CAP: Context-Aware Pruning for Semantic Segmentation

Wei He Meiqing Wu Mingfu Liang Siew-Kei Lam

School of Computer Science and Engineering, Nanyang Technological University,
50 Nanyang Ave, Singapore

{wei005, n18061811}@e.ntu.edu.sg {meiqingwu, assklam}@ntu.edu.sg

Abstract

Network pruning for deep convolutional neural networks (CNNs) has recently achieved notable research progress on image-level classification. However, most existing pruning methods are not catered to or evaluated on semantic segmentation networks. In this paper, we advocate the importance of contextual information during channel pruning for semantic segmentation networks by presenting a novel Context-aware Pruning framework. Concretely, we formulate the embedded contextual information by leveraging the layer-wise channels interdependency via the Context-aware Guiding Module (CAGM) and introduce the Context-aware Guided Sparsification (CAGS) to adaptively identify the informative channels on the cumbersome model by inducing channel-wise sparsity on the scaling factors in batch normalization (BN) layers. The resulting pruned models require significantly lesser operations for inference while maintaining comparable performance to (at times outperforming) the original models. We evaluated our framework on widely-used benchmarks and showed its effectiveness on both large and lightweight models. On Cityscapes dataset, our framework reduces the number of parameters by 32%, 47%, 54%, and 63%, on PSPNet101, PSPNet50, ICNet, and SegNet, respectively, while preserving the performance.

1. Introduction

Semantic segmentation, which predicts a semantic label for all pixels in the given image, plays a vital role in applications such as autonomous driving, robot navigation, *etc*. It has achieved significant progress in recent years due to the advancement of CNNs [1, 6, 12, 22, 37]. Nonetheless, like image-level classification, the state-of-the-art models [5, 37] for semantic segmentation also have a large number of parameters and require high computational cost for such dense prediction. This hinders their deployment on mobile or embedded devices with limited resources and a strict requirement on inference latency. A large body of research has been dedicated to overcome these deployment challenges by reducing model size and floating-point operations. One direction is to design efficient lightweight models [36, 28] directly. Another orthogonal research area is to accelerate the models by removing the parameter redundancies via network pruning [11, 19, 20, 16]. However, most existing pruning methods are mainly evaluated for image-level classification networks, and their generalization on semantic segmentation networks are seldom discussed.

Lately, for improving accuracy, a few works have begun to exploit the contextual information in semantic segmentation [8, 35, 34, 38], where they mostly attempt to integrate the global context hints with either attention [31] or non-local mechanisms [32]. These works motivated our hypothesis that since contextual information is crucial for performance, they can also serve as an essential cue to guide pruning. Particularly, when pruning segmentation networks, emphasis must be given to preserving contextual informative features, whilst channels that exhibit lesser important contextual properties can be discarded. Our approach departs from all existing network pruning methods that are agnostic to the contextual information in the intermediate features.

In this paper, we propose a novel network compression framework, called *Context-aware Pruning*. The framework is based on the crucial insight of semantic parsing, wherein the determination of pixel semantics requires abundant aggregation of local abstract features with its surrounding information. A *Context-aware Guiding Module* (CAGM) is proposed to quantify the contextual information among channels into a guiding vector. Next, to distinguish the critical channels in the original model, a *Context-aware Guided Sparsification* (CAGS) approach is introduced to sparsify the channel-wise scaling factors in the batch normalization (BN) layers [18] under the guidance. By enforcing the scaling factors to zero, the corresponding channels can be re-

garded as redundant since their output will be scaled to zero, and hence, these filters can be potentially removed. Since the BN layer is generally employed in most networks, our framework can be easily applied. Moreover, for CNNs with no normalization layers, simple pseudo scaling factors [16] can be introduced. Our main contributions are as follows:

- To the best of our knowledge, this is the first work to explore contextual information for guiding channel pruning tailored to semantic segmentation. In the proposed framework, CAGM quantifies the contextual information using the channel's association. CAGS is introduced to induce channel-wise sparsity under the formulated contextual guidance, where the structured penalty is emphasized or de-emphasized accordingly. Besides, the proposed optimization is able to reveal the informative channels adaptively from different inputs.
- Our work is the first to expose large opportunities for pruning both large and lightweight semantic segmentation models, where a good generalization is demonstrated via quantitative results. It not only effectively removes redundancies in large networks like PSPNet, but can also prune lightweight networks like ICNet.
- On various benchmarks (*e.g.*, CamVid, Cityscapes), extensive experimental results show the advantages of our framework, which can effectively generate compact models for various state-of-the-art segmentation networks with significantly fewer parameters and floating-point operations, while maintaining better performance to (at times outperforming) the original models, comparing to all other baseline methods.

