41.78	85.24	26.5M	OFD	Ours	**63.48**	**90.85**	21.7M
-	FBS	53.62	88.75	19.7M	SPKD	NS	58.26	89.55	26.5M
-	CAP	53.61	88.58	22.2M	SPKD	Ours	58.20	89.18	21.9M

5 Conclusion

We have proposed a distillation-based channel pruning method for lightening a deep image matting network. In the pruning step, we train a student network that has the same architecture with a teacher network using the distillation-based sparsification loss. Then, we remove channels that have low scaling factor of BN layer. Finally, we train the pruned student network using the same distillation loss utilized in the pruning step. Experimental results demonstrate that our distillation-based channel pruning method successfully reduces the number of parameters. The lightweight network obtained by the proposed method achieves significantly better performance than other lightweight networks with similar capacity. We analyze the proposed channel pruning technique through extensive ablation studies.

Acknowledgements This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2022-00155857, Artificial Intelligence Convergence Innovation Human Resources Development (Chungnam National University)) and the National Research Foundation of Korea (NRF) grant funded by the Korea government. (MSIT, No.2021R1A4A1032580 and No.2022R1C1C1009334).

References

1. Aksoy, Y., Aydin, T.O., Pollefeys, M.: Designing effective inter-pixel information flow for natural image matting. In: Proc. of Computer Vision and Pattern Recognition (CVPR) (2017)
2. Cai, S., Zhang, X., Fan, H., Huang, H., Liu, J., Liu, J., Liu, J., Wang, J., Sun, J.: Disentangled image matting. In: Proc. of Int'l Conf. on Computer Vision (ICCV) (2019)
3. Changpinyo, S., Sandler, M., Zhmoginov, A.: The power of sparsity in convolutional neural networks. ArXiv **abs/1702.06257** (2017)
4. Chen, Q., Li, D., Tang, C.K.: Knn matting. In: Proc. of Computer Vision and Pattern Recognition (CVPR) (2012)
5. Chen, X., Zou, D., Zhou, S.Z., Zhao, Q., Tan, P.: Image matting with local and nonlocal smooth priors. In: Proc. of Computer Vision and Pattern Recognition (CVPR) (2013)
6. Chen, X., Wang, Y., Zhang, Y., Du, P., Xu, C., Xu, C.: Multi-task pruning for semantic segmentation networks. ArXiv **abs/2007.08386** (2020)
7. Cho, D., Kim, S., Tai, Y.W.: Consistent matting for light field images. In: Proc. of European Conf. on Computer Vision (ECCV). pp. 90–104. Springer (2014)
8. Cho, D., Kim, S., Tai, Y.W., Kweon, I.S.: Automatic trimap generation and consistent matting for light-field images. IEEE Trans. on Pattern Anal. Mach. Intell. (TPAMI) **39**(8), 1504–1517 (2016)
9. Cho, D., Tai, Y.W., Kweon, I.S.: Deep convolutional neural network for natural image matting using initial alpha mattes. IEEE Trans. on Image Processing (TIP) **28**(3), 1054–1067 (2018)
10. Cho, D., Tai, Y.W., Kweon, I.: Natural image matting using deep convolutional neural networks. In: Proc. of European Conf. on Computer Vision (ECCV) (2016)
11. Choi, I., Lee, M., Tai, Y.W.: Video matting using multi-frame nonlocal matting laplacian. In: Proc. of European Conf. on Computer Vision (ECCV) (2012)
12. Chuang, Y.Y., Curless, B., Salesin, D.H., Szeliski, R.: A bayesian approach to digital matting. In: Proc. of Computer Vision and Pattern Recognition (CVPR) (2001)
13. Everingham, M., Eslami, S.M.A., Gool, L.V., Williams, C.K.I., Winn, J.M., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision **111**, 98–136 (2014)
14. Frankle, J., Carbin, M.: The lottery ticket hypothesis: Finding sparse, trainable neural networks. In: International Conference on Learning Representations (2019)
15. Gao, X., Zhao, Y., Łukasz Dudziak, Mullins, R., zhong Xu, C.: Dynamic channel pruning: Feature boosting and suppression. In: International Conference on Learning Representations (2019)
16. Gastal, E.S.L., Oliveira, M.M.: Shared sampling for real-time alpha matting. In: Eurographics (2010)
17. Ge, S., Zhao, S., Li, C., Li, J.: Low-resolution face recognition in the wild via selective knowledge distillation. IEEE Trans. on Image Processing (TIP) **28**(4), 2051–2062 (2019)
18. Han, S., Pool, J., Tran, J., Dally, W.J.: Learning both weights and connections for efficient neural networks. In: Proc. of Neural Information Processing Systems (NeurIPS). p. 1135–1143 (2015)
19. He, K., Rhemann, C., Rother, C., Tang, X., Sun, J.: A global sampling method for alpha matting. In: Proc. of Computer Vision and Pattern Recognition (CVPR) (2011)

