Class Specialized Knowledge Distillation

Li-Yun Wang[1], Anthony Rhodes[2], and Wu-chi Feng[1]

[1] Portland State University, Portland OR 97201, United States
{liyuwang, wuchi}@pdx.edu
[2] Intel Labs, Santa Clara CA 95054, United States
anthony.rhodes@intel.com

Abstract. Knowledge Distillation (KD) is a compression framework that transfers distilled knowledge from a teacher to a smaller student model. KD approaches conventionally address problem domains where the teacher and student network have equal numbers of classes for classification. We provide a knowledge distillation solution tailored for class specialization, where the user requires a compact and performant network specializing in a subset of classes from the class set used to train the teacher model. To this end, we introduce a novel knowledge distillation framework, Class Specialized Knowledge Distillation (CSKD), that combines two loss functions: Renormalized Knowledge Distillation (RKD) and Intra-Class Variance (ICV) to render a computationally-efficient, specialized student network. We report results on several popular architectural benchmarks and tasks. In particular, CSKD consistently demonstrates significant performance improvements over teacher models for highly restrictive specialization tasks (e.g., instances where the number of subclasses or datasets is relatively small), in addition to outperforming other state-of-the-art knowledge distillation approaches for class specialization tasks.

Keywords: Neural Network Compression · Class Specialization · Knowledge Distillation.

1 Introduction

Researchers have demonstrated the success of Deep Convolutional Neural Networks (DCNNs) on a wide range of computer vision applications including image recognition [35][63][5], instance-based and pixel-level image segmentation [50][53][33], and object localization in images and videos [47][7][49][25]. Oftentimes, state-of-the-art DCNN models are unwieldy and require a significant amount of computation time and memory space for training and inference, which can limit their real-world usability, particularly for mobile and edge applications. Neural network compression techniques have been dedicated to alleviating these issues by removing less activated parameters in complex models [15][38][11][64][19][27][55] or leveraging knowledge distillation [17][40][57][18][51][1][34] to train a smaller student network.

Knowledge distillation is a teacher-student learning methodology that aims to train a compact student neural network by replicating the implicit knowledge

encoded in a larger teacher model. In general, KD generates a pre-trained neural network (i.e., a teacher network) and transfers the teacher's knowledge to a student network by minimizing the difference between the outputs of the two networks. Yun et al. [56] combine a self-knowledge distillation technique with class-wise prediction regularization to tackle the issue of overfitting for neural network training. By penalizing the predictive distribution between similar samples, their approach achieves competitive classification accuracy. Muller et al. [32] show that the student network can experience sub-optimal knowledge transfer from the teacher network when using coarsely-defined class labels. Their work improved the knowledge transfer between the teacher and student by enabling the teacher to partition class labels into multiple subclasses.

There exist a wide range of practical applications and use cases for class specialized neural networks [45][20][22]. Many general-purpose ensembling methods in machine learning [9][36] leverage specialized *experts*, including [59][43]. In addition, most AI-assisted real-world manufacturing processes require fine-grain model specialization [60][13][39][4][3][8][48], as do a variety of deployed models in Medicine [60][13], Biology [39][4], Agriculture [3][8], and vital supply chain operations [48]. These specialization domain challenges are also frequently exacerbated due to inherent data scarcity and annotation costs.

Several recent works have called attention to class or task specification problems in relation to knowledge distillation. Shen et al. [43] and Zaras et al. [59] aggregate the knowledge from multiple teacher networks and transfer it to the student. Morgado et al. [31] use a teacher network fine-tuned on a specific task as guidance to train task-specific proxy layers in a student network. This method also focuses on specialized tasks, but it requires a fine-tuning of the remaining parameters in the proxy layers. Kao et al. [21] present a KD technique to improve the overall accuracy on weak classes by transferring distilled outputs from multiple teacher networks to a single student network. Notably, this approach does not produce a compact, specialized network for specific subclasses.

In this paper, we focus on the problem of training a compact student network for explicit specialized classes applicable in real-world specialization tasks to simultaneously reduce compute overhead and improve data efficiency costs. We present a novel KD framework to generate a compact student network for class specialization by restricting knowledge transfer from the teacher model to a subset of *classes of interest*. This specialized knowledge transfer is effected primarily using Renormalized Knowledge Distillation (RKD) loss. We furthermore regularize this knowledge distillation process by simultaneously minimizing the intra-class variance for latent representations among all subclasses in the student network with the introduction of Intra-Class Variation (ICV) loss. We show that these two loss functions work in tandem to bolster class specialization performance for compact student networks through empirical experiments and qualitative analyses. Our proposed technique is generalizable across a variety of different model architectures and vision tasks, including image classification and transfer learning applications.