2. Related Works

Semantic Segmentation. Since the advent of FCN [22], various architectures have been proposed for semantic segmentation. Among the encoder-decoder style, Unet [29] and SegNet [1] share encoder's features or information via shortcut connection and the pooling indices, respectively. Dilated-convolution-based models enlarge the receptive field via dilated/atrous convolution operations to extract and aggregate larger sub-region context. To gather multi-scale contextual information from high-level features, Deeplab series models [4, 5] use the atrous spatial pyramid pooling (ASPP), and PSPNet [37] introduces a pyramid pooling module (PPM). However, PSPNet101 has 70.43 million parameters (approximately 282MB memory storage) and requires 574.9 Giga FLOPs per image per inference in the Cityscapes dataset, and its inference speed on a TitanX GPU is only 0.78fps [36]. This violates the real-time requirement of many real-world applications.

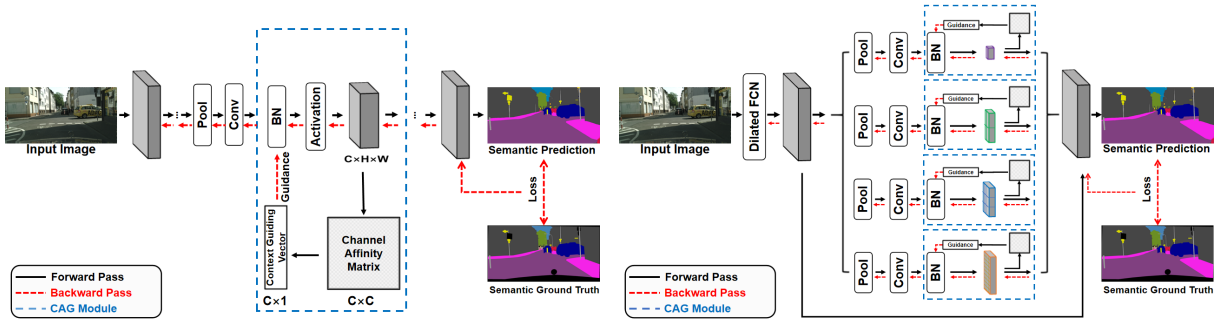
To cater to the real-time constraints of low-power devices, attempts have been made to directly design lesser accurate but lightweight segmentation models. ENet [28] gains notable acceleration at the cost of significant accuracy drop, while [36] presented ICNet, balancing the performance. However, designing an efficient and compact segmentation structure is time-consuming and requires many trial-and-error attempts. Moreover, as shown in Section 4, lightweight architectures such as ICNet still possess massive redundancies that can be discarded by our method.

Network Pruning. There is a general consensus that over-parameterization of most CNNs are necessary for training but not for inference. Network pruning aims to remove redundancies in the over-parameterized models for faster inference while maintaining most model capacity. Among them, the non-structured pruning methods [11, 25] require specialized library or hardware support, while structured pruning [33, 27, 23, 15] approaches focus on pruning the entire structure for easier implementation (*e.g.*, kernel, filter, and even layer). SSL [33] sparsifies structures with Group Lasso, while other sparsification-based methods identify the redundancies on multi-level structures by imposing sparsity [16, 20, 10]. SFP [13] keeps updating the pruned filters by setting their weights to zero instead of removing. Different pruning criteria are regarded as the indicators for redundant filters and have been studied [17, 26, 24], *e.g.*, filter L1-norm [19] and filter Geometric Median [14]. Unlike the traditional compression, more recent conditional computing based methods [2, 9] selectively switch the channels on/off based on the runtime activation. However, such methods still need to deploy the entire complex models to maintain the representation capacity for selections, and hence they are not suitable for systems with tight memory constraints. Also, the effectiveness of the above-mentioned pruning approaches on the more complex semantic segmentation models is seldom discussed.

Currently, a research gap exists for pruning methods that are catered to semantic segmentation networks. As shown in our experiments, when existing pruning methods on classification networks (*i.e.*, [20] and [14]) are applied to semantic segmentation networks, the pruned model suffers from substantial performance degradation. On the other hand, addressing the critical role of contextual information in semantic segmentation, our work utilizes the contextual priors to guide the pruning of unimportant channels, and the empirical results show that our framework leads to compact models that outperform the above-mentioned model compression approaches and also the state-of-the-art conditional computing based acceleration method.

3. Proposed Method

In this section, we first present the intuition behind our *Context-aware Pruning* framework for compressing segmentation networks, before describing the proposed *Context-aware Guiding Module* (CAGM) and accompany-



(a) Basic CAGM (b) CAGMs in Pyramid Pooling Module

Figure 1. Given an input feature map $\mathbf{M}_i \in \mathbb{R}^{C_i \times H_i \times W_i}$ in layer i, CAGM first calculates the *Channel Affinity Matrix* $\mathbf{A}_i \in \mathbb{R}^{C_i \times C_i}$. Then, the squeezing operation is applied to compute the *Context Guiding Vector* $\mathbf{V}_i \in \mathbb{R}^{C_i \times 1}$, which carry the adaptive penalty strength for the corresponding scaling factors in the BN layer during CAGS. Note that we only use the CAGM during backward pass to penalize the scaling factor. After sparsification, CAGM can be removed without affecting the output. (a) shows the basic CAGM in a certain layer, and (b) illustrates how the CAGMs are positioned within the Pyramid Pooling Module.

ing *Context-aware Guided Sparsification* (CAGS).

