Multi-Knowledge Aggregation and Transfer for Semantic Segmentation

Yuang Liu, Wei Zhang*, Jun Wang*

East China Normal University
Shanghai, China
{frankliu624, zhangwei.thu2011, wongjun}@gmail.com

Abstract

As a popular deep neural networks (DNN) compression technique, knowledge distillation (KD) has attracted increasing attentions recently. Existing KD methods usually utilize one kind of knowledge in an intermediate layer of DNN for classification tasks to transfer useful information from cumbersome teacher networks to compact student networks. However, this paradigm is not very suitable for semantic segmentation, a comprehensive vision task based on both pixel-level and contextual information, since it cannot provide rich information for distillation. In this paper, we propose a novel multi-knowledge aggregation and transfer (MKAT) framework to comprehensively distill knowledge within an intermediate layer for semantic segmentation. Specifically, the proposed framework consists of three parts: Independent Transformers and Encoders module (ITE), Auxiliary Prediction Branch (APB), and Mutual Label Calibration (MLC) mechanism, which can take advantage of abundant knowledge from intermediate features. To demonstrate the effectiveness of our proposed approach, we conduct extensive experiments on three segmentation datasets: Pascal VOC, Cityscapes, and CamVid, showing that MKAT outperforms the other KD methods.

Introduction

Semantic segmentation is a crucial and challenging vision task, which aims to assign a semantic category to each pixel in an image. With the spectacular advances of deep neural networks, semantic segmentation has achieved significant progress and shown great potential in many practical applications, such as autonomous driving (Levinson et al. 2011; Huang et al. 2018), scene understanding (Xiao, Sigal, and Jae Lee 2017; Sigurdsson et al. 2020), and image editing (Morel, Petro, and Sbert 2012). The DNN-based segmentation methods, *e.g.*, FCN (Long, Shelhamer, and Darrell 2015), SegNet (Badrinarayanan, Kendall, and Cipolla 2017), RefineNet (Lin et al. 2017), DeepLab (Chen et al. 2015, 2017a,b), and PSPNet (Zhao et al. 2017), have achieved remarkable performance but bear up the problems of cumbersome architectures and expensive computation. This prevents the segmentation networks from running on mobile or edge devices, limiting the development of numerous practical applications in the mobile internet era. To tackle the problems, the workflow of compressing a cumbersome network to a compact network is often leveraged.

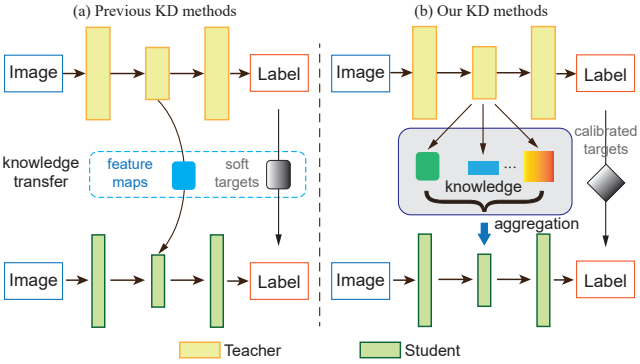


Figure 1: Comparison of the (a) conventional KD methods and (b) our MKAT framework.

In this paper, we investigate knowledge distillation (Hinton, Vinyals, and Dean 2015), a popular model compression way in classification tasks (Deng et al. 2009) through transferring knowledge from a cumbersome teacher network to a compact student network. Thanks to the simplicity of classification, which aims to distinguish the classes through image-level features, existing KD methods (Zagoruyko and Komodakis 2017; Yim et al. 2017; Park et al. 2019; Peng et al. 2019; Chung et al. 2020) could efficiently train the compact student network depending on certain knowledge. Numerous distillation methods have been developed and various kinds of knowledge are available. The selection of distillation strategies and knowledge is usually empirical, and nobody could guarantee which strategy is the best for a specific task without experiments (Ji and Zhu 2020; Menon et al. 2020). Different from classification, semantic segmentation is a high-level and comprehensive vision task, in which not only the extracted features but also the contextual relationships parameterized by the networks are critical for pixel prediction in a full image. Hence, the student learning a single type of knowledge from a teacher network's layer is far from enough in segmentation task. Although there are tiny methods distilling knowledge from multiple teachers to supervise the student in classification (You et al. 2017; Zhu, Gong et al. 2018; Liu, Zhang, and Wang 2020), they also only consider a single knowledge

*Corresponding authors.

form, mainly the soft labels. Besides, the multi-teacher KD methods are very limited due to the dilemma of information combination, and more cumbersome teacher networks cause more computation and storage cost. It's even impossible to deploy multi-teacher KD methods for segmentation, within which the DNNs are expensive to perform inference. To better explain the necessity of knowledge aggregation for segmentation distillation learning, we introduce an example about the physics curriculum learning. As we all know, the physics curriculum mainly consists of optics, thermodynamics, electricity, magnetism, mechanics, *etc*. If a student only learns the knowledge of optics from one or more excellent physics teachers, he still cannot solve the comprehensive electromagnetic problems in an exam. So an outstanding student is supposed to learn as comprehensive knowledge as possible, analogously in distillation learning.

To cope with the above issues, we carefully design a general multi-knowledge aggregation and transfer (MKAT) framework to leverage multiple types of heterogeneous knowledge distilled from the teacher backbone to boost the student comprehensively. Due to different kinds of knowledge may be in various shapes and feature spaces, *e.g*., gram matrix (Yim et al. 2017) and correlation matrix (Peng et al. 2019), we develop independent transformers and encoders (ITE) modules both for the teacher and student to transform these heterogeneous knowledge into a consist shape. And a siamese auxiliary prediction branch (APB) is introduced to agglomerate the comprehensive knowledge and reconstruct the semantic features with the supervision of the teacher network. Moreover, APB creates a proxy online learning (POL) environment to improve the student with less computation, making the student learn from the teacher step-by-step and mitigating the big gap between two models. In addition, to leverage the knowledge from the soft labels generated by the teacher network and avoid conflicts with intermediate knowledge, we present a mutual label calibration (MLC) mechanism to assist the learning of the student and APB. We conduct a detailed ablation study to verify the effectiveness of each component in our MKAT framework.

The overall contributions of this paper are summarized as follows:

- We propose a novel multi-knowledge aggregation and transfer (MKAT) framework for semantic segmentation. To our best knowledge, it represents the first effort to exploit multiple knowledge from one intermediate layer for distillation.
- Two independent transformers and encoders (ITE) modules and a siamese auxiliary prediction branch (APB) are developed to distill and aggregate the heterogeneous knowledge.
- Based on APB, we introduce the proxy online learning (POL) and mutual label calibration (MCL) mechanisms to boost the performance of student networks.
- Extensive experiments are conducted on three image segmentation datasets and our proposed approach outperforms the state-of-the-art methods.