The contributions of our work are as follows:

1. We introduce a novel KD technique using the proposed RKD and the ICV loss functions for class specialization problems.
2. We empirically evaluate our proposed technique on standard benchmarks models and image datasets. Our experiments show that the proposed technique is competitive with, and frequently outperforms, the state-of-the-art KD techniques for specialized student networks.
3. We further demonstrate the generalizability of our proposed technique by generating specialized neural networks on both image classification and transfer learning tasks.

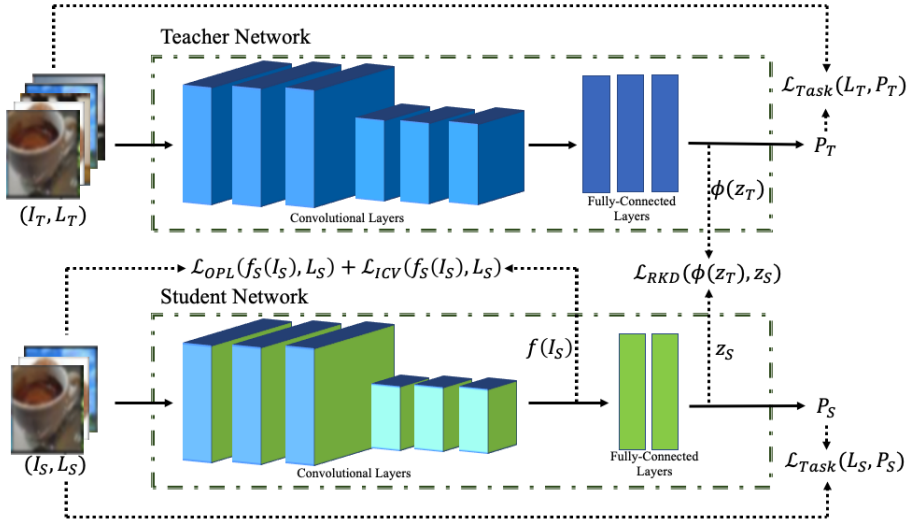

Fig. 1. Overview of our CSKD approach. During the knowledge distillation stage, the student network calculates the Renormalized Knowledge Distillation Loss ($\mathcal{L}_{RKD}$), Orthogonal Projection Loss ($\mathcal{L}_{OPL}$), and the Intra-Class Variance Loss ($\mathcal{L}_{ICV}$) given the teacher network trained on training data with all classes. $f_S(I_S)$ represents a feature extractor that outputs feature embeddings for the student network given an image batch. ϕ is a mapping function that chooses a subset of logits from the teacher and transfers it to the student. P_T and P_S are a prediction for the teacher and the student. z_T and z_S represent a logit for the teacher and the student, respectively.

2 Related Work

2.1 Knowledge Distillation

Many knowledge distillation approaches have been applied to network model compression problems by training a student network with fewer parameters that

nevertheless achieve competitive performance with large-scale models [17][34] [65][1][6][51][40][57]. These approaches mainly focus on the transfer of probability-based knowledge [17][6], latent representation knowledge [57][40][34][1][51], or combinations of both types of knowledge [65] to the student. Another common research tactic for knowledge distillation centers around self-knowledge transfer approaches. Yun et al. [56] leverage the concept of self-distillation to propose a class-wise prediction regularization for reducing overfitting and improving model generalization. Zhang et al. [61] propose a self-distillation framework that extracts representations of knowledge from different depths of attention modules to enhance model performance without considering teacher networks. Zheng et al. [62] utilize a self-guidance technique where they train the predictions of multiple sub-networks (student networks) to match the predictions of a complete network (a teacher network) to strengthen model generalization. Other research focuses on KD techniques for multiple teacher networks [44][29][59]. Each of these methods partitions the data into multiple subsets associated with different classes and then executes various heterogenous or homogeneeous KD processes. Lastly, they aggregate the representations of knowledge from all teacher networks and transfer them to the student. Although the aforementioned approaches can successfully produce a compressed variant of the teacher model, they do not explicitly generate a lightweight student network to solve class specialization problems.

