Rethinking Knowledge Distillation with Raw Features for Semantic Segmentation

Tao Liu*, Chenshu Chen*,†, Xi Yang, Wenming Tan
Hikvision Research Institute
Hangzhou, Zhejiang, China
{liutao46, chenchenshu, yangxi6, tanwenming}@hikvision.com

Abstract

Most existing knowledge distillation methods for semantic segmentation focus on extracting various sophisticated knowledge from raw features. However, such knowledge is usually manually designed and relies on prior knowledge as in traditional feature engineering. In this paper, we aim to propose a simple and effective feature distillation method using raw features. To this end, we revisit the pioneering work in feature distillation, FitNets, which simply minimizes the mean squared error (MSE) loss between the teacher and student features. Our experiments show that this naive method yields good results, even surpassing some well-designed methods in some cases. However, it requires carefully tuning the weight of distillation loss. By decomposing the loss function of FitNets into a magnitude difference term and an angular difference term, we find the weight of the angular difference term is affected by the magnitudes of the teacher features and the student features. We experimentally show that the angular difference term plays a crucial role in feature distillation and the magnitude of the features produced by different models may vary significantly. Therefore, it is hard to determine a suitable loss weight for various models. To avoid the weight of the angular distillation term being affected by the magnitude of the features, we propose Angular Distillation and explore distilling angular information along different feature dimensions for semantic segmentation. Extensive experiments show that our simple method exhibits great robustness to hyper-parameters and achieves state-of-the-art distillation performance for semantic segmentation.

1. Introduction

Recent works on backbones [7, 21, 31] and segmentation frameworks [3, 29, 33] have greatly improved the performance of semantic segmentation. However, these high-performance models often require a lot of memory and computational overhead. Lightweight models are preferred in real-time applications due to limited resources. As a result, there is growing interest in how to reduce the model size while maintaining decent performance.

The knowledge distillation (KD) introduced by [8] was proven to be a promising way to solve this problem. Its key idea is to transfer the knowledge from a cumbersome model (teacher) to a compact one (student). [8] defines the knowledge as soft labels produced by the teacher and supervises the student with both ground truth labels and soft labels. FitNets [17] extends this idea to the intermediate representation of the model by making the student directly mimic the teacher's hidden layer features. Inspired by this, many feature-based KD methods emerged later. Instead of distilling the raw features as in FitNets [17], most existing feature-based methods prefer to extract various forms of knowledge from raw features, such as attention map [30], Gramian matrix [28], pair-wise similarity [13] and low-level texture knowledge [9]. However, this kind of knowledge is usually manually designed and relies on various prior knowledge as in traditional feature engineering.

In this paper, we aim to propose a simple and effective feature distillation method using raw features. We therefore revisit the simplest feature distillation method proposed in FitNets [17], which minimizes the mean squared error (MSE) loss between the teacher and student features. In this paper, we refer to the KD method proposed by FitNets [17] as naive feature distillation. We are surprised to find that this naive feature distillation method can achieve good results, even outperforming some recent methods that are carefully designed for semantic segmentation (see Fig. 1). However, the performance of this method is sensitive to the loss weight. The appropriate loss weight for different models varies significantly in some cases. To find out the underlying reasons, we decompose the loss function of FitNets into a magnitude difference term and an angular difference term, and reveal that the weight of the angular differ-

[*] Equal contribution
[†] Corresponding author

ence term is affected by the magnitude of the feature. We further demonstrate through experiments that the angular difference term is the key to achieving good performance, while the magnitude of the features produced by different models may vary greatly. This explains why it is difficult for the naive feature distillation method to determine a loss weight that suits different models.

In order to make the weight of the angular difference term independent of the magnitude of the features, we propose to distill only the angular information of the features for semantic segmentation. More importantly, we explore distilling angular information along different feature dimensions, and demonstrate that the dimension of angular information has a significant impact on the distillation performance of semantic segmentation. Although some methods also utilize angular information [15, 20], none of them have considered the dimension of angular information, which is critical for semantic segmentation. Furthermore, as discussed in Sec. 2.1, their approaches to distill the angular information are quite sophisticated, and they also involve other forms of feature distillation. In comparison, our method utilizes a straightforward angular distillation loss. Extensive experiments on Cityscapes, Pascal VOC, and ADE20K demonstrate the effectiveness and robustness of our method.

Our main contributions are summarized as follows:

- We decompose the loss function of FitNets [17] into a magnitude difference term and an angular difference term, and point out the reason why it is sensitive to the loss weight is that the weight of the angular difference term is affected by the magnitude of the feature.
- We explore distilling angular information along different feature dimensions and show that the dimension of angular distillation has a significant impact on the distillation performance of semantic segmentation.
- Without relying on the sophisticated knowledge of manual design, our method achieves state-of-the-art distillation performance for semantic segmentation and is robust to hyper-parameters.