Related Work

Semantic Segmentation. Semantic segmentation is a foundational but challenging vision task, and has achieved remarkable results thanks to the rapid development of fully convolutional networks. Various works such as FCN (Long, Shelhamer, and Darrell 2015), DeepLab (Chen et al. 2015, 2017a,b) and PSPNet (Zhao et al. 2017), always exploit sophisticated backbone networks (*e.g*., ResNet (He et al. 2016), DenseNet (Huang et al. 2017)) to learn discriminative feature representations for dense prediction. And exploiting multi-scale context also benefits segmentation, for example, atrous convolution (Chen et al. 2015, 2017a,b), pyramid pooling module (Zhao et al. 2017), context encoding (Yu et al. 2020), *etc*, has been developed in recent years. However, these approaches always involve unbearable storage or expensive computation for mobile applications.

Meanwhile, designing highly efficient segmentation networks attracts much attention from the community. In addition to adopting some lightweight feature extraction networks (*e.g*., MobileNet (Sandler et al. 2018), ShuffleNet (Ma et al. 2018)), most works pay attention to explore efficient convolutional segmentation architectures (*e.g*., ENet (Paszke et al. 2016), ERFNet (Romera et al. 2017), ESNet (Lyu et al. 2019)). With the proliferation of mobile applications, the demand for efficient segmentation increases, and knowledge distillation is a potential way.

Knowledge Distillation. As a popular model compression (Ba and Caruana 2014) paradigm, knowledge distillation (KD) (Hinton, Vinyals, and Dean 2015; Wang and Yoon 2020) has made significant progress in visual classification, mainly through distilling probability soft targets (Park et al. 2019; Sarfraz, Arani, and Zonooz 2019; Xie et al. 2020; Zhang et al. 2020) or intermediate features (Zagoruyko and Komodakis 2017; Huang and Wang 2017; Tung and Mori 2019; Heo et al. 2019). More and more applications are arising, including object detection (Dai et al. 2021; Guo et al. 2021), person re-identification (Wu et al. 2019; Porrello, Bergamini, and Calderara 2020), pose estimation (Weinzaepfel et al. 2020; Zheng et al. 2021) and so on.

However, all the above KD methods only pay attention to single intermediate knowledge or just jointly exploit soft labels and intermediate feature maps. Though some researchers propose to distilling knowledge from multiple teachers (You et al. 2017; Liu, Zhang, and Wang 2020), they mainly collect one kind of knowledge, *i.e*., soft labels, and do not apply to complex tasks due to the huge computation of teachers.

There are tiny KD methods for segmentation (Liu et al. 2019; He et al. 2019), in which Liu *et al*. (Liu et al. 2019) presented two structured knowledge distillation schemes, pair-wise distillation and holistic distillation, while He *et al*. (He et al. 2019) optimized the feature similarity with adaptive auto-encoder to handle the inconsistency between the features of the student and teacher network. In fact, the pair-wise and pixel-wise knowledge introduced by Liu *et al*. (Liu et al. 2019) or the adaptation mechanism (He et al. 2019) can be regarded as a special form under our MKAT framework. They notice structured information, but ignore the diversity of knowledge in feature maps, which is critical for dense prediction tasks like semantic segmentation.

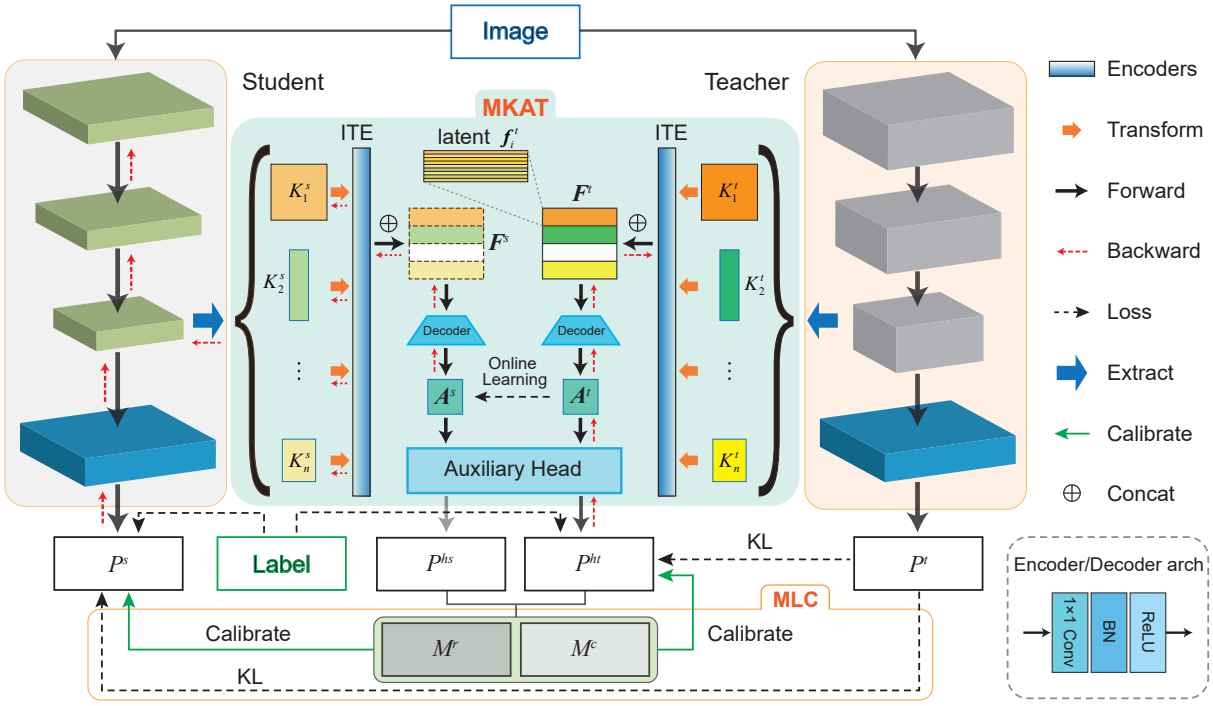


Figure 2: Overview of our proposed framework.