2.2 Task Specialization

Some relevant research work on network model compression for class specialization applications propose the KD framework among multiple teacher networks and one or more student networks [43][59][21] or prune a neural network via a non-KD technique [12]. Shen et al. [43] propose a knowledge amalgamation framework to combine with teacher outputs from multiple pre-trained teacher models and leverage the combined teacher outputs to learn a lightweight student network for comprehensive classification. Like Shen's work [43], Zaras et al. [59] utilize a similar idea to ensemble multiple teacher networks trained on non-overlapping subclasses given the entire class and efficiently distill the knowledge from all the teachers to a compact student network for the whole class. Kao et al. [21] also propose an ensemble-based KD framework for making a student network by leveraging multiple expert networks (teacher networks) to better guide student knowledge acquisition. As a result of aggregating the predictions of multiple teacher-student frameworks that specialize in specific tasks, the trained student network achieves better classification performance. Gabbay et al. [12] define a value-locality-based network compression algorithm to search for the specific neurons of the activations associated with high degree values given specialized tasks and replace these neurons with the associated average arithmetic values. Since this method does not use fine-tuning, their approach can only match the performance of the uncompressed neural network. Each of these prior approaches require specialization of the teacher network while distilling knowledge injectively (that is, for an equal number of classes) to the student. By contrast,

our proposed KD framework focuses on explicitly tailoring the distillation process itself to the required task specialization problem.

3 Our Approach

In this section we provide an overview of our CSKD framework and give details of each loss function used to train our student network. Our method is designed to generate a lightweight student network for specialized classes (e.g., 5 or 10 specific classes) given the entire set of class (e.g., 100 or 200 classes) using the RKD and ICV loss functions. Figure 1 shows the overview of the proposed knowledge distillation framework for neural network compression tasks on specialized classes. RKD loss effectively transfers the teacher logit information associated with the target classes of interest to the student, so we can integrate the latent data representations learned by the teacher into the student training process. In addition, we introduce ICV loss to regularize student training by minimizing the intra-class variance of feature embeddings output by the penultimate layer of the student network.

3.1 Knowledge Distillation

In this section, we revisit the orthodox knowledge distillation methodology [2] [17]. Given a teacher network T and a student network S, we define f_T and f_S as a function approximated by a deep neural network for the teacher and the student. We also define z_T and z_S as a logit for the teacher and the student network. Then, we consider X^2, the set of tuples with two distinct elements, as a set of data points. Specifically, we denote $X^2 = \{(x_i, x_j) \mid i \neq j\}$ henceforth.

In the teacher-student framework setting, knowledge distillation [17] aims to minimize the following objective function, given the logits of the teacher and the student network:

$$\mathcal{L}_{KD} = \alpha \sum_{x_i \in \mathcal{X}} L(z_T,\ z_S) \tag{1}$$

where L is a loss function minimizing the difference between the output of the teacher and the student. Additionally, α is a hyperparameter used to control the severity of penalty for the knowledge distillation loss.

Researchers have proposed a variety of different loss functions to calculate the difference between two logits or feature embeddings. Hinton et al. [17] normalize logits of the teacher and the student via softmax and leverage Kullback-Leibler (KL) divergence to calculate the difference between them for the loss function L:

$$\sum_{x_i \in \mathcal{X}} \mathcal{KL}(softmax(\frac{z_T}{\tau}),\ softmax(\frac{z_S}{\tau})) \tag{2}$$

where τ is a hyperparameter, temperature, that controls the smoothness of the probability distribution. As τ increases, the $\mathcal{KL}$ loss is more sensitive to differences between the teacher and student logits.

Unlike the Hinton et al. work, Romero et al. [40] transfer knowledge from the teacher to the student by minimizing the difference between the feature embeddings of the teacher and the student for the loss function L:

$$\sum_{x_i \in \mathcal{X}} \|f_T(x_i) - M(f_S(x_i))\|^2 \tag{3}$$

where $\|.\|$ represents the Euclidean norm. M is a mapping function that takes the feature embeddings of the student as input and aligns the embeddings of the student with the feature embeddings of the teacher.

As in [40][17], many researchers have proposed related knowledge distillation methods [57][54][51][1][34][30] based on Eq. 1. Notably, these methods transfer all outputs of the teacher network to the student network. As such, these methods can only be used in conventional knowledge distillation frameworks, and are therefore not directly applicable with specialization tasks.

3.2 Renormalized Knowledge Distillation

We introduce Renormalized Knowledge Distillation (RKD) as a mechanism to transfer only part of the outputs of the teacher to a specialized student network. Unlike conventional knowledge distillation approaches, RKD loss leverages a mapping function ϕ to select a subset of the logits from the teacher and normalizes this subset via the softmax function. Once the renormalized teacher logits are generated, the loss transfers the logit information solely for the classes of interest from the teacher to the student.