Unlike image classification, semantic segmentation emphasizes more on local-to-global features aggregation [37, 8, 35]. Our method exploits the property wherein spatial semantic contextual information can be captured via multi-scale pooling or downsampling [37, 35, 4], which are further fused via various strategies to facilitate pixel-wise prediction. However, from the high-level semantic features that embed such contextual information, the associations or combinations of different channel maps may also contribute differently towards useful contextual information. For example, in a well-trained network, a particular association of channels activation in the feature maps may represent specific useful context with semantic meaning (*e.g.*, driving lane) for a semantic class (*e.g.*, car), while a different channels association may provide another contextual hint. Thus, to condense such knowledge, channels that always provide useful contextual clues under diverse inputs should be preserved, while others can be considered for removal.

Most existing pruning methods ignore such channels association on features, and thus the pruned semantic segmentation model performs poorly when channels with influential contextual information are removed. However, it is not straightforward to directly measure channel contextual importance, since different contextual information may come from different channels and they may not be independent to each other. Our framework leverages the channel maps affinity for guiding structured sparsification to discover contextual informative channels and exploit the contextual redundancy for pruning semantic segmentation networks.

3.1. Context-Aware Guiding Module (CAGM)

As shown in Figure 1, to quantify the channel's association on high-level feature maps that constitutes contextual information, we propose the *Context-aware Guiding Module* (CAGM), which is applied to the feature maps adjacent to a pooling layer on top of the original network. CAGM measures the integrated channels interdependency into vector layer-wise, namely *Context Guiding Vector*, which is used to direct the contextual informative channels selection in *Context-aware Guiding Sparsification* (CAGS) (discussed in Section 3.2).

Mathematically, we denote the feature maps in layer i as $\mathbf{M}^i \in \mathbb{R}^{C_i \times H_i \times W_i}$, where C_i is the number of channels, H_i and W_i represent the spatial size. CAGM first calculates the symmetrical *Channel Affinity Matrix* $\mathbf{A}^i \in \mathbb{R}^{C_i \times C_i}$ from $\mathbf{M}^i$.

The feature maps $\mathbf{M}^i \in \mathbb{R}^{C_i \times H_i \times W_i}$ are reshaped into $\mathbf{P}^i \in \mathbb{R}^{C_i \times N_i}$, where $N_i = H_i \times W_i$ is the squeezed spatial size. Next, each element in $\mathbf{A}^i$ is obtained via a dot product of the reshaped feature maps:

$$a^i_{j,k} = \mathbf{P}^i_{j,:} \cdot \mathbf{P}^i_{k,:}, \tag{1}$$

where $a^i_{j,k}$ indicates the affinity level between channel j and channel k in layer i. Here, we adopt the dot product similarity, which considers the vectors in-between angle and the magnitude.

Secondly, from the obtained *Channel Affinity Matrix* $\mathbf{A}$, CAGM computes the *Contextual Guiding Vector* $\boldsymbol{\beta}$ in each layer, which integrates the affinity of channels into scalars. Considering row j in $\mathbf{A}^i \in C_i \times C_i$, each element $a^i_{j,:}$ represent the affinity level between the channel j and the remaining channels in layer i. Then, we normalize $a^i_{j,:}$ into the same scale, *i.e.*, between zero to one, and then squeeze $a^i_{j,:}$ into one dimension using the summation $\sum_k a^i_{j,k}$ for integration.

Finally, the *Contextual Guiding Vector* $\boldsymbol{\beta}^i$ is obtained by:

$$\boldsymbol{\beta}_j^i = \frac{\sum_k a_{j,k}^i - Min(\sum_k a_{:,k}^i)}{Max(\sum_k a_{:,k}^i) - Min(\sum_k a_{:,k}^i)} \in [0,1], \quad (2)$$

where $Max(\cdot)$ and $Min(\cdot)$ compute the maximum and minimum value along the channel dimension. We denote each scalar in $\boldsymbol{\beta}$ as the *Guiding Factor*, where $\boldsymbol{\beta}_j^i$ is the *Guiding Factor* for channel j in layer i. Iterating all channels in this layer i, we can acquire their corresponding *Guiding Factor*, which represents individual integrated interdependent level with other channels given the network input.

Since the activation on feature maps varies from the network inputs, the corresponding $\boldsymbol{\beta}$ will be input-dependent as well. To incorporate the mini-batch training, we conduct additional *min-max* normalization on $\boldsymbol{\beta}$ along the batch dimension each mini-batch before CAGS, in order to achieve the batch-dependent integration. In this way, both the numerical stability and scale of $\boldsymbol{\beta}$ are maintained. It is worth mentioning that no additional trainable parameters or supervision are introduced in this module, and it is also different from the self-attention mechanism.

Note that, for SegNet, our CAGMs are applied to the feature maps in the encoder part, in which, multiple pooling and sub-sampling operations are implemented to extract the potential spatial context and to achieve translation invariance over spatial dimension. Similarly, for PSPNet or ICNet with a dilated backbone, our CAGMs are concatenated to the Pyramid Pooling module, where the features carry richer sub-region contextual information in different scales. In Figure 1, we illustrate such design, and the ablation study in Section 5 validates its effectiveness.