Proposed Approach

Preliminaries and Motivation

Semantic Segmentation is a densely pixel-level prediction task for each RGB image $X \in \mathbb{R}^{3\times H\times W}$ with C categories, and the soft targets distillation loss for segmentation can be formulated as:

$$\mathcal{L}_{kl}(P^s, P^t) = \sum_{i}^{HW} D_{KL}\left(\texttt{softmax}(P_i^s/\tau)||\texttt{softmax}(P_i^t/\tau)\right), \tag{1}$$

where $P^s, P^t \in \mathbb{R}^{C\times H\times W}$ are the prediction maps output by student $\mathcal{S}$ and teacher $\mathcal{T}$ respectively, D_{KL} is the Kullback-Leibler (KL) Divergence, and τ is the distillation temperature. P^s and P^t are given as: $P^s = \mathcal{S}(X;\boldsymbol{\theta}^s)$, $P^t = \mathcal{T}(X;\boldsymbol{\theta}^t)$, where $\boldsymbol{\theta}^s$ and $\boldsymbol{\theta}^t$ are the parameters of the student and teacher networks. Usually, only $\boldsymbol{\theta}^s$ is learnable while $\boldsymbol{\theta}^t$ is fixed in distillation learning.

Meanwhile, the intermediate features based distillation aims to transfer the high-level information extracted from a teacher's layers to a student. Various intermediate knowledge forms are available as summarized in Table 1, and they are with different shapes and representation spaces. Moreover, different knowledge forms express information in different perspectives. For instance, the similarity matrix (Tung and Mori 2019) measures the similarity among pixels in the image, Gram matrix (Yim et al. 2017) establishes the association between feature channels and conveys the style of the image, and channel and spatial attention maps represent the long-range dependence among pixels on the channel and space respectively. To take advantage of various useful information for segmentation, we propose the MKAT framework as shown in Figure 2.

Method	Knowledge Form	Shape
AT (Zagoruyko and Komodakis 2017)	Feature Norm	$H' \times W'$
AFD (Wang et al. 2019)	Feature Norm	$C' \times H' \times W'$
SP (Tung and Mori 2019)	Similarity Matrix	$C' \times H'(W') \times H'(W')$
FSP (Yim et al. 2017)	Gram Matrix	$C' \times C'$
CAD (Liu, Zhang, and Wang 2021)	Channel Attention	$C' \times C'$
SAD (Liu, Zhang, and Wang 2021)	Spatial Attention	$H'W' \times H'W'$

Table 1: Six kinds of knowledge adopted in our framework. C',H' and W' are the channel, width and height of the feature maps, respectively.

Knowledge Aggregation

We denote the knowledge set distilled from the feature maps output by the backbone of the teacher segmentation network as $\boldsymbol{\Omega}^t = \{K_1^t, K_2^t, \cdots, K_n^t\}$, while $\boldsymbol{\Omega}^s = \{K_1^s, K_2^s, \cdots, K_n^s\}$ is from the student network. n is the number of the employed knowledge, and different knowledge forms are in different sizes and representation spaces as listed in Table 1. To transfer them to a unified feature space and preserve the specific information, we propose the independent transformers and encoders (ITE) module for a teacher and a student respectively. It is noteworthy that the teacher and student backbones with different architectures (*e.g.*, ResNet (He et al. 2016) and MobileNet (Sandler et al. 2018)) may output feature maps in inconsistent sizes, which causes the inconsistency of aggregating knowledge between the teacher and student, and an adaptive average pooling layer can optionally be employed following the teacher's backbone before ITE. In addition, the inconsistent channels of the feature maps can be ignored, because our ITE will transform the

distilled knowledge by channel dimension.

Firstly, we define a series of transform operations $\{\texttt{Trans}_i(\cdot)|i \in \{1, 2, \cdots, n\}\}$ to transform all kinds of knowledge maps in $\mathbf{\Omega}^t$ and $\mathbf{\Omega}^s$ to the same width and height, *i.e.*, H' and W', which is written as the following:

$$\hat{K}_i^t = \texttt{Trans}_i(K_i^t)\,,\ \hat{K}_i^s = \texttt{Trans}_i(K_i^s)\,, \tag{2}$$

in which $\hat{K}_i^t$ and $\hat{K}_i^s$ are the transformed knowledge matrices from teacher and student respectively. Each $\texttt{Trans}_i(\cdot)$ works for a kind of knowledge K_i, and if the K_i has been in the target size, $\texttt{Trans}_i(\cdot)$ will do nothing, such as feature normalization maps (Zagoruyko and Komodakis 2017; Wang et al. 2019). And other transform operations mainly work through multiplying the extracted knowledge matrix K_i and the reshaped original feature maps following (Liu, Zhang, and Wang 2021). Specifically, for the similarity matrix (SP) (Tung and Mori 2019), which could be in two different shapes, the $\texttt{Trans}_i(\cdot)$ works by adding the two results of the above multiplications.

Then, each transformed knowledge matrix $\hat{K}_i$ will be projected into a latent space $\boldsymbol{f}_i$ by the encoder $\mathcal{E}_i(\hat{K}_i; \boldsymbol{\theta}_i)$ with learnable parameters $\boldsymbol{\theta}_i$. In order to preserve information, the encoders only involve spatial transformation through a projection layer. We denote the encoders on the side of teacher or student as

$$\begin{aligned}\boldsymbol{f}_i^t &= \mathcal{E}_i^t(\hat{K}_i^t; \boldsymbol{\theta}_i^{te}) = \texttt{relu}(\texttt{norm}(\texttt{conv}_{1\times1}(\hat{K}_i^t; \boldsymbol{\theta}_i^{te})))\,,\\ \boldsymbol{f}_i^s &= \mathcal{E}_i^s(\hat{K}_i^s; \boldsymbol{\theta}_i^{se}) = \texttt{relu}(\texttt{norm}(\texttt{conv}_{1\times1}(\hat{K}_i^s; \boldsymbol{\theta}_i^{se})))\,,\end{aligned} \tag{3}$$

in which $\boldsymbol{f}_i^t, \boldsymbol{f}_i^s$ are the latent knowledge output by encoder $\mathcal{E}_i^t, \mathcal{E}_i^s$ with learnable parameters $\boldsymbol{\theta}^{te}, \boldsymbol{\theta}^{se}$, respectively. The encoders all only consist of a 1×1 convolution layer, followed by a batch normalization layer and ReLU function.

At last, we collect and stack the latent knowledge $\boldsymbol{f}_i^t, \boldsymbol{f}_i^s$ from the teacher and student respectively, obtaining two comprehensive knowledge matrices $\boldsymbol{F}^t$ and $\boldsymbol{F}_i^s$:

$$\boldsymbol{F}^t = [\boldsymbol{f}_1^t, \boldsymbol{f}_2^t, \cdots, \boldsymbol{f}_n^t]\,,\ \boldsymbol{F}^s = [\boldsymbol{f}_1^s, \boldsymbol{f}_2^s, \cdots, \boldsymbol{f}_n^s]\,. \tag{4}$$

But in this situation, there is lack of learning objective for the ITE modules and the stacked knowledge needs to be associated and amalgamated. To cope with these challenges, we present an elegant solution by introducing an auxiliary prediction branch consisting of two decoders and a shared auxiliary head.