2. Related work

2.1. Knowledge distillation

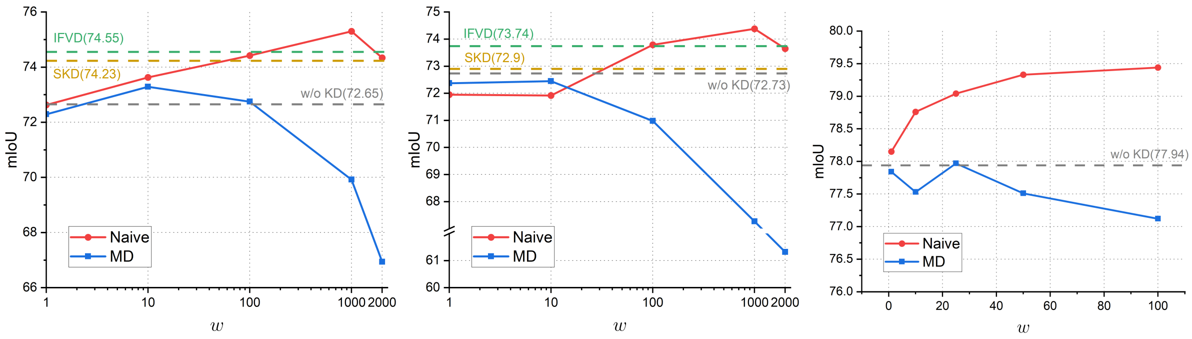
Existing KD methods can be roughly divided into logits-based, feature-based and relation-based according to the type of knowledge. Logits-based methods use class probabilities of the teacher as soft labels to supervise the student. Feature-based methods take the features of intermediate layers as knowledge. Relation-based methods focus on the relationships between different layers or data samples.

Among these methods, the feature-based methods are more related to this paper. FitNets [17] is the first KD method to take the features of the intermediate layers as knowledge. After that, many methods focusing on different aspects of feature distillation have been proposed, such as designing various forms of new knowledge from raw features [16, 30], changing the teacher's or student's training strategies to facilitate distillation [11, 35], and adaptively utilizing multiple layers of features for distillation [1, 10]. Differently, we revisit the naive feature distillation method introduced in FitNets [17] and analyze the possible reasons for its limited performance.

Given the existence of several KD methods that leverage the angular information [15, 20], we summarize the key distinctions between our method and them. Firstly, we investigate distilling angular information along different feature dimensions, demonstrating that the dimensions of angular information are critical for semantic segmentation. Although prior studies have employed angular distillation for image classification [15, 20], the effects of the feature dimensions during angular distillation have not been analyzed. As for semantic segmentation, to the best of our knowledge, there are currently no methods that employ angular distillation. Secondly, we employ simple L2 normalization on the raw features to directly derive the angular information for distillation. In contrast, existing methods employ more involved techniques to distill angular information. For instance, [20] adopts locality-sensitive hashing (LSH) to help the student mimicking the direction of teacher features, while [15] requires constructing triplets of samples and calculating the angle formed by three samples in the feature space. Thirdly, in terms of the composition of the feature distillation loss, our method consists solely of an angular distillation term. In contrast, existing methods typically incorporate additional distillation losses beyond angular differences, such as MSE loss in [20] or Euclidean distance loss in [15], which implicitly distills both magnitude and angular information simultaneously.

2.2. Knowledge distillation for semantic segmentation

Applying KD methods for image classification to semantic segmentation in a straightforward way may not yield satisfactory results. As a result, some KD methods tailored for semantic segmentation have been proposed. [24] uses the local similarity between a pixel and its 8 neighbors on the feature map as knowledge. [13] distills the long-range dependency by computing the pairwise similarity on the feature map and enforces high-order consistency between the outputs of the teacher and student through adversarial learning. [22] proposes to transfer the intra-class feature variation of the teacher to the student. [19] focuses on channel information by softly aligning the activation of each channel between the teacher and student, which is more effective on logits than on features. Unlike them, our method does not



(a) T:PSPNet-R101, S:PSPNet-R18

(b) T:PSPNet-R101, S:PSPNet-MV2

(c) T:UPerNet-SwinB, S:UPerNet-SwinT

Figure 1. Distillation performance at varying loss weights on Cityscapes validation set. T: Teacher. S: Student. Naive: Naive feature distillation. MD: Magnitude Distillation. w/o KD: Baseline student without KD. SKD [13] and IFVD [22] are prior KD methods for semantic segmentation. w: Loss weight of $\mathcal{L}_{naive}$ or $\mathcal{L}_{md}$.

rely on sophisticated knowledge of manual design and tedious distillation strategies such as adversarial learning. Extensive experiments on semantic segmentation demonstrate the simplicity and effectiveness of our method.

3. Method

3.1. Analysis of naive feature distillation

In this paper, we refer to the KD method proposed by FitNets [17] as naive feature distillation. It encourages the student to have the same feature activation as the teacher. Let $\boldsymbol{F}^s \in \mathbb{R}^{C\times H\times W}$ and $\boldsymbol{F}^t \in \mathbb{R}^{C\times H\times W}$ denote the feature maps of the student and teacher, respectively, where C is the number of channels, H and W are the height and width. For simplicity, we assume that $\boldsymbol{F}^s$ has the same dimensions as $\boldsymbol{F}^t$. This can be achieved by applying a feature transformation (*e.g.*, 1×1 convolution) to $\boldsymbol{F}^s$. The naive feature distillation minimizes the mean squared error (MSE) loss between $\boldsymbol{F}^s$ and $\boldsymbol{F}^t$:

$$\mathcal{L}_{naive} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{F}_i^t - \boldsymbol{F}_i^s)^2 \tag{1}$$

where $N = C \times H \times W$.

As shown in Fig. 1, we experimentally find that the naive feature distillation is able to achieve good results, and even surpasses some well-designed methods for semantic segmentation (*i.e*. SKD [13] and IFVD [22]). However, the naive feature distillation is sensitive to the loss weight, thus its good performance relies on carefully tuning the hyperparameters. For example, when the teacher is PSPNet-R101 and the student is PSPNet-R18 or PSPNet-MV2, it requires a large loss weight (*e.g.*, 1000) to get good results. Instead, a relatively small loss weight is appropriate when the teacher is UPerNet-SwinB and the student is UPerNet-SwinT.