Formally, we define RKD loss as follows:

$$\hat{z_T} = \phi(z_T) \tag{4}$$

$$\mathcal{L}_{RKD} = \alpha \sum_{x_i \in \mathcal{X}} \mathcal{KL}(softmax(\frac{\hat{z_T}}{\tau}) \text{ , } softmax(\frac{z_S}{\tau})) \tag{5}$$

where $\hat{z_T}$ is the output of the mapping function ϕ for the teachers logits, ϕ is the mapping function that identifies the subset of the teacher logits corresponding with the specialized classes of interest, and $\mathcal{KL}$ is the KL divergence loss minimizing the difference between the teacher and student logits. By applying the mapping function, RKD loss can select any number of the logits from the teacher and transfer specific knowledge associated with the logits from the classes of interest regardless of the logit dimensions of the teacher and the student networks. Notice that RKD generalizes as conventional knowledge distillation loss, so that when the teacher and student logit dimensions are equal, Eq. 5 reduces to Eq. 2 and Eq. 1.

3.3 Regularization Loss for Feature Embeddings

RKD loss transfers the part of logits associated with classes of interest for subclass specialization from the teacher to the student, but it does not directly leverage the feature embeddings learned from a network to enhance classification performance for model training. It is well-known [24][10][46][26][14] that DNNs encode feature embeddings hierarchically. Accordingly, researchers have applied these hierarchically-generated features to a large variety of different problems [34][51][57][53][42] to achieve competitive performance. Inspired by previous works [34][51][57], we similarly utilize feature embeddings to enhance classification performance for the student by maximizing inter-class *angular* distance while simultaneously minimizing intra-class feature *spatial* variance.

To maximize the inter-class embedding distance given feature embeddings from the penultimate layer, we adopt Orthogonal Projection Loss (OPL) [37] to enforce class-wise orthogonality in the feature embeddings. OPL is defined as follows:

$$s = \Big(\sum_{\substack{i,j \in B \\ y_i = y_j}} \langle f(\boldsymbol{x}_i), f(\boldsymbol{x}_j) \rangle \Big) \; / \; \Big(\sum_{\substack{i,j \in B \\ y_i = y_j}} 1 \Big) \tag{6}$$

$$d = \Big(\sum_{\substack{i,k \in B \\ y_i \neq y_k}} \langle f(\boldsymbol{x}_i), f(\boldsymbol{x}_k) \rangle \Big) \; / \; \Big(\sum_{\substack{i,k \in B \\ y_i \neq y_k}} 1 \Big) \tag{7}$$

$$\mathcal{L}_{OPL} = (1 - s) + |d| \tag{8}$$

where $f(\boldsymbol{x})$ is a feature embedding given an input, $\langle ., . \rangle$ is a similarity measurement function that takes two input vectors and computes the cosine value between the vectors, and B is a mini-batch size. Minimizing OPL equates to enforcing orthogonality between embeddings of data points in different classes. This result is achieved by calculating the similarity of input pairs with the same class and the dissimilarity of input pairs with distinct classes in an image batch. OPL thus effectively increases the angular distance of inter-class embeddings in the feature representation space for model training.

Although OPL can render class-wise orthogonality in the feature embeddings, it nevertheless only considers the cosine similarity and dissimilarity between input pairs without regard for minimizing the spatial extent of intra-class embeddings within a single class. To help achieve this end, we introduce Intra-Class Variance (ICV) loss to enforce intra-class variance minimization for all feature embeddings with the same target class by calculating the variance of the outputs of the penultimate layer for each target class. In our experiments, we demonstrate that ICV loss can be used in tandem with OPL to further improve the effectiveness of learned latent embeddings.

Formally, we formulate the ICV loss using the following equations:

$$\boldsymbol{f_{var}} = \psi(f_S(\boldsymbol{X}_i)) \quad i = 1 \; ... \; C \tag{9}$$

$$\mathcal{L}_{ICV} = \sum_{i=1}^{C} \| \boldsymbol{f_{var}} \|_F \tag{10}$$

where f_S is an approximate feature extractor function via student neural networks, $f_S(\boldsymbol{X}_i)$ is an $N \times D$ feature embedding matrix with the same target class, ψ is a function that calculates variance for every embedding, $\boldsymbol{f_{var}}$ is a $1 \times D$ vector, and $\|.\|_F$ is a normalization function that computes the Frobenius norm for the embedding vectors. To calculate gradients for weight updating, we need to ensure that ICV loss is differentiable. ICV is comprised of conventional variance and $L2$ norm for calculations for matrices, indicating that ICV is made up of differentiable functions, and therefore differentiable. Thus, if one defines the output of the network forward pass: $o_i = \mathcal{L}_{ICV}(f_S(\boldsymbol{X}_i))$, then the gradients required for backpropagation of the network can be obtained via $g_i = \frac{\partial \mathcal{L}_{ICV}}{\partial w}$, where w denotes a network parameter.