Auxiliary Prediction Branch

Although comprehensive knowledge matrices are distilled and transformed by ITEs, each latent knowledge is independent. To aggregate the stacked latent knowledge, we conduct two unified decoders following the ITEs respectively, which are similar to the encoders in architectures. Through a 1×1 convolution layer, the decoder can project the comprehensive knowledge to a unified feature space, achieving knowledge aggregation. We denote the two decoders as:

$$\begin{aligned}\boldsymbol{A}^t &= \mathcal{D}^t(\boldsymbol{F}^t; \boldsymbol{\theta}^{td}) = \texttt{relu}(\texttt{norm}(\texttt{conv}_{1\times1}(\boldsymbol{F}^t; \boldsymbol{\theta}^{td})))\,,\\ \boldsymbol{A}^s &= \mathcal{D}^s(\boldsymbol{F}^s; \boldsymbol{\theta}^{sd}) = \texttt{relu}(\texttt{norm}(\texttt{conv}_{1\times1}(\boldsymbol{F}^s; \boldsymbol{\theta}^{sd})))\,,\end{aligned} \tag{5}$$

in which $\boldsymbol{A}^t, \boldsymbol{A}^s$ are the aggregate knowledge matrices output by decoders $\mathcal{D}^t, \mathcal{D}^s$ with learnable parameters $\boldsymbol{\theta}^{td}, \boldsymbol{\theta}^{sd}$ respectively. It is worth mentioning that the parameters of the encoders and decoders are very small, and only tiny amount of computation is added compared with the segmentation network.

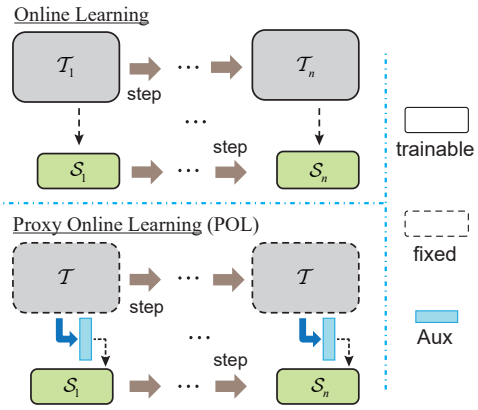


Figure 3: Vanilla online and our POL mechanism.

With these insights, we can establish the distillation loss based on knowledge aggregation:

$$\mathcal{L}_{ka}(\boldsymbol{A}^s, \boldsymbol{A}^t) = \left|\left|\boldsymbol{A}^s - \boldsymbol{A}^t\right|\right|_1\,. \tag{6}$$

The loss function for student network is given as follows,

$$\mathcal{L}^S = \mathcal{L}_{ce}(P^s, Y) + \alpha\mathcal{L}_{ka}(\boldsymbol{A}^s, \boldsymbol{A}^t)\,, \tag{7}$$

where $\mathcal{L}_{ce}$ is the cross entropy loss, α is the coefficient.

Now, we have addressed the knowledge aggregation challenge, but a new issue has arisen — the learnable encoders and decoders in the teacher side need a training objective to preserve and aggregate different knowledge. Hence, we employ a fully-convolutional auxiliary head $\mathcal{H}$ following the decoders, which usually works as classifier on the top of segmentation network, such as PSPNet (Zhao et al. 2017), DeepLab (Chen et al. 2015, 2017a,b). In this way, the comprehensive knowledge is involved in the segmentation task through the auxiliary prediction branch (APB) and all the ancillary components could update via gradient descent. The prediction map P^{ht} from the teacher side output by APB is denoted as $P^{ht} = \mathcal{H}(\boldsymbol{A}^t; \boldsymbol{\theta}^h)$, where $\boldsymbol{A}^t$ is the aggregate knowledge matrix from the teacher and $\boldsymbol{\theta}^h$ is the parameters of the auxiliary head. APB learns from not only the ground truth, but also the soft labels from the teacher network. The loss function of APB or auxiliary head is given as follows:

$$\mathcal{L}^H = \mathcal{L}_{ce}(P^{ht}, Y) + \beta\mathcal{L}_{kl}(P^{ht}, P^t)\,, \tag{8}$$

in which β is a hyperparameter for balancing the cross entropy and KL loss.

With the assistance of APB, all kinds of the distilled knowledge are transformed and aggregated for a common objective, making the knowledge aggregation and transfer more reliable. Moreover, the lightweight APB and ITE on the teacher side bring an additional benefit for the student network learning — it creates a proxy online learning (POL) environment with lightweight components for the student network, shown in Figure 3. Online learning (Zhao et al. 2020; Yang et al. 2019;

Xie et al. 2019) is an efficient distillation strategy by guiding the student step-by-step with less gaps. However, the vanilla online learning methods (Zhao et al. 2020; Yang et al. 2019; Xie et al. 2019) require the cumbersome teacher network updating parameters synchronously, the Achilles' heel of which is the expensive memory and computation. Our POL mechanism could guide the student with $\mathcal{L}_{ka}$ step-by-step via only updating the lightweight components.

Mutual Label Calibration

Following previous distillation methods (Ahn et al. 2019; Liu et al. 2019; Chung et al. 2020), we also attempt to enhance the performance by leveraging soft labels information except intermediate features. However, the offline soft targets learning is inconsistent with the online knowledge aggregation learning and degrades the performance unexpectedly, as shown in Table 5. Specifically, the student is supposed to learn from the gradually updated and aggregated knowledge (or in POL), while the additional soft labels are ultimate and fixed (or for offline learning). To tackle the problem, we propose a mutual label calibration (MLC) algorithm, adopting the extra prediction map P^{hs} output by the siamese auxiliary head forward on the student side.