To figure out why the naive feature distillation is sensible to the loss weight, we start by analyzing its loss function. From the perspective of vectors, we can reformulate $\boldsymbol{F}^s$ and $\boldsymbol{F}^t$ as:

$$\begin{aligned} \boldsymbol{F}^t &= ||\boldsymbol{F}^t||\boldsymbol{y} = m\boldsymbol{y} \\ \boldsymbol{F}^s &= ||\boldsymbol{F}^s||\boldsymbol{x} = n\boldsymbol{x} \end{aligned} \tag{2}$$

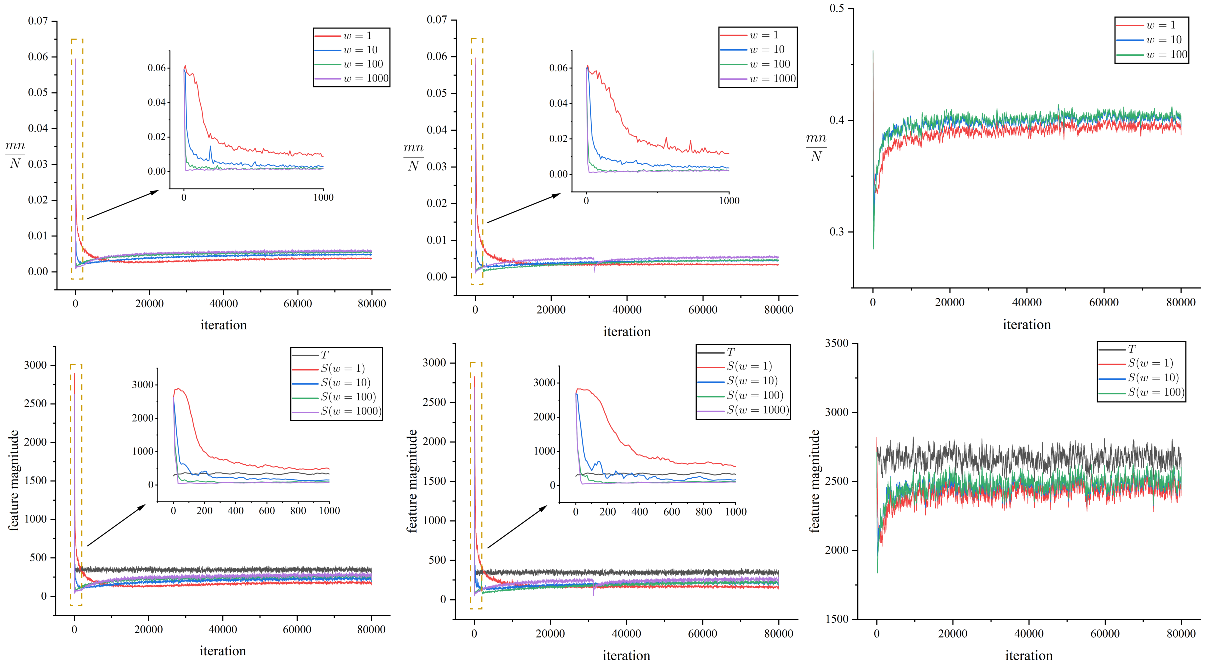
where n and m denote the magnitudes of $\boldsymbol{F}^s$ and $\boldsymbol{F}^t$, respectively, and $\boldsymbol{x}$ and $\boldsymbol{y}$ are unit vectors. Then $\mathcal{L}_{naive}$ in Eq. (1) can be reformulated as:

$$\begin{aligned} \mathcal{L}_{naive} &= \frac{1}{N}\sum_{i=1}^{N}(m\boldsymbol{y}_i - n\boldsymbol{x}_i)^2 \\ &= \frac{1}{N}(m^2\sum_{i=1}^{N}\boldsymbol{y}_i^2 + n^2\sum_{i=1}^{N}\boldsymbol{x}_i^2 - 2mn\sum_{i=1}^{N}\boldsymbol{y}_i\boldsymbol{x}_i) \\ &= \frac{1}{N}(m^2||\boldsymbol{y}||^2 + n^2||\boldsymbol{x}||^2 - 2mn\,\boldsymbol{y}\cdot\boldsymbol{x}) \\ &= \frac{1}{N}(m^2 + n^2 - 2mn\cos\theta) \\ &= \frac{1}{N}[(m-n)^2 + 2mn(1-\cos\theta)] \end{aligned} \tag{3}$$

where θ denotes the angle between $\boldsymbol{x}$ and $\boldsymbol{y}$, *i.e*., the angle between $\boldsymbol{F}^s$ and $\boldsymbol{F}^t$. The first term in Eq. (3) minimizes the magnitude difference between $\boldsymbol{F}^s$ and $\boldsymbol{F}^t$, and the second term minimizes the angular difference between $\boldsymbol{F}^s$ and $\boldsymbol{F}^t$ but is affected by the magnitude.

Inspired from Eq. (3), we record the values of $\frac{mn}{N}$, m, and n during training in Fig. 2. It can be seen that in the case of PSPNet-R101 as the teacher and PSPNet-R18 (Fig. 2a) or PSPNet-MV2 (Fig. 2b) as the student, m has a relatively small value, and n decreases rapidly at the beginning and then keeps at a small value close to m. Since N is usually a large value ($N = 2048 \times 64 \times 128 = 2^{24}$ in this case), the value of $\frac{mn}{N}$ is quite small.