3.2. Context-aware Guided Sparsification (CAGS)

As the CAGM is applied over the well-optimized unpruned models, the initial $\boldsymbol{\beta}$ is able to reflect the meaningful channel-to-channel interdependency information from different context. However, the computed $\boldsymbol{\beta}$ is dependent to each input. To leverage $\boldsymbol{\beta}$ under all given data, we introduce the *Context-aware Guided Sparsification* (CAGS), which empirically proves to be valid for contextual informative channels selection.

For notation, given a model Φ with the parameters set θ, we first denote the vanilla loss function for semantic segmentation in Equation 3:

$$L(\theta) = \frac{1}{n}\sum_{k=1}^{n} J(\mathbf{Y}_k; \Phi(\mathbf{X}_k, \theta)) + R_w(\theta), \quad (3)$$

where $\mathbf{X}_k \in \mathbb{R}^{3\times H\times W}$ and $\mathbf{Y}_k \in \mathbb{R}^{H\times W}$ are the k_{th} input RGB image and its pixel-wise semantic labels, $k = 1,\ldots,n$. $J(\cdot)$ is the pixel-level standard segmentation loss. Mostly, we wish to minimize the standard loss together with the regularization term R_w (*e.g.*, L2 regularization) on the parameters set to achieve a better generalization and to avoid over-fitting.

To enable the CNNs to obtain better generalization and faster convergence, batch normalization (BN) [18] has been generally employed in most modern architectures, and it can be formulated as follows:

$$y = \gamma\frac{x-\mu}{\sqrt{\sigma^2+\epsilon}} + \beta, \quad (4)$$

where μ and σ are the statistical mean and standard deviation over the mini-batch input features. The affine parameters γ and β are the learnable scaling and shifting factors. For channel pruning, the scaling factor γ in the BN layers can be an importance indicator for the corresponding channels. Channels with close-to-zero γ can be regarded as redundant since it is scaled to near-zero activation and they contribute less to the prediction.

To learn the redundant channels, the straightforward attempt is to induce sparsity on the whole scaling factors set ζ with vanilla L1 regularization [20] along with $L(\theta)$:

$$\min_{\theta} L(\theta) + \lambda\sum_{\gamma\in\zeta} R_s(\gamma), \quad (5)$$

where $R_s(\cdot) = |\cdot|$ is the regular sparsification term and λ is a constant to determine the global sparsification strength.

While this approach can enforce the selective γ to zero, it will lead to unnecessary performance loss and misclassification in semantic segmentation because the scaling factors of informative channels are penalized equally to others. This leads to unsatisfactory pruning results. We consider such an approach as our baseline for comparison (see Figure 2). In fact, when different channels combination or association provide different contextual hints, this provides a prior to help differentiate useful channels. The sparsification penalty should be de-emphasized for the channels that give overall useful information from the input images.

As such, we incorporate the *Contextual Guiding Vector* $\boldsymbol{\beta}$ as a prior to the sparsification of channel-wise scaling factors to adaptively impose different penalty strength, namely *Context-aware Guided Sparsification* (CAGS):

$$\min_{\theta} L(\theta) + \lambda_1\sum_{\gamma\in\zeta/\delta} R_s(\gamma) + \lambda_2\sum_{\substack{\gamma\in\delta\\ \beta\in\delta'}} (1-\beta)R_c(\gamma), \quad (6)$$

where ζ and δ are the scaling factors set over all the BN layers in Φ, and over the selective BN layers with the CAGM respectively. δ' is the corresponding set of guiding factors. Note that $R_c(\cdot)$ is the contextual penalty term for sparsifying the scaling factor with the context guiding factor β, where $R_c(\cdot) = |\cdot|$ and is multiplied with $(1-\beta)$ accordingly. Furthermore, λ_1 and λ_2 are the hyperparameters to