$$P^{hs} = \mathcal{H}(\boldsymbol{A}^s; \boldsymbol{\theta}^h)\,, \tag{9}$$

where $\boldsymbol{A}^s$ is the aggregate knowledge from the student output by $\mathcal{D}^s$. To separate the pixels predicted correctly guided by the aggregate knowledge, a calibration mask map M^c is computed by:

$$M_i^c = \begin{cases} 1\,, \text{ if } \phi(P_j^{hs}) = \phi(P_j^{ht}) \text{ and } \phi(P_j^{ht}) = Y_j \\ 0\,, \text{ otherwise}\,, \end{cases} \tag{10}$$

where $j \in \{1, 2, \cdots, HW\}$ is the pixel index, Y is the ground truth and $\phi(\cdot)$ is the label generation function with the prediction maps output by segmentation networks, *e.g.*, `argmax`. Meanwhile, the complementary map of M^c is $M^r = \mathbf{1} - M^c$. For the student, it's supposed to learn from the soft labels P^t where the aggregate knowledge cannot cover. Hence, the loss function of the student is rewritten as:

$$\begin{aligned} \mathcal{L}^S = &\mathcal{L}_{ce}(P^s, Y) + \alpha\mathcal{L}_{ka}(\boldsymbol{A}^s, \boldsymbol{A}^t) \\ &+ \beta\mathcal{L}_{kl}(P^s \odot M^r \oplus M^c, P^t \odot M^r \oplus M^c)\,, \end{aligned} \tag{11}$$

in which $\odot$ is the Hadamard product, and $\oplus$ is the matrix addition operation.

In addition, to promote the ITE and APB on the teacher side to pay attention to the pixel-level soft labels that the student ignores in soft labels learning, we reformulate the calibrated loss function of the auxiliary head as:

$$\mathcal{L}^H = \mathcal{L}_{ce}(P^{ht}, Y) + \beta\mathcal{L}_{kl}(P^{ht} \odot M^c \oplus M^r, P^t \odot M^c \oplus M^r)\,. \tag{12}$$

Training Pipeline

In distillation procedure, all the ancillary components, *i.e.*, ITEs and APB, and the student network update parameters synchronously. We update the ITE on the teacher side and APB by minimizing the loss $\mathcal{L}^H$, and the gradients will be backpropagated to $\boldsymbol{\theta}^{te}$ by the following formulas:

$$\begin{aligned} \frac{\partial\mathcal{L}^H}{\partial\boldsymbol{\theta}_i^{te}} = &\frac{\partial(\mathcal{L}^{ce} + \beta\mathcal{L}_{kl})}{\partial\mathcal{H}(\boldsymbol{A}^t; \boldsymbol{\theta}^h)} \frac{\partial\mathcal{H}(\boldsymbol{A}^t; \boldsymbol{\theta}^h)}{\partial\mathcal{D}^t(\boldsymbol{F}^t; \boldsymbol{\theta}^{td})} \frac{\partial\mathcal{D}^t(\boldsymbol{F}^t; \boldsymbol{\theta}^{td})}{\partial\mathcal{E}_i^t(\hat{K}_i^t; \boldsymbol{\theta}_i^{te})} \frac{\partial\mathcal{E}_i^t(\hat{K}_i^t; \boldsymbol{\theta}_i^{te})}{\boldsymbol{\theta}_i^{te}}, \\ &i \in \{1, 2, \cdots, n\}\,. \end{aligned} \tag{13}$$

The student is trained by minimizing $\mathcal{L}^S$, and the n encoders and the decoder on the student side are updated by:

$$\begin{aligned} \frac{\partial\mathcal{L}^S}{\partial\boldsymbol{\theta}^s} = &\frac{\partial\mathcal{L}_{ce}}{\partial\mathcal{S}(X; \boldsymbol{\theta}^s)} \frac{\partial\mathcal{S}(X; \boldsymbol{\theta}^s)}{\partial\boldsymbol{\theta}^s} + \beta\frac{\partial\mathcal{L}_{kl}}{\mathcal{S}(X; \boldsymbol{\theta}^s)} \frac{\partial\mathcal{S}(X; \boldsymbol{\theta}^s)}{\partial\boldsymbol{\theta}^s} \\ &+ \alpha\sum_i^n \frac{\partial\mathcal{L}_{ka}}{\partial\mathcal{D}^s(\boldsymbol{A}^s; \boldsymbol{\theta}^{sd})} \frac{\partial\mathcal{D}^s(\boldsymbol{A}^s; \boldsymbol{\theta}^{sd})}{\partial\mathcal{E}_i^s(\hat{K}_i^s; \boldsymbol{\theta}_i^{se})} \frac{\partial\mathcal{E}_i^s(\hat{K}_i^s; \boldsymbol{\theta}_i^{se})}{\partial\hat{K}_i^s} \frac{\partial\hat{K}_i^s}{\partial\boldsymbol{\theta}^s}\,. \end{aligned} \tag{14}$$

Note that the gradients of decoder $\mathcal{D}^t(\cdot, \boldsymbol{\theta}^{td})$ and $\mathcal{D}^t(\cdot, \boldsymbol{\theta}^{sd})$ have been calculated during backpropagating encoders (Eq. 13) and student (Eq. 14), so we omit them in this section.

Experiments

Datasets

Pascal VOC 2012. It (Everingham and Winn 2011) contains 20 foreground object classes and an extra background class. Following (Chen et al. 2017a; Zhao et al. 2017), we use the additional annotation provided by (Hariharan et al. 2011), resulting in 10,582 labeled images for training.

Cityscapes. Cityscapes (Cordts et al. 2016) is for urban scene understanding and contains 30 classes with only 19 classes used for evaluation. It contains 2,975 fine annotation images for training, 500 for validation, and 1,525 for testing.

CamVid. CamVid (Brostow et al. 2008) is an automotive dataset, containing 367 training and 233 testing images, each with 720×960 pixels. Methods are evaluated on the most frequent 11 classes.

Implementation Details

Training setup. Our approach is implemented by PyTorch. We employ DeelpLabV3 with ResNet101 as a teacher network on Cityscapes, while DeelpLabV3 with ResNet50 for VOC and CamVid. The student networks are default DeelpLabV3 architecture with compact backbones (*i.e.*, ResNet18 or MobileNet). All the models are trained alone or with different KD methods by mini-batch stochastic gradient descent (SGD) with the momentum (0.9) and the weight decay (0.0005) for 120 epochs. Following (Chen et al. 2017b), we employ a poly learning rate policy where the initial learning rate is multiplied by $(1 - \frac{iter}{total\_iters})^{0.9}$ after each iteration. And, the initial learning rate of the backbone and encoders is set to 0.007 which is 0.1 times that of the auxiliary head or classifier. The batch size is set to 8, but 4 when testing on Cityscapes due to the super resolution. For data augmentation, we apply random horizontal flipping and random cropping (crop-size 513/768/540 for VOC/Cityscapes/CamVid) during training. The hyperparameters $\{\tau, \beta\}$ are set to $\{6.0, 0.5\}$ for CamVid and $\{10.0, 0.5\}$ for VOC/Cityscapes, respectively.