Considering that $\frac{2mn}{N}$ is the weight of the angular difference term in Eq. (3) and a large loss weight (*e.g.*, 1000) is



(a) T:PSPNet-R101, S:PSPNet-R18 (b) T:PSPNet-R101, S:PSPNet-MV2 (c) T:UPerNet-SwinB, S:UPerNet-SwinT

Figure 2. The values of $\frac{mn}{N}$, m, and n in Eq. (3) during training for the naive feature distillation on Cityscapes. The first row records the values of $\frac{mn}{N}$. The second row records the magnitudes of the teacher features (m) and the student features (n). Note that the parameters of the teacher model are fixed, so the value of m for a sample is constant during training. T: Teacher. S: Student. w: Loss weight of $\mathcal{L}_{naive}$.

required to get good performance for PSPNet-R18 (Fig. 1a) and PSPNet-MV2 (Fig. 1b), we initially assume that a large loss weight, in this case, can ensure a suitable factor for the angular difference term so that the role of the angular difference term for feature distillation can be fully exploited.

To further verify the above assumption, we define Magnitude Distillation, which minimizes the first term in Eq. (3), *i.e.*,

$$\mathcal{L}_{md} = \frac{1}{N}(||\boldsymbol{F}^t|| - ||\boldsymbol{F}^s||)^2 \qquad (4)$$

As shown in Fig. 1, Magnitude Distillation apparently fails to reach the performance of the naive feature distillation, even far worse than the baseline without distillation in some cases. This indicates that the magnitude difference term contributes little to the naive feature distillation, while it is the angular difference term that plays a crucial role.

As for the case where the teacher is UPerNet-SwinB and the student is UPerNet-SwinT (Fig. 2c), the value of $\frac{mn}{N}$ is clearly larger than that in Figs. 2a and 2b since both m and n have a relatively large value. As a result, the naive feature distillation can get good results (Fig. 1c) with a relatively small loss weight (*e.g.*, 100).

In summary, the angular difference term is crucial for the naive feature distillation but the weight of the angular difference term is affected by the magnitude of the features. Since the magnitude of the features produced by different models may vary significantly, it is hard to determine a suitable loss weight for various models.

3.2. Angular distillation for semantic segmentation

Aiming to make the weight of the angular difference term independent of the feature's magnitude, we first attempt to turn the magnitudes of the teacher and student features into constant values. A simple way is to perform L2 normalization for the teacher and student features before calculating the MSE loss. This results in $m = n = 1$ and the weight of the angular difference term becomes $\frac{2}{N}$. Considering that $\frac{2}{N}$ is a very small value in most cases, we then eliminate N in Eq. (3) to make the weight of the angular difference term have a reasonable value. Specifically, we minimize the following loss :

$$\mathcal{L}_{lad} = \sum_{i=1}^{N}(\frac{\boldsymbol{F}_i^s}{||\boldsymbol{F}^s||} - \frac{\boldsymbol{F}_i^t}{||\boldsymbol{F}^t||})^2 \qquad (5)$$

Similar to Eq. (3), Eq. (5) can be reformulated as:

$$\mathcal{L}_{lad} = 2(1 - \cos\theta) \qquad (6)$$

where θ denotes the angle between $\boldsymbol{F}^s$ and $\boldsymbol{F}^t$. It means that Eq. (5) essentially treats the *entire features* of a layer from the teacher or student as a vector and minimizes the

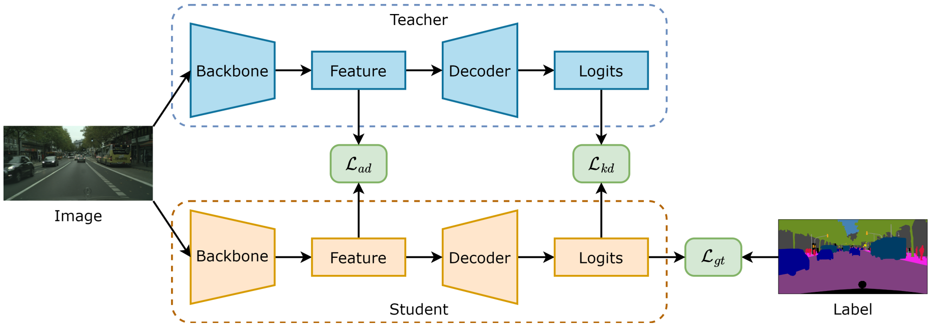


Figure 3. Our distillation pipeline for semantic segmentation. $\mathcal{L}_{ad}$ is the proposed feature distillation loss, which can be $\mathcal{L}_{lad}$, $\mathcal{L}_{cad}$ or $\mathcal{L}_{pad}$. $\mathcal{L}_{kd}$ is the conventional distillation loss [8] on logits. $\mathcal{L}_{gt}$ is the cross-entropy loss for semantic segmentation.

angular difference between them. Therefore, we refer to the proposed method that minimizes the loss in Eq. (5) as Layer-wise Angular Distillation (LAD).

The feature used for distillation in semantic segmentation is a fine-grained tensor that contains both spatial and channel dimensions. The features of different dimensions usually carry different information. Therefore, we argue that it is necessary to take the dimension of the feature into account when performing Angular Distillation for semantic segmentation.

In addition to treating the features of a layer as a vector like LAD, we can treat the features of a channel as a vector, and minimize the angular difference between features of the same channel from the teacher and student. We refer to this as Channel-wise Angular Distillation (CAD). Similar to Eq. (5), it minimizes the following loss:

$$\mathcal{L}_{cad} = \frac{1}{C}\sum_{c=1}^{C}\sum_{h=1}^{H}\sum_{w=1}^{W}\left(\frac{\boldsymbol{F}^{s}_{c,h,w}}{||\boldsymbol{F}^{s}_{c,:,:}||} - \frac{\boldsymbol{F}^{t}_{c,h,w}}{||\boldsymbol{F}^{t}_{c,:,:}||}\right)^2 \tag{7}$$

where $\boldsymbol{F}_{c,:,:} \in \mathbb{R}^{H\times W}$ denotes the features of the c-th channel.