Dataset		Methods	mIoU(%)	#Params(M)(%↓)	#FLOPs(G)(%↓)
CamVid	SegNet	Unpruned	55.60	29.45	106.73
		FPGM[14]	52.54	15.63(46.92%↓)	32.95(69.13%↓)
		NS-20%[20]	54.78	12.25(58.40%↓)	47.81(55.20%↓)
		BN-Scale-20%	55.73	12.50(57.39%↓)	44.54(58.27%↓)
		Ours-20%	**57.12**	11.47(61.05%↓)	53.14(50.21%↓)
		Ours-30%	56.37	**6.03(79.52%↓)**	**30.01(71.88%↓)**
Cityscapes	PSPNet101	Unpruned	78.40	70.44	557.04
		FPGM[14]	74.84	**36.09(48.76%↓)**	**280.68(49.61%↓)**
		NS-60%[20]	75.70	48.47(31.19%↓)	368.03(33.93%↓)
		BN-Scale-60%	74.88	48.81(30.71%↓)	370.49(33.49%↓)
		Ours-60%	**77.82**	47.84(32.08%↓)	363.21(34.80%↓)
		Ours-70%	75.27	39.74(43.58%↓)	296.25(46.82%↓)
	PSPNet50	Unpruned	76.99	51.45	403.00
		FPGM[14]	74.59	27.06(47.40%↓)	207.31(48.56%↓)
		NS-50%[20]	73.57	23.61(54.11%↓)	199.78(50.43%↓)
		BN-Scale-50%	73.85	**23.59(54.15%↓)**	**199.43(50.51%↓)**
		Ours-60%	**75.59**	27.31(46.92%↓)	233.67(42.02%↓)
		Ours-70%	73.94	23.78(53.78%↓)	203.19(49.58%↓)
	ICNet	Unpruned†	64.59	12.21	40.13
		FPGM[14]	62.00	6.45(47.18%↓)	22.96(42.79%↓)
		NS-60%[20]	60.02	6.90(43.49%↓)	22.75(43.31%↓)
		BN-Scale-60%	59.68	6.96(43.00%↓)	22.82(43.13%↓)
		Ours-60%	**62.38**	**5.56(54.46%↓)**	**21.16(47.27%↓)**
	SegNet	Unpruned	56.10	29.45	326.59
		FPGM[14]	51.60	15.63(46.92%↓)	100.51(69.22%↓)
		NS-20%[20]	56.85	11.85(59.76%↓)	188.16(42.39%↓)
		BN-Scale-20%	59.95	11.92(59.52%↓)	**150.47(53.93%↓)**
		Ours-20%	**61.16**	**10.76(63.46%↓)**	178.23(45.43%↓)

† Train from scratch

Table 1. Quantitative pruning results on CamVid and Cityscapes **test set**.

determine the basic penalty strength on the regular penalty term $R_s(\cdot)$ and the contextual penalty term $R_c(\cdot)$.

When feeding different inputs during the forward pass, we consider the channels indicating overall high channel-to-channel interdependency to be contextual informative. Moreover, the contextual importance that is measured by $\boldsymbol{\beta}$ can also provide guidance for sparsification by using term $(1-\beta)$ in Equation 6. We denote it as the channel-wise contextual sparsification guidance. The larger the integrated interdependency level, the smaller the sparsification penalty strength that will be enforced due to the smaller sparsification guidance. Our proposed CAGS tends to preserve these channels during the scaling factors sparsification and penalize the rest of the channels relatively more. During back-propagation along with the standard loss, CAGS enables the model to learn to balance the segmentation target with the aim of selecting informative channels under the guidance prior. Each guidance $(1-\beta)$ adaptively scales the L1 penalty gradient for the channel-wise $\gamma \in \delta$, *i.e.* imposing less force on the contextually important channels.

As mentioned in the previous section, our purpose is to prune channels and preserve important contextual information as much as possible. After several epochs inducing sparsity with CAGS, the whole scaling factors set ζ of the cumbersome network become sparse, enabling us to determine the useless channels with the smallest scaling factors γ. Finally, we obtain a compact model after pruning and finetuning. We will provide ablation studies to illustrate the effectiveness of the proposed framework.

4. Experiments

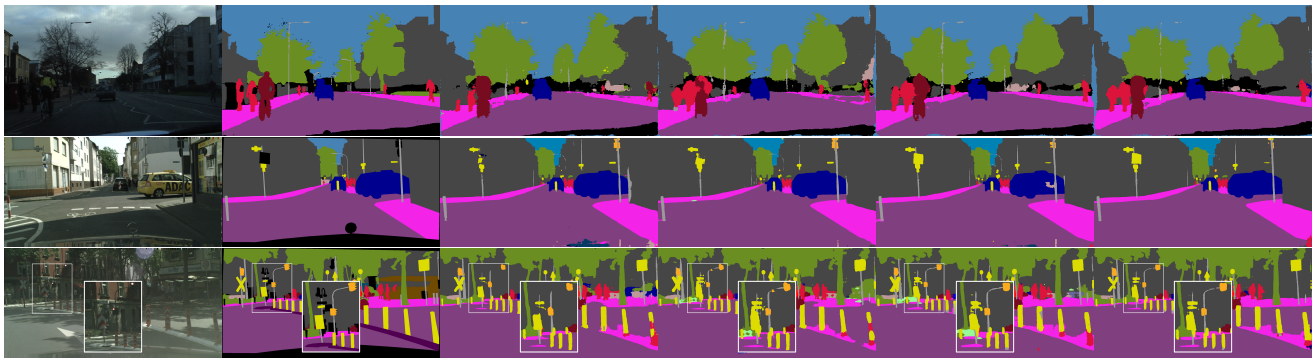
We empirically evaluate the performance of our method on various networks and two benchmarks (i.e., CamVid [3] and Cityscapes [7]). The details of benchmarks are described in the supplementary material.

4.1. Implementation Details

The implementation of our framework includes three stages as follows: normal training, sparsity inducing, and pruning and finetuning.