3.4 Training with Losses

Our final student model is trained using a combination of losses, including, including RKD and ICV losses, a task-specific loss, and OPL. The task-specific loss can be a conventional cross-entropy loss, say, in the case of classification problems, or a different, bespoke loss for function for different problem domains. Our total loss for the student network is defined:

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \alpha \mathcal{L}_{RKD} + \beta \mathcal{L}_{OPL} + \gamma \mathcal{L}_{ICV} \tag{11}$$

where $\mathcal{L}_{task}$ is the task-specific loss, α is a hyperparameter for the RKD loss, β is a hyperparameter for OPL, and γ is a hyperparameter for the ICV loss. All of these hyperparameters can help to regularize the $\mathcal{L}_{task}$ loss and thereby improve model performance. We utilize a heuristically-chosen small positive value for all the hyperparameters, since $\mathcal{L}_{task}$ is still the most crucial guideline for student network training.

4 Experimental Results

We empirically evaluate CSKD on image classification and transfer learning specialization tasks. In addition, we conduct ablation studies that assess the effect of different numbers of subclasses for specialization performance, in addition to testing different hyperparameter values for each of the loss functions appearing in (11). For fair comparisons, we follow the same experimental settings as reported in relevant baseline KD research in our experiments [1][34]. We then choose the student networks in accordance with standard knowledge distillation teacher-student comparison practices [51][1].

4.1 Ablation Study Setup

We first conduct ablation studies with respect to the number of subclasses used for specialization, in addition to the hyperparameter values used in our loss function. Our experiments encompass several different benchmark architectures,

Model: (Teacher, Student)	**(WRN-40-2, WRN-16-1)**			
Benchmark	CIFAR10	CIFAR100		
(Class Frac., Subclass Num.)	(50%, 5)	(5%, 5)	(10%, 10)	(50%, 50)
Teacher ($\mathcal{L}_{\mathcal{CE}}$)	94.98%	81.60%	77.5%	74.56%
Student ($\mathcal{L}_{\mathcal{CE}}$)	93.68%	93.20%	89.20%	72.82%
Student ($\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}}$)	93.82%	93.59%	89.60%	73.24%
Student ($\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}} + \mathcal{L}_{\mathcal{OPL}}$)	94.42%	93.99%	89.80%	73.32%
Student ($\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}} + \mathcal{L}_{\mathcal{OPL}} + \mathcal{L}_{\mathcal{ICV}}$)	**94.66%**	**95.19%**	**90.70%**	**74.24%**

Table 1. The table shows top-1 accuracy comparisons for distinct subclasses of the entire class for WRN. $\mathcal{L}_{\mathcal{CE}}$ represents the cross-entropy loss in the experiment. The hyperparameter of each loss term (e.g., $\mathcal{L}_{\mathcal{RKD}}$, $\mathcal{L}_{\mathcal{OPL}}$, and $\mathcal{L}_{\mathcal{ICV}}$) is 0.1.

including standard residual networks (Resnet) [16] and wide residual networks (WRN) [58] across the CIFAR10, CIFAR100, and Tiny ImageNet [23] benchmark datasets. We consider these ablation studies to better understand the effects of the proposed loss functions and to test the robustness of our method for different degrees of task specialization. For all ablation experiments, we train the teacher network on the normative training sets (e.g., all 10 or 100 classes for CIFAR10 and CIFAR100, respecetively) and test the trained network on the images associated with specializeed classes (e.g., 5, 10, and 50 classes). Additionally, we train and test the student networks only on the identified subclasses.

Model: (Teacher, Student)	**(Resnet34, Resnet18)**		
Benchmark	Tiny ImageNet		
(Class Frac., Subclass Num.)	(2.5%, 5)	(5%, 10)	(25%, 50)
Teacher ($\mathcal{L}_{\mathcal{CE}}$)	56.00%	59.20%	60.08%
Student ($\mathcal{L}_{\mathcal{CE}}$)	73.60%	75.19%	67.03%
Student ($\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}}$)	76.80%	75.59%	67.99%
Student ($\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}} + \mathcal{L}_{\mathcal{