Furthermore, we can also treat the features of each spatial point as a vector, and minimize the angular difference between features of the same spatial point from the teacher and student. We refer to this as Point-wise Angular Distillation (PAD), and its loss is as follows:

$$\mathcal{L}_{pad} = \frac{1}{H\times W}\sum_{c=1}^{C}\sum_{h=1}^{H}\sum_{w=1}^{W}\left(\frac{\boldsymbol{F}^{s}_{c,h,w}}{||\boldsymbol{F}^{s}_{:,h,w}||} - \frac{\boldsymbol{F}^{t}_{c,h,w}}{||\boldsymbol{F}^{t}_{:,h,w}||}\right)^2 \tag{8}$$

where $\boldsymbol{F}_{:,h,w} \in \mathbb{R}^{C}$ denotes the features of the spatial point (h, w).

The difference between LAD, CAD, and PAD is that the dimensions of the angular information used for distillation are different. The comparison of these three Angular Distillation methods is in Sec. 4.4.

Our distillation pipeline for semantic segmentation is shown in Fig. 3. Following the previous methods [13, 19, 22], we apply the conventional distillation loss [8] on logits as well. The total loss of our pipeline is as follows:

$$\mathcal{L} = \lambda_{ad}\mathcal{L}_{ad} + \lambda_{kd}\mathcal{L}_{kd} + \mathcal{L}_{gt} \tag{9}$$

where $\mathcal{L}_{ad}$ is our feature distillation loss, which can be $\mathcal{L}_{lad}$, $\mathcal{L}_{cad}$ or $\mathcal{L}_{pad}$. $\mathcal{L}_{kd}$ is the conventional distillation loss [8] on logits, and $\mathcal{L}_{gt}$ is the cross-entropy loss for semantic segmentation.

4. Experiments

4.1. Datasets

Our experiments are conducted mainly on three popular semantic segmentation datasets. **Cityscapes** [4] is a dataset for urban scene understanding that contains 2975/500/1525 finely annotated images for train/val/test. It contains 30 classes and 19 of them are used for evaluation. **Pascal VOC** [5] contains 20 common objects and one background class. We use the augmented dataset with extra annotations provided by [6] resulting in 10582 and 1449 images for train and validation. **ADE20K** [34] contains 150 classes and is divided into 20210/2000/3352 images for training/val/test.

4.2. Implementation details

Network architectures. Following the previous methods [13, 19, 22, 25, 27], we adopt PSPNet [33] or DeepLabV3 [2] with ResNet101 [7] backbone as the teacher, and adopt different segmentation models (PSPNet and DeepLabV3) and backbones (ResNet18 and MobileNetV2 [18]) as the student. When there is a different number of channels between the teacher and student features, a 1×1 convolution layer followed by BN and ReLU is applied to the student features for dimension alignment.

Training details. We use the pretrained teacher model and keep its parameters fixed during distillation. Unless otherwise stated, we use Stochastic Gradient Descent (SGD) as the optimizer with a batch size of 16, a weight decay of

Method	val mIoU (%)		
	Cityscapes	**VOC**	**ADE20K**
T: PSPNet-R101	79.76	78.52	44.39
S: PSPNet-R18	72.65	71.35	35.03
+ SKD [13]	74.23	72.01	35.26
+ IFVD [22]	74.55	72.00	35.92
+ CWD [19]	75.91	73.49	36.78
+ MGD [27]	75.90	74.98	36.84
+ CAD (Ours)	76.77	75.72	38.99
+ LAD (Ours)	**76.86**	**75.74**	**39.63**
S: PSPNet-MV2	72.73	69.14	33.33
+ SKD [13]	72.90	69.62	33.39
+ IFVD [22]	73.74	69.45	33.85
+ CWD [19]	74.73	71.28	35.26
+ MGD [27]	71.42	52.67	20.36
+ CAD (Ours)	74.73	73.03	36.82
+ LAD (Ours)	**75.76**	**74.13**	**38.92**
S: DeepLabV3-R18	74.96	71.98	37.19
+ SKD [13]	75.32	73.03	36.91
+ IFVD [22]	76.01	72.87	37.66
+ CWD [19]	77.13	73.78	38.64
+ CAD (Ours)	**77.24**	76.31	39.44
+ LAD (Ours)	77.23	**76.33**	**41.12**
S: DeepLabV3-MV2	73.98	69.92	35.14
+ SKD [13]	75.78	70.13	35.11
+ IFVD [22]	75.24	70.32	35.35
+ CWD [19]	76.59	71.68	36.49
+ CAD (Ours)	77.36	74.91	37.91
+ LAD (Ours)	**77.47**	**74.93**	**39.66**

Table 1. Comparison with state-of-the-art methods on validation sets of Cityscapes, Pascal VOC and ADE20K. "T" and "S" denote the teacher and student, respectively. "R101", "R18" and "MV2" denote ResNet101, ResNet18 and MobileNetV2, respectively. The **best**/second best results are marked in bold/underline.

Method	test mIoU (%)	
	Cityscapes	**ADE20K**
T: PSPNet-R101	78.14	36.35
S: PSPNet-R18	73.02	29.13
+ CWD [19]	74.73	29.93
+ MGD [27]	74.20	30.17
+ CAD (Ours)	**75.01**	31.35
+ LAD (Ours)	74.94	**32.42**
S: PSPNet-MV2	72.41	27.27
+ CWD [19]	74.09	28.48
+ CAD (Ours)	74.30	30.43
+ LAD (Ours)	**74.70**	**31.94**

Table 2. Comparison with state-of-the-art methods on Cityscapes and ADE20K test sets. The **best**/second best results are marked in bold/underline.