Method	Type	mIoU(%)	#Params	#FLOPs
ResNet50	Teacher	75.80	39.64M	51.29G
MobileNet	Student	68.87		
KL	L	70.42		
AT	F	69.85		
Hint	F	69.00		
AFD	F	70.77		
SP	S	70.14		
VID	F,L	70.74		
FSP	S	69.59	5.11M	22.24G
SAD	S	70.21		
CAD	S	69.20		
mKD	F,S,L	68.98		
mKD3	F,S,L	69.75		
mKD+KL	F,S,L	68.87		
Ours	F,S,L	**71.86**		
ResNet18	Student	69.38		
KL	L	70.45		
MT3	L	71.54		
KA	F	71.11	15.90M	11.37G
SKD	F,L	71.73		
Ours	F,S,L	**72.42**		

Table 2: Comparison of different KD methods on VOC. The symbols 'F', 'S' and 'L' denote different types of knowledge, *i.e.*, intermediate **F**eature, **S**tructural relation and soft **L**abel.

And α is chosen from [10,15], default by 10. The channel of the latent knowledge is set to 256 in all the experiments.

Metrics. We employ mean Intersection over Union (mIoU) to measure the performance. Floating point operations (FLOPs) and parameters (Params) are adopted to measure the computation and storage cost of the segmentation networks.

Comparison with the State-of-the-Arts

Results on VOC. For the VOC dataset, we evaluate our approach on DeelpLabV3 with MobileNetV2 and ResNet18 backbones. The size of the two student networks are only about 13% and 30% of the teacher respectively while the computation are about 44% and 30% of the teacher. As shown in Table 2, we compare our proposed approach with various KD algorithms in different types, including intermediate feature based KD (**Hint** (Romero et al. 2015), **AT** (Zagoruyko and Komodakis 2017), **AFD** (Wang et al. 2019), **VID** (Ahn et al. 2019)), structural relation based KD (**FSP** (Yim et al. 2017), **SP** (Tung and Mori 2019)) and soft labels based KD (**KL** (Hinton, Vinyals, and Dean 2015)). In addition to the common KD methods, we also test the multi-teacher KD method (You et al. 2017) and methods specific to segmentation tasks (**KA** (He et al. 2019), **SKD** (Liu et al. 2019), **SAD** and **CAD** (Liu, Zhang, and Wang 2021)). '**MT3**' (You et al. 2017) is the multi-teacher KD method with three teachers with the same architecture. 'mKD' is implemented by simply combining different distillation losses (default six, similar to our MKAT) without knowledge aggregation, specifically, 'mKD3' indicates three distillation losses are employed. Our approach can boost the performance of the students by about 3%, and outperform existing KD methods. However, simply combining different distillation losses cannot take advantage of all the knowledge, and the additional soft targets loss (KL) makes the results worse.

Results on Cityscapes. Table 3 lists the detailed quantitative results of 4 architectures with different KD methods on Cityscapes. For ResNet18, the original student yields 67.7% mIoU, and our method amazingly boosts the performance to 71.34% (3.57% improved). The two real-time segmentation architectures are improved by about 2 points with our approach. More visualizations of results are presented in appendix. The multi-teacher method 'MT3' averages the soft labels of three teachers to enhance the knowledge, but it costs more computation and storage due to the employed cumbersome teacher networks. As for 'mKD' strategy, we find it's difficult to select and balance each distillation loss, and the student cannot capture the relationships among various knowledge and unify them.

		ResNet18	MobileNet	ENet	ERFNet
Method	Type	11.77M	5.11M	0.36M	2.07M
–	Student	67.77	71.61	60.14	67.68
KL	L	69.12	72.52	60.65	68.63
AT	F	69.54	72.40	60.67	68.19
Hint	F	68.97	71.98	60.91	68.71
AFD	F	68.95	72.26	61.36	68.95
SP	S	69.48	72.69	61.60	68.84
VID	F,L	69.58	72.90	61.77	69.96
FSP	S	68.84	72.35	60.86	68.71
SAD	S	69.17	72.71	61.88	69.08
CAD	S	68.54	72.54	61.49	68.83
MT3	L	69.49	72.83	61.38	68.82
KA	F	69.05	72.62	61.25	68.59
SKD	F,S,L	69.73	73.33	61.93	69.18
mKD	F,S,L	69.31	73.05	61.44	68.79
mKD+KL	F,S,L	69.14	72.85	61.16	68.60
mKD3	F,S,L	69.58	72.97	61.59	68.87
Ours	F,S,L	**71.34**	**73.98**	**62.77**	**69.63**

Table 3: Comparison of different KD methods on Cityscapes.

Results on CamVid. We further carry out experiments on CamVid to further verify the distillation ability of the proposed MKAT on real autonomous driving dataset, as shown in Table 4. We adopt two kinds of ResNet18-based segmentation architectures: PSPNet ('PSP-R18') (Zhao et al. 2017) and DeelpLabV3 ('DP-R18') (Chen et al. 2017b), both of which are improved by about 2 points. In particular, the MobileNet has only about 13% parameters of the teacher ResNet50, but achieves more than 97% performance of ResNet50. More visualizations of results on CamVid are available in appendix.

Ablation Study

We conduct three sets of ablation studies based on DeelpLabV3 (with ResNet18) on Cityscapes and PSPNet (with ResNet18) on CamVid.

Effectiveness of POL & MLC. We introduce the POL and MLC mechanisms to guide the student learning and calibrate features and labels knowledge learning. To verify the performance of them, we conduct experiments with different settings on three datasets. As shown in Table 5, compared with the online setup, the offline setup, which means adopting well-trained and fixed ITEs and APB, could reduce at least

Method	#Params	#FLOPs	mIoU (%)
ENet	0.36M	5.54G	58.24
ESNet	1.66M	32.02G	66.58
ResNet50(teacher)	39.64M	128.15G	67.18
MobileNet	5.11M	56.40G	63.74
MobileNet(ours)	5.11M	56.40G	**65.18**
DP-R18	15.90M	28.39G	60.30
DP-R18(ours)	15.90M	28.39G	**62.47**
PSP-R18	12.92M	26.04G	60.22
PSP-R18(ours)	12.92M	26.04G	**62.19**

Table 4: The performance on the CamVid val set.