4.2. Normal Training

For CamVid dataset, the initial learning rate is set to 0.01, and we apply the cosine annealing decay policy $\frac{1}{2} \cdot initial\_lr \cdot (1 + \cos(\frac{iter}{total\_iter}\pi))$ for 450 epochs train-



(a) Input Image (b) Ground Truth (c) NS [20] (d) BN-Scale (e) Unpruned (f) Ours

Figure 2. Qualitative results on CamVid (top) and Cityscapes (bottom). The first row visualizes CamVid prediction results from SegNet with ratio 20% and the last two rows are from the Cityscapes dataset on PSPNet101 with ratio 60%. The white rectangle highlights the regions where our approach is able to preserve more detailed information compared to the baseline.

ing in mini-batch size 8. We employ a stochastic gradient descent (SGD) optimizer with momentum coefficient 0.9 and weight decay coefficient 0.0005.

For Cityscapes, we trained SegNet [1] with 512×1024 resized input for 450 epochs in batch size 8, and Adam optimizer is used. For PSPNet [37], we randomly crop the input images into the size 713×713 and train the model using Inplace-ABN [30] and SGD optimizer with momentum for 200 epochs. To train ICNet, we use the same setting from PSPNet but with inputs in full size. We employ a poly decay strategy with power 0.9 on the learning rate, which is multiplied by $(1 - \frac{iter}{total\_iter})^{0.9}$ in each iteration. The initial learning rates are 0.001 for SegNet, and 0.01 for PSPNet and ICNet. In addition, multiple data augmentations are adopted, such as random scaling, random rotation, random translation, and random flipping. Due to the performance loss when using the pretrained PSPNet50 backbone in the Caffe framework for ICNet, we re-implemented ICNet and trained it from scratch for our evaluation, following the implementation in the original paper.

4.3. Sparsity Inducing

Before pruning, we apply our CAGM on the optimized models after the normal training stage first and then induce sparsity on the scaling factors γ with CAGS in a few epochs, to distinguish the contextual informative channels from the given training data.

After CAGS, the magnitude of scaling factors γ indicates the channel-wise saliency considering the contextual information, and can be used for pruning. The hyper-parameters λ_1 and λ_2 are set to 0.0001 and 0.001. We provide an ablation study on such settings in Figure 3, where the sparsity level with λ in different values are shown, and $\lambda_1 = 0, \lambda_2 = 0$ represents the original model. λ_2 is always 10 times as large as λ_1, in order to balance the effect of multiplying the guidance term $(1 - \beta) \in [0, 1]$. Note that, since CAGMs are applied to provide pruning guidance only, they can be harmlessly removed after inducing sparsity.

4.4. Pruning and Finetuning

Each scaling factor γ in BN regulates the channel outputs into various magnitude. The lesser the scaling factor, the lesser contribution its channel makes to the final prediction. Therefore, we can discard channels with the smallest γ in a global and greedy manner. Instead of setting layer-wise prune ratios, we assign one global prune ratio for reference and remove channels based on their channel-wise importance global ranking via one-shot pruning. In the supplementary, we show the pruned architectures that are determined automatically with our framework.

Note that the removal of a specific channel is equivalent to removing a convolution kernel in the previous layer. To maintain the network architecture, we also need to remove the following channel in all the incoming convolution kernels. To avoid the case where all pruning candidate channels are within the same layer, we preserve 10% channels in each layer. We adopt the same pruning strategy for pruning the residual block as the prior works, where the last convolution layer and downsampling layers are preserved to match the feature volume for summation. For SegNet, the max-pooling indices are shared in-between layers. Thus, for channels that are determined to be removed in either the Encoder or the Decoder part, their corresponding channels for indices sharing should also be pruned.

Since there is an inevitable performance drop after pruning, we finetune the pruned models in the same setting as the training stage, but with a smaller learning rate. On Cityscapes, the finetuning epochs is 50 for PSPNet and ICNet, and is 100 for SegNet. On Camvid, SegNet is finetuned for 200 epochs. Finally, the compact models are evaluated based on the given metrics and compared with baselines that are re-implemented in the same finetune settings.

4.5. Baselines

Existing pruning methods are not specifically tailored for semantic segmentation networks. As such, for comparison, we evaluated the impact of popular baseline pruning methods (originally evaluated for image-level classification) on widely-used semantic segmentation models. Our baselines include BN-Scale, NS (Network Slimming [20]), FPGM [14] and the recently proposed conditional computing based method, CCGN [2]. Details of baselines are discussed in the supplementary material.

FPGM is one of the state-of-the-art pruning methods for image classification, which prunes filters based on their Euclidean distance with other filters layer-wise. We follow the pruning criterion formulation and the optimal predefined layer-wise prune ratio in the literature.

BN-Scale and NS method both use the scaling factors γ in BN as the pruning indicator. The former serves as a naive baseline, and the latter is the widely-used pruning method. In BN-Scale, the original model is directly pruned based on the scaling factors magnitude after normal training. In NS, the regular sparsity will be induced on all the scaling factors before pruning.

Our method and the first two baselines belong to the automatic pruning approach, while FPGM performs pruning given the manually specified pruned ratio for each layer and results in a predefined pruned architecture [21]. The overall pruning results comparison is illustrated in Table 1, and we will show the pruning results when FPGM is implemented in an automatic pruning manner as ours in the supplementary material. Although CCGN [2] does not prune the original network, we adopted it as a baseline as it is the latest network acceleration method that provides comprehensive results on semantic segmentation. The acceleration comparison is shown in Table 3.