Method	val mIoU (%)		
	Cityscapes	**VOC**	**ADE20K**
T: DeepLabV3-R101*	78.07	77.67	42.70
S: DeepLabV3-R18*	74.21	73.21	33.91
+ CIRKD [25]*	76.38	74.50	35.41
+ CAD (Ours)	**76.81**	74.94	37.18
+ LAD (Ours)	76.78	**75.53**	**38.61**

Table 3. Comparison with CIRKD [25] on validation sets of Cityscapes, Pascal VOC, and ADE20K. The **best**/second best results are marked in bold/underline. * indicates results from CIRKD [25].

Method	test mIoU (%)	
	Cityscapes	**ADE20K**
T: DeepLabV3-R101*	77.46	35.30
S: DeepLabV3-R18*	73.45	28.80
+ CIRKD [25]*	75.05	29.87
+ CAD (Ours)	**75.33**	30.92
+ LAD (Ours)	75.20	**31.37**

Table 4. Comparison with CIRKD [25] on test sets of Cityscapes and ADE20K. The **best**/second best results are marked in bold/underline. * indicates results from CIRKD [25].

0.0005, and a momentum of 0.9. We use the "poly" learning rate policy where the learning rate equals to $base\_lr * (1 - \frac{iter}{max\_iter})^{power}$. We set the base learning rate to 0.01 and power to 0.9. We train 80k iterations for Cityscapes and Pascal VOC and 160k iterations for ADE20K. We apply random horizontal flipping, random scaling (from 0.5 to 2.0), and random cropping on the input images as data augmentation during training. The crop size for Cityscapes, Pascal VOC, and ADE20K are 512×1024, 512×512, and 512×512, respectively. We use single-scale testing for all datasets. Unless stated, the features from the last layer of the backbone are used for distillation in our method. λ_{ad} is set to 10, and λ_{kd} is set to 10 following [13, 19, 22].

4.3. Comparison with state-of-the-art methods

We compare our method with recent KD methods for semantic segmentation on Cityscapes, Pascal VOC, and ADE20K. Due to the poor performance of their teacher and student baselines, we re-implemented SKD [13], IFVD [22] and CWD [19] based on their released code. The hyperparameters related to distillation loss are set according to their recommended values. The results of MGD [27] are reproduced by us using the code released by the authors. For a fair comparison, all methods use exactly the same training and testing strategies as described in Sec. 4.2. It is important to note that SKD, IFVD, and CWD use the adversarial distillation loss [13] on logits to improve performance, while our method does not. Note that the results of PAD are not reported in this section, and a detailed comparison between LAD, CAD and PAD is provided in Sec. 4.4.

Tab. 1 shows the results on various student models with different backbones and decoders (PPM [33] and ASPP [2]). Our method significantly improves the per-

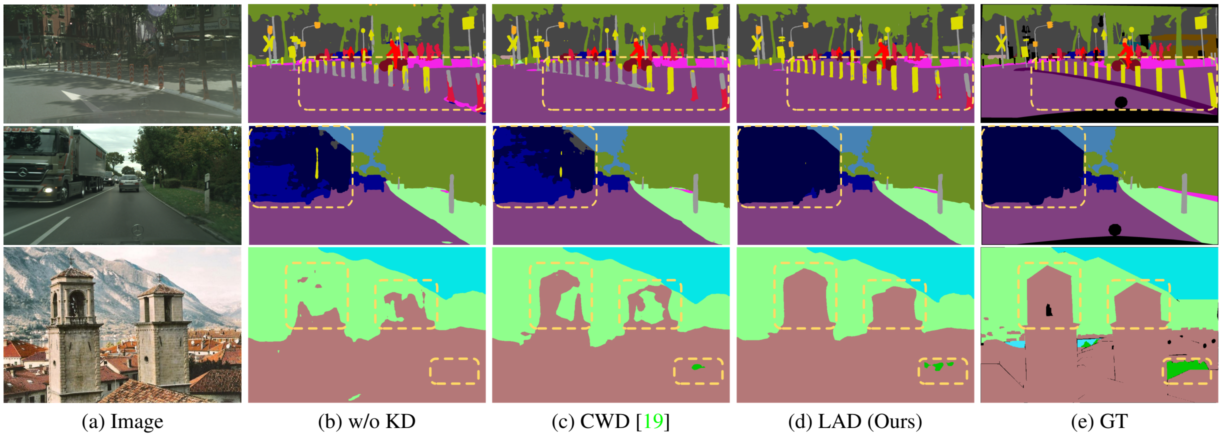


Figure 4. Qualitative results of semantic segmentation. The student is PSPNet-R18, and the teacher is PSPNet-R101. The first two rows present results on the Cityscapes validation set, while the last row shows results on the ADE20K validation set.

formance of baseline students without KD. For example, the performance gains for PSPNet-R18 under LAD are 4.21%, 4.39%, and 4.60% on Cityscapes, Pascal VOC, and ADE20K, respectively. Although we utilize the features from the last layer of the backbone for distillation by default, the performance gains are not much affected by the backbone architecture.

More importantly, our method consistently outperforms other methods by a large margin under various experimental setups, especially on Pascal VOC and ADE20K. For example, LAD outperforms CWD [19], the previous state-of-the-art KD method for semantic segmentation, by 2.85%, 3.66%, 2.48%, and 3.17% when using PSPNet-R18, PSPNet-MV2, DeepLabV3-R18, and DeepLabV3-MV2 as the student on challenging ADE20K. The comparison on Cityscapes and ADE20K test sets is shown in Tab. 2. Our method still shows a significant performance advantage compared to existing methods.