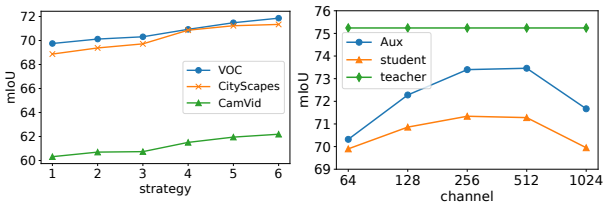
0.3% mIoU. And, the MLC, the part in the orange box in Figure 2, helps the student selectively learn from the teacher's soft labels and brings improvements of $0.3 \sim 0.6$ points. In addition, when adopting the whole pixel-level soft labels of each image from the teacher network ('w/ KL': replace MLC with KL loss), it will cause serious damage to the performance of the student, shown in the last column of the table. This verifies the necessity of the MLC mechanism for a student to take advantage of more comprehensive and consistent knowledge. The MLC takes KA as premise, so we abandon the ablation analysis of KA.

Dataset	online	offline	w/o MLC	w/ KL
Cityscapes	71.34	70.85	70.89	68.92
CamVid	62.19	61.90	61.82	60.12

Table 5: Ablation study of POL and MLC. The 'online' means updating the student and other ancillary modules (*i.e.*, ITE, APB) synchronously, while the 'offline' means complying distillation with pre-trained ancillary modules.

Different strategies of knowledge aggregation. In our framework, there are six kinds of knowledge adopted, and up to $\sum_{i=1}^{6} \binom{i}{6}$ combinations of them. Hence, it's impossible to test all the combination strategies. Under the premise of different amounts of knowledge, we randomly select 6 combinations for each strategy and average the test results, as shown in Figure 4a. We can see that as the amount of aggregate knowledge increases, the performance of our approach gradually improves on all three datasets. Meanwhile, when the number of selected knowledge forms is less than three, the performance level could be close to previous distillation methods. It is worth noting that even under the same knowledge amount setting, different choices may lead to big differences. Even so, our framework can make full use of the information by assembling as many kinds of knowledge as possible, avoiding manual selection and combination.

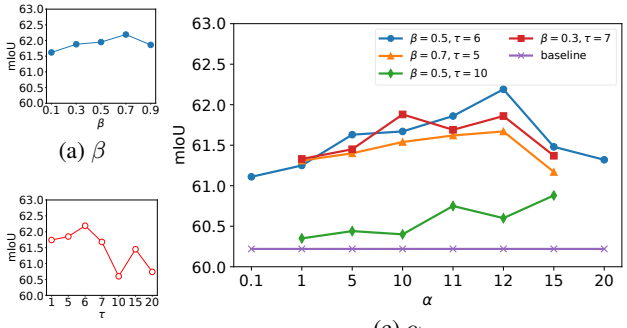
The sensitivity of hyperparameters. The channel of each latent knowledge directly affects the representation space, which in turn affects the distillation effect. To study the sensitivity of the hyperparameter m, we apply a hierarchy of grids of different channels $\{64, 128, 256, 512, 1024\}$. Experimental results of the auxiliary head and student network are shown in Figure 4b. Since the auxiliary head is trained directly based on the teacher's deep features, it has higher performance. Generally, the more channels of the latent knowledge, the stronger the ability to express various information, but too many channels could lead to insufficient distillation.



(a) aggregation strategy (b) latent channel

Figure 4: Training with different numbers of knowledge types and channels.

Built upon the baseline, an additional ablation study on other three hyperparameters α, β, and τ is shown in Figure 5. For vanilla knowledge distillation, β and τ are critical for soft labels learning. And in our approach, α is employed for balancing the intermediate learning. After a simple grid search optimizing for α and τ, we choose several groups of them, and adjust the α various from 0.1 to 20. Under a suitable setting of β and τ, $\alpha \in [10, 12]$ could stabilize the performance at $61.5\% \sim 62.0\%$ mIoU.



(a) β (b) τ (c) α

Figure 5: Impacts of different hyperparameters on CamVid.

Conclusion

In this paper, we present a novel multi-knowledge aggregation and transfer framework (MKAT) tailed for semantic segmentation. Different from the existing KD methods, MKAT explicitly distills multiple knowledge from a teacher's interminable layer to guide a student network at different perspectives. The proxy online learning (POL) and mutual label calibration (MLC) mechanisms both are the additional benefits brought by the auxiliary prediction branch, and can boost the student network without specific design. MKAT achieves comparable performance with state-of-the-art KD methods in several benchmarks. In the future, we could consider utilizing more kinds of knowledge and develop more efficient metric methods between the aggregate knowledge. Moreover, it's meaningful to extend the proposed approach to other high-level vision tasks, such as object detection (Guo et al. 2021) and pose estimation (Zheng et al. 2021).

Acknowledgement

This work was supported in part by the Fundamental Research Funds for the Central Universities, Key Laboratory of Advanced Theory and Application in Statistics and Data Science, Ministry of Education, and Project of 2021 Outstanding Doctoral Student Academic Innovation Plan ECNU under Grant No. YBNLTS2021-037.

References

Ahn, S.; Hu, S. X.; Damianou, A.; Lawrence, N. D.; and Dai, Z. 2019. Variational information distillation for knowledge transfer. In *CVPR*, 9163–9171.

Ba, J.; and Caruana, R. 2014. Do deep nets really need to be deep? *NeurIPS*, 27: 2654–2662.

Badrinarayanan, V.; Kendall, A.; and Cipolla, R. 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE TPAMI*, 39(12): 2481–2495.

Brostow, G. J.; Shotton, J.; Fauqueur, J.; and Cipolla, R. 2008. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 44–57.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2015. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2017a. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4): 834–848.

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H. 2017b. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Chung, I.; Park, S.; Kim, J.; and Kwak, N. 2020. Feature-map-level online adversarial knowledge distillation. In *ICML*.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 3213–3223.

Dai, X.; Jiang, Z.; Wu, Z.; Bao, Y.; Wang, Z.; Liu, S.; and Zhou, E. 2021. General Instance Distillation for Object Detection. In *CVPR*, 7842–7851.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255.

Everingham, M.; and Winn, J. 2011. The pascal visual object classes challenge 2012 (voc2012) development kit. *PASCAL*, 8.

Guo, J.; Han, K.; Wang, Y.; Wu, H.; Chen, X.; Xu, C.; and Xu, C. 2021. Distilling Object Detectors via Decoupled Features. In *CVPR*, 2154–2164.

Hariharan, B.; Arbeláez, P.; Bourdev, L.; Maji, S.; and Malik, J. 2011. Semantic contours from inverse detectors. In *ICCV*, 991–998.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

He, T.; Shen, C.; Tian, Z.; Gong, D.; Sun, C.; and Yan, Y. 2019. Knowledge adaptation for efficient semantic segmentation. In *CVPR*, 578–587.

Heo, B.; Kim, J.; Yun, S.; Park, H.; Kwak, N.; and Choi, J. Y. 2019. A comprehensive overhaul of feature distillation. In *ICCV*, 1921–1930.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *CVPR*, 4700–4708.