4.6. Quantitative and Qualitative Results

We present quantitative and qualitative comparisons in Table 1 and Figure 2. Note that the reduction in #Params and #FLOPs may not be consistent due to the pruned channels selections on different layers. We compare the pruned models with similar #Params. Additionally, the actual runtime speedups after pruning with our method, the per-class prediction performance comparison, and more visualization comparison on different images will be illustrated in the supplementary material.

In Table 1, we evaluate the pruned models using the mean Intersection-over-Union (mIoU), the number of parameters (#Params), and the floating-point operations (#FLOPs). #FLOPs of SegNet are reported based on the input size 512×1024 and 360×480 for Cityscapes and CamVid, respectively, while PSPNet and ICNet are reported in 713×713 and 1024×2048. All the test results on Cityscapes are submitted and evaluated by the benchmark server. *Ours-x%* denotes the pruning using our method with a global prune ratio x%. The same notation applies for the baselines BN-Scale and NS. Note that in the same prune ratio x%, the reduction on #Params and #FLOPs vary for different methods, due to the selection differences on channels to be pruned. Table 1 shows the pruning results on ratios x% that have similar #Params reduction and the closest performance with the unpruned model. From the results, it is evident that our approach can effectively reduce #Params and #FLOPs, compared to all the baselines. Specifically:

1. In terms of #Params reduction, the proposed method achieves the best pruning performance, in another word, much efficient pruned architectures are automatically discovered. For instance, for Cityscapes and PSPNet101, *Ours-60%* achieves 77.82 mIoU while NS-60% [20] and BN-Scale-60% can only achieve 75.70 and 74.88 mIoU with a larger number of parameters. For SegNet, *Ours-20%* is able to outperform the original model with 61% lesser parameters and 45% fewer FLOPs. These pruned models from the same unpruned are with similar #Params, but ours is more efficient.

2. The visualization on Figure 2(f) shows that the pruned models of our method can preserve most of the prediction precision on small objects from the original model, while the ones in other baselines lead to information loss and misclassification in varying degrees.

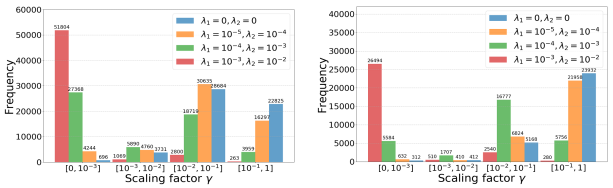
5. Ablation Studies

5.1. Pruning Ratio

Large pruning ratio may result in high model capacity loss leading to the inability to recover the segmentation performance. On the other hand, small pruning ratios will not lead to effective compression for the given requirements. Hence, the right balance between the model size and performance is necessary. We compare the model performance with different pruning ratios after finetuning in Figure 4. It can be observed that it is possible to maintain original accuracy by keeping a maximal pruning ratio within certain intervals (*e.g.*, between 0.5 and 0.7 for PSPNet).

5.2. Hyperparameters λ_1 and λ_2

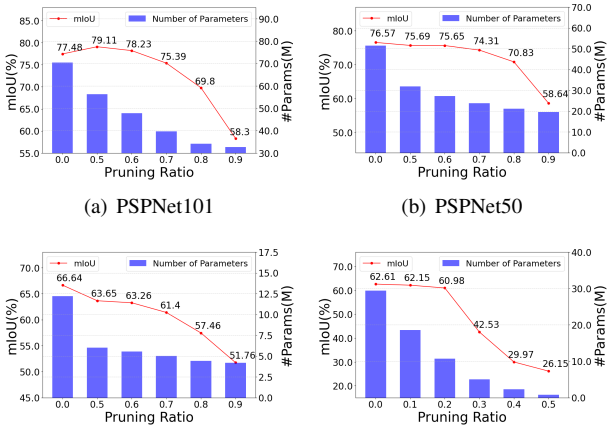
In Section 3.2, λ_1 and λ_2 are used to adjust the strength on the regular sparsification term and the contextual sparsification term, and in CAGS, pair $\lambda_1 = 0.0001, \lambda_2 = 0.001$ is preferable. As shown in Figure 3, different λ pair results in models with different sparsity level on γ. When a larger λ pair forces more γ towards zero, the model's performance will be negatively impacted as well. For instance, in the PSPNet101 experiments on Cityscapes validation set, while smaller λ pair, *i.e.*, $\lambda_1 = 0.0001, \lambda_2 = 0.001$



(a) PSPNet101 (b) PSPNet50

(c) ICNet (d) SegNet

Figure 3. Ablation study on different values of λ_1 and λ_2 pair. Histograms in different colors stand for the frequency of scaling factors γ in the models under different λ pairs after CAGS.