CIRKD [25] adopts DeepLabV3-R101 as the teacher instead of PSPNet-R101. In addition, the experimental setup of CIRKD [25] has some differences from ours described in Sec. 4.2. For example, they use an initial learning rate of 0.02 and the total training iterations of 40k. For a fair comparison with CIRKD [25], we implemented our method based on their released code and obtained the results of our method exactly following their experimental setup. As shown in Tabs. 3 and 4, our method significantly outperforms the CIRKD [25] on all datasets.

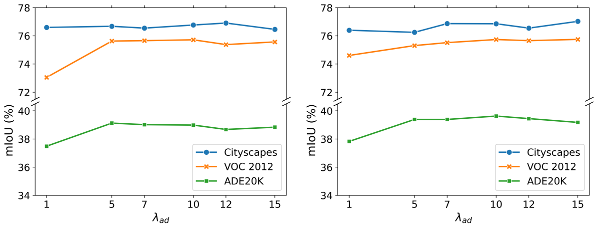
4.4. Ablation study

In this section, we give extensive experiments to investigate the effectiveness and characteristics of our method and discuss the choice of some hyper-parameters.

Weight of distillation loss. The feature distillation loss in our method is weighted by λ_{ad} in Eq. (9). We conduct extensive experiments to investigate the sensitivity of our method to λ_{ad}. As shown in Fig. 5, both CAD and LAD exhibit excellent robustness to the loss weight.

Position of distillation. We conduct experiments using features from the last layer of the backbone, the last layer of the decoder, and the final prediction layer. Note that the optimal loss weight may vary for different distillation positions, but we use the same loss weight for all distillation positions for simplicity. Here we also give the results of the naive feature distillation to serve as a baseline, with the same loss weight as our method. From Tab. 5 we can observe that 1) our method works best at the backbone, then at the decoder, and worst at the prediction layer, 2) LAD performs the best among our methods at the backbone, with slightly better performance than CAD, 3) CAD exhibits the best robustness at different distillation positions, and 4) PAD performs the worst among our methods but is still better than the naive feature distillation in general. The performance gap between LAD, CAD, and PAD indicates that the dimension of angular distillation is critical. We believe that the lack of spatial context is the reason for the poor results of PAD. As shown in Fig. 6, PAD fails to make the student learn a spatially similar pattern of feature activations as the teacher. PAD does not take into account the spatial context because the angular information of the features at different spatial locations is calculated and distilled independently. In contrast, both LAD and CAD involve spatial context.

Generalization over different networks. Following the previous methods [13, 19, 22], the above experiments are mainly conducted on segmentation models with a plain encoder-decoder architecture like PSPNet without skip connections. In this section, we conduct experiments based on



(a) CAD

(b) LAD

Figure 5. Ablation study about the sensitivity of our method to the loss weight λ_{ad}. The teacher is PSPNet-R101, and the student is PSPNet-R18.

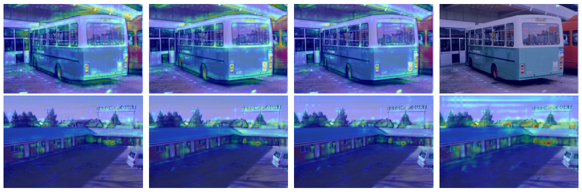
Method	val mIoU (%)		
	Cityscapes	VOC	ADE20K
Naive-backbone	74.50	71.98	35.36
PAD-backbone	75.52	74.81	39.36
CAD-backbone	76.77	75.72	38.99
LAD-backbone	**76.86**	**75.74**	**39.63**
Naive-decoder	74.97	70.73	35.98
PAD-decoder	75.02	71.26	35.43
CAD-decoder	**75.44**	**73.67**	**38.60**
LAD-decoder	75.27	71.39	36.62
Naive-logits	73.73	63.84	29.68
PAD-logits	**75.25**	69.43	31.18
CAD-logits	75.19	**71.40**	**37.92**
LAD-logits	74.96	70.11	31.18

Table 5. Ablation study about the distillation positions. The teacher is PSPNet-R101, and the student is PSPNet-R18. "Naive" denote naive feature distillation. "-backbone", "-decoder" and "-logits" denote distillation positions. The **best**/second best results are marked in bold/underline.

UPerNet [23], which adopts FPN [12] to fuse multi-level features in an inherent and pyramidal hierarchy. In addition, we use the Transformer backbone for the teacher and student, which has a completely different architecture from the CNN. Specifically, the teacher's backbone is Swin-B [14], while the student's backbone is Swin-T [14]. As shown in Tab. 6, our method greatly improves the performance of the baseline student without KD and outperforms CWD [19]. The results verify the effectiveness of our method again and further demonstrate the promising generality of our method over different networks.

4.5. Experiments on object detection

We also apply our method to object detection. To make a fair comparison with recent methods, we use the same experimental setup as in [27]. As shown in Tab. 7, our method achieves competitive performance compared to state-of-the-art methods. This indicates a promising generality of our method. We provide additional results on object detection in the supplementary material.



(a) Teacher (b) LAD (c) CAD (d) PAD

Figure 6. Heatmaps of feature activations from the teacher (PSPNet-R101) and the student (PSPNet-R18).

Method	val mIoU (%)	
	Cityscapes	ADE20K
T: UPerNet-SwinB	81.17	47.99
S: UPerNet-SwinT	77.94	43.72
+ CWD [19]	79.38	45.08
+ LAD (Ours)	79.51	45.47
+ CAD (Ours)	**79.54**	**46.12**

Table 6. Ablation study about the generalization of our method over different networks on validation sets of Cityscapes and ADE20K. The **best**/second best results are marked in bold/underline.