Huang, X.; Cheng, X.; Geng, Q.; Cao, B.; Zhou, D.; Wang, P.; Lin, Y.; and Yang, R. 2018. The apolloscape dataset for autonomous driving. In *CVPRW*, 954–960.

Huang, Z.; and Wang, N. 2017. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*.

Ji, G.; and Zhu, Z. 2020. Knowledge Distillation in Wide Neural Networks: Risk Bound, Data Efficiency and Imperfect Teacher. In *NeurIPS*.

Levinson, J.; Askeland, J.; Becker, J.; Dolson, J.; Held, D.; Kammel, S.; Kolter, J. Z.; Langer, D.; Pink, O.; Pratt, V.; et al. 2011. Towards fully autonomous driving: Systems and algorithms. In *IVS*, 163–168. IEEE.

Lin, G.; Milan, A.; Shen, C.; and Reid, I. 2017. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 1925–1934.

Liu, Y.; Chen, K.; Liu, C.; Qin, Z.; Luo, Z.; and Wang, J. 2019. Structured knowledge distillation for semantic segmentation. In *CVPR*, 2604–2613.

Liu, Y.; Zhang, W.; and Wang, J. 2020. Adaptive multi-teacher multi-level knowledge distillation. *Neurocomputing*, 415: 106–113.

Liu, Y.; Zhang, W.; and Wang, J. 2021. Source-free domain adaptation for semantic segmentation. In *CVPR*, 1215–1224.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*, 3431–3440.

Lyu, H.; Fu, H.; Hu, X.; and Liu, L. 2019. Esnet: Edge-based segmentation network for real-time semantic segmentation in traffic scenes. In *ICIP*, 1855–1859.

Ma, N.; Zhang, X.; Zheng, H.-T.; and Sun, J. 2018. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, 116–131.

Menon, A. K.; Rawat, A. S.; Reddi, S. J.; Kim, S.; and Kumar, S. 2020. Why distillation helps: a statistical perspective. *arXiv preprint arXiv:2005.10419*.

Morel, J.-M.; Petro, A. B.; and Sbert, C. 2012. Fourier implementation of Poisson image editing. *Pattern Recognition Letters*, 33(3): 342–348.

Park, W.; Kim, D.; Lu, Y.; and Cho, M. 2019. Relational knowledge distillation. In *CVPR*, 3967–3976.

Paszke, A.; Chaurasia, A.; Kim, S.; and Culurciello, E. 2016. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*.

Peng, B.; Jin, X.; Liu, J.; Li, D.; Wu, Y.; Liu, Y.; Zhou, S.; and Zhang, Z. 2019. Correlation congruence for knowledge distillation. In *ICCV*, 5007–5016.

Porrello, A.; Bergamini, L.; and Calderara, S. 2020. Robust Re-Identification by Multiple Views Knowledge Distillation. In *ECCV*, 93–110.

Romera, E.; Alvarez, J. M.; Bergasa, L. M.; and Arroyo, R. 2017. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE T-ITS*, 19(1): 263–272.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2015. Fitnets: Hints for thin deep nets. In *ICLR*.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 4510–4520.

Sarfraz, F.; Arani, E.; and Zonooz, B. 2019. Noisy Collaboration in Knowledge Distillation. In *ICLR*.

Sigurdsson, G. A.; Alayrac, J.-B.; Nematzadeh, A.; Smaira, L.; Malinowski, M.; Carreira, J.; Blunsom, P.; and Zisserman, A. 2020. Visual Grounding in Video for Unsupervised Word Translation. In *CVPR*, 10850–10859.

Tung, F.; and Mori, G. 2019. Similarity-preserving knowledge distillation. In *ICCV*, 1365–1374.

Wang, K.; Gao, X.; Zhao, Y.; Li, X.; Dou, D.; and Xu, C.-Z. 2019. Pay attention to features, transfer learn faster CNNs. In *ICLR*.

Wang, L.; and Yoon, K.-J. 2020. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *arXiv preprint arXiv:2004.05937*.

Weinzaepfel, P.; Brégier, R.; Combaluzier, H.; Leroy, V.; and Rogez, G. 2020. DOPE: Distillation Of Part Experts for whole-body 3D pose estimation in the wild. In *ECCV*, 380–397.

Wu, A.; Zheng, W.-S.; Guo, X.; and Lai, J.-H. 2019. Distilled person re-identification: Towards a more scalable system. In *CVPR*, 1187–1196.

Xiao, F.; Sigal, L.; and Jae Lee, Y. 2017. Weakly-supervised visual grounding of phrases with linguistic structures. In *CVPR*, 5945–5954.

Xie, J.; Lin, S.; Zhang, Y.; and Luo, L. 2019. Training convolutional neural networks with cheap convolutions and online distillation. *arXiv preprint arXiv:1909.13063*.

Xie, Q.; Luong, M.-T.; Hovy, E.; and Le, Q. V. 2020. Self-training with noisy student improves imagenet classification. In *CVPR*, 10687–10698.

Yang, C.; Xie, L.; Su, C.; and Yuille, A. L. 2019. Snapshot distillation: Teacher-student optimization in one generation. In *CVPR*, 2859–2868.

Yim, J.; Joo, D.; Bae, J.; and Kim, J. 2017. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*, 4133–4141.

You, S.; Xu, C.; Xu, C.; and Tao, D. 2017. Learning from multiple teacher networks. In *KDD*, 1285–1294.

Yu, C.; Wang, J.; Gao, C.; Yu, G.; Shen, C.; and Sang, N. 2020. Context Prior for Scene Segmentation. In *CVPR*, 12416–12425.

Zagoruyko, S.; and Komodakis, N. 2017. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*.

Zhang, Y.; Lan, Z.; Dai, Y.; Zeng, F.; Bai, Y.; Chang, J.; and Wei, Y. 2020. Prime-Aware Adaptive Distillation. In *ECCV*, 658–674.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *CVPR*, 2881–2890.

Zhao, H.; Sun, X.; Dong, J.; Chen, C.; and Dong, Z. 2020. Highlight every step: Knowledge distillation via collaborative teaching. *IEEE Transactions on Cybernetics*.

Zheng, K.; Lan, C.; Zeng, W.; Liu, J.; Zhang, Z.; and Zha, Z.-J. 2021. Pose-Guided Feature Learning with Knowledge Distillation for Occluded Person Re-Identification. In *Multimedia*. ACM.

Zhu, X.; Gong, S.; et al. 2018. Knowledge distillation by on-the-fly native ensemble. In *NeurIPS*, 7517–7527.