20. He, K., Sun, J., Tang, X.: Fast matting using large kernel matting laplacian matrices. In: Proc. of Computer Vision and Pattern Recognition (CVPR) (2010)
21. He, W., Wu, M., Liang, M., Lam, S.K.: Cap: Context-aware pruning for semantic segmentation. In: Proc. of Winter Conference on Applications of Computer Vision (WACV). pp. 960–969 (2021)
22. Heo, B., Kim, J., Yun, S., Park, H., Kwak, N., Choi, J.Y.: A comprehensive overhaul of feature distillation. In: Proc. of Int'l Conf. on Computer Vision (ICCV) (2019)
23. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: Proc. of Neural Information Processing Systems Workshops (NeurIPSW) (2015)
24. Huang, Z., Wang, N.: Like what you like: Knowledge distill via neuron selectivity transfer. ArXiv **abs/1707.01219** (2017)
25. Karacan, L., Erdem, A., Erdem, E.: Image matting with kl-divergence based sparse sampling. In: Proc. of Int'l Conf. on Computer Vision (ICCV) (2015)
26. Ko, S., Park, J., Chae, B., Cho, D.: Learning lightweight low-light enhancement network using pseudo well-exposed images. IEEE Signal Process. Lett. (SPL) **29**, 289–293 (2022)
27. Lee, P., Wu, Y.: Nonlocal matting. In: Proc. of Computer Vision and Pattern Recognition (CVPR) (2011)
28. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE Trans. on Pattern Anal. Mach. Intell. (TPAMI) **30**(2), 0162–8828 (2008)
29. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. In: International Conference on Learning Representation (ICLR) (2017)
30. Li, Y., Lu, H.: Natural image matting via guided contextual attention. In: Association for the Advancement of Artificial Intelligence (AAAI) (2020)
31. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning efficient convolutional networks through network slimming. In: Proc. of Int'l Conf. on Computer Vision (ICCV). pp. 2736–2744 (2017)
32. Liu, Z., Sun, M., Zhou, T., Huang, G., Darrell, T.: Rethinking the value of network pruning. In: ICLR (2019)
33. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 3431–3440 (2015)
34. Lu, H., Dai, Y., Shen, C., Xu, S.: Indices matter: Learning to index for deep image matting. In: Proc. of Int'l Conf. on Computer Vision (ICCV) (2019)
35. Lutz, S., Amplianitis, K., Smolic, A.: Alphagan: Generative adversarial networks for natural image matting. In: British Machine Vision Conf. (BMVC) (2018)
36. Molchanov, D., Ashukha, A., Vetrov, D.: Variational dropout sparsifies deep neural networks. In: Proc. of Int'l Conf. on Machine Learning (ICML). p. 2498–2507 (2017)
37. Qiao, Y., Liu, Y., Yang, X., Zhou, D., Xu, M., Zhang, Q., Wei, X.: Attention-guided hierarchical structure aggregation for image matting. In: Proc. of Computer Vision and Pattern Recognition (CVPR) (2020)
38. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241 (2015)
39. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
40. Shahrian, E., Rajan, D., Price, B., Cohen, S.: Improving image matting using comprehensive sampling sets. In: Proc. of Computer Vision and Pattern Recognition (CVPR) (2013)

41. Shahrian, E., Rajan, D.: Weighted color and texture sample selection for image matting. In: Proc. of Computer Vision and Pattern Recognition (CVPR) (2012)
42. Shen, X., Tao, X., Gao, H., Zhou, C., Jia, J.: Deep automatic portrait matting. In: Proc. of European Conf. on Computer Vision (ECCV) (2016)
43. Sun, J., Jia, J., Tang, C.K., Shum, H.Y.: Poisson matting. ACM Trans. on Graph. (ToG) **23**(3), 315–321 (2004)
44. Sun, Y., Tang, C.K., Tai, Y.W.: Semantic image matting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11120–11129 (June 2021)
45. Tanaka, H., Kunin, D., Yamins, D.L., Ganguli, S.: Pruning neural networks without any data by iteratively conserving synaptic flow. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 6377–6389. Curran Associates, Inc. (2020)
46. Tang, J., Aksoy, Y., Oztireli, C., Gross, M., Aydin, T.O.: Learning-based sampling for natural image matting. In: Proc. of Computer Vision and Pattern Recognition (CVPR) (2019)
47. Tung, F., Mori, G.: Similarity-preserving knowledge distillation. In: Proc. of Int'l Conf. on Computer Vision (ICCV) (2019)
48. Wang, J., Cohen, M.F.: Optimized color sampling for robust matting. In: Proc. of Computer Vision and Pattern Recognition (CVPR) (2007)
49. Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H.: Learning structured sparsity in deep neural networks. In: Proc. of Neural Information Processing Systems (NeurIPS) (2016)
50. Xu, N., Price, B.L., Cohen, S., Huang, T.S.: Deep image matting. In: Proc. of Computer Vision and Pattern Recognition (CVPR) (2017)
51. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: Proc. of Computer Vision and Pattern Recognition (CVPR) (2017)
52. Yoon, D., Park, J., Cho, D.: Lightweight deep cnn for natural image matting via similarity-preserving knowledge distillation. IEEE Signal Processing Letters **27**, 2139–2143 (2020)
53. Yu, Q., Zhang, J., Zhang, H., Wang, Y., Lin, Z., Xu, N., Bai, Y., Yuille, A.: Mask guided matting via progressive refinement network. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 1154–1163 (2021)
54. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
55. Zheng, Y., Kambhamettu, C.: Learning based digital matting. In: Proc. of Int'l Conf. on Computer Vision (ICCV) (2009)