(a) PSPNet101 (b) PSPNet50

(c) ICNet (d) SegNet

Figure 4. Ablation study on pruning ratio and model performance. Each bar represents the number of parameters after pruning with different global ratio, and the solid line denotes their corresponding mIoU on Cityscapes validation set.

(in green), results in mIoU of 77.57 and induces suitable sparsity from the unpruned (in blue), larger λ pair, *i.e.*, $\lambda_1 = 0.0001, \lambda_2 = 0.001$ (in red), leads to a significantly sparser model with 63.63 mIoU.

5.3. Position of CAGM

We placed CAGM adjacent to the pooling layers, where the feature maps with the potential spatial context information are leveraged. To justify, in Table 2, we conduct an ablation experiment that applies CAGM on all layers (denoted as *CAGM on All*). It shows that such positioning in our framework (*CAP*) is sufficient, since it consistently leads to better pruning performance compared to *CAGM on All*, especially on lightweight models. The reason could be that the lightweight models have a relatively larger downsampling rate within fewer layers and feature maps after each downsampling operation capture richer spatial information. Using such information as guidance benefits the evaluation on the contextual informative channels. And when other feature maps that contain less useful information are also utilized by CAGM, the advantage will be less noticeable and requires more computation in run-time memory.

Models	CAP	CAGM on All	Prune Ratio	mIoU(%)
PSPNet101	✓		0.6	78.23
		✓	0.6	77.99
PSPNet50	✓		0.7	74.31
		✓	0.7	73.21
SegNet	✓		0.2	60.98
		✓	0.2	57.85
ICNet	✓		0.6	63.25
		✓	0.6	61.22

Table 2. Ablation study on the position of CAGM. *CAP* is our proposed framework, and *CAGM on All* is when we apply the CAGM on all layers. The mIoU in different architectures are reported on Cityscapes **validation set**.

Methods	mIoU(%)	#Params(M)(%↓)	#FLOPs(G)(%↓)
Unpruned	76.57	51.45	403.0
CCGN*[2]	71.90	70.44(0%↓)	23.70%↓
CCGN-1 [2]	74.40	70.44(0%↓)	23.50%↓
CCGN-2 [2]	74.70	70.44(0%↓)	5.00%↓
Ours-60%	**75.65**	**27.31(46.92%↓)**	**233.67(42.02%↓)**

* Without pretrain

Table 3. Comparison with CCGN [2] on Cityscapes **validation set**.

6. Conclusion

The proposed *Context-aware Pruning* framework utilizes channels association to exploit parameters redundancy in terms of contextual information for accelerating semantic segmentation. Our method effectively preserves contextual informative channels after pruning. Experiments on benchmarks show our advantages over popular pruning baselines for both large and lightweight state-of-the-art architectures. Our framework can complement other pruning schemes (*e.g.*, iterative pruning) or compression techniques (*e.g.*, quantization) to improve the performance further, and also has great potential for other challenging vision tasks.

7. Acknowledgments

This research project is supported in part by the National Research Foundation Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme with the Technical University of Munich at TUMCREATE.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[2] Babak Ehteshami Bejnordi, Tijmen Blankevoort, and Max Welling. Batch-shaping for learning conditional channel gated networks. In *International Conference on Learning Representations*, 2020.

[3] Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *European conference on computer vision*, pages 44–57. Springer, 2008.

[4] 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.

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

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[8] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[9] Xitong Gao, Yiren Zhao, Łukasz Dudziak, Robert Mullins, and Cheng zhong Xu. Dynamic channel pruning: Feature boosting and suppression. In *International Conference on Learning Representations*, 2019.

[10] Ariel Gordon, Elad Eban, Ofir Nachum, Bo Chen, Hao Wu, Tien-Ju Yang, and Edward Choi. Morphnet: Fast & simple resource-constrained structure learning of deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1586–1595, 2018.

[11] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pages 1135–1143, 2015.

[12] 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.

[13] Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang. Soft filter pruning for accelerating deep convolutional neural networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2234–2240, 2018.

[14] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2019.

[15] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1389–1397, 2017.

[16] Zehao Huang and Naiyan Wang. Data-driven sparse structure selection for deep neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 304–320, 2018.

[17] Zhongzhan Huang, Xinjiang Wang, and Ping Luo. Convolution-weight-distribution assumption: Rethinking the criteria of channel pruning. *arXiv preprint arXiv:2004.11627*, 2020.

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[19] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[20] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2736–2744, 2017.

[21] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. In *International Conference on Learning Representations*, 2018.

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[23] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE international conference on computer vision*, pages 5058–5066, 2017.

[24] Deepak Mittal, Shweta Bhardwaj, Mitesh M Khapra, and Balaraman Ravindran. Recovering from random pruning: On the plasticity of deep convolutional neural networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 848–857. IEEE, 2018.

[25] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *International Conference on Machine Learning*, pages 2498–2507, 2017.

[26] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11264–11272, 2019.

[27] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016.

[28] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[30] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5639–5647, 2018.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[32] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[33] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Advances in neural information processing systems*, pages 2074–2082, 2016.

[34] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018.

[35] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2018.

[36] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 405–420, 2018.

[37] 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.

[38] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 593–602, 2019.