Teacher	Student	mAP	AP$_S$	AP$_M$	AP$_L$
RetinaNet -X101 (41.0)	RetinaNet-R50	37.4	20.6	40.7	49.7
	FKD [32]	39.6	22.7	43.3	52.5
	CWD [19]	40.8	22.7	44.5	55.3
	FGD [26]	40.7	22.9	45.0	54.7
	MGD [27]	41.0	23.4	45.3	55.7
	LAD (Ours)	41.0	23.3	45.2	55.1

Table 7. Comparison with state-of-the-art methods on COCO validation set. "X101" denotes ResNeXt101.

5. Conclusion

In this paper, we revisit the naive feature distillation method proposed in FitNets [17] with the aim of proposing a simple and effective feature distillation method. With the analysis of the loss function of FitNets [17] and well-designed experiments, we show that the sensitivity of this naive method to hyper-parameters is due to the fact that the weight of the angular difference term is affected by the magnitude of the features. Based on this, we propose three angular distillation methods for semantic segmentation. Experimental results show that our method achieves state-of-the-art performance and exhibits excellent robustness to hyper-parameters.

There is room for further exploration on how to utilize angular information of the feature for distillation. In addition, we focus on how to effectively perform feature distillation between manually assigned pairs of teacher-student intermediate layers, without considering the utilization of multi-layer features. We leave these for future work.

References

[1] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 7028–7036. AAAI Press, 2021. 2

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587 [cs]*, Dec. 2017. 5, 6

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV*, volume 11211 of *Lecture Notes in Computer Science*, pages 833–851. Springer, 2018. 1

[4] 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 *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 3213–3223. IEEE Computer Society, 2016. 5

[5] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. 5

[6] Bharath Hariharan, Pablo Arbelaez, Lubomir D. Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In Dimitris N. Metaxas, Long Quan, Alberto Sanfeliu, and Luc Van Gool, editors, *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 991–998. IEEE Computer Society, 2011. 5

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. 1, 5

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531 [cs, stat]*, Mar. 2015. 1, 5

[9] Deyi Ji, Haoran Wang, Mingyuan Tao, Jianqiang Huang, Xian-Sheng Hua, and Hongtao Lu. Structural and statistical texture knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16876–16885, 2022. 1

[10] Mingi Ji, Byeongho Heo, and Sungrae Park. Show, attend and distill: Knowledge distillation via attention-based feature matching. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 7945–7952. AAAI Press, 2021. 2

[11] Xiao Jin, Baoyun Peng, Yichao Wu, Yu Liu, Jiaheng Liu, Ding Liang, Junjie Yan, and Xiaolin Hu. Knowledge distillation via route constrained optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1345–1354, 2019. 2

[12] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 8

[13] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured knowledge distillation for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 2604–2613. Computer Vision Foundation / IEEE, 2019. 1, 2, 3, 5, 6, 7

[14] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 8

[15] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019. 2

[16] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XI*, volume 11215 of *Lecture Notes in Computer Science*, pages 283–299. Springer, 2018. 2

[17] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 1, 2, 3, 8

[18] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4510–4520. IEEE Computer Society, 2018. 5

[19] Changyong Shu, Yifan Liu, Jianfei Gao, Zheng Yan, and Chunhua Shen. Channel-wise knowledge distillation for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5311–5320, Oct. 2021. 2, 5, 6, 7, 8

[20] Guo-Hua Wang, Yifan Ge, and Jianxin Wu. Distilling knowledge by mimicking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8183–8195, 2022. 2

[21] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui

Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(10):3349–3364, 2021. 1

[22] Yukang Wang, Wei Zhou, Tao Jiang, Xiang Bai, and Yongchao Xu. Intra-class feature variation distillation for semantic segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, Lecture Notes in Computer Science, pages 346–362, Cham, 2020. Springer International Publishing. 2, 3, 5, 6, 7

[23] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, Lecture Notes in Computer Science, pages 432–448, Cham, 2018. Springer International Publishing. 8

[24] Jiafeng Xie, Bing Shuai, Jianfang Hu, Jingyang Lin, and Wei-Shi Zheng. Improving fast segmentation with teacher-student learning. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, page 205. BMVA Press, 2018. 2

[25] Chuanguang Yang, Helong Zhou, Zhulin An, Xue Jiang, Yongjun Xu, and Qian Zhang. Cross-image relational knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12319–12328, 2022. 5, 6, 7

[26] Zhendong Yang, Zhe Li, Xiaohu Jiang, Yuan Gong, Zehuan Yuan, Danpei Zhao, and Chun Yuan. Focal and global knowledge distillation for detectors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 4633–4642. IEEE, 2022. 8

[27] Zhendong Yang, Zhe Li, Mingqi Shao, Dachuan Shi, Zehuan Yuan, and Chun Yuan. Masked generative distillation. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, Lecture Notes in Computer Science, pages 53–69, Cham, 2022. Springer Nature Switzerland. 5, 6, 8

[28] Junho Yim, Donggyu Joo, Ji-Hoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 7130–7138. IEEE Computer Society, 2017. 1

[29] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VI*, volume 12351 of *Lecture Notes in Computer Science*, pages 173–190. Springer, 2020. 1

[30] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 1, 2

[31] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Mueller, R. Manmatha, Mu Li, and Alexander Smola. Resnest: Split-attention networks. *arXiv:2004.08955 [cs]*, Apr. 2020. 1

[32] Linfeng Zhang and Kaisheng Ma. Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 8

[33] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6230–6239. IEEE Computer Society, 2017. 1, 5, 6

[34] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5122–5130, July 2017. 5

[35] Yichen Zhu and Yi Wang. Student customized knowledge distillation: Bridging the gap between student and teacher. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5057–5066, 2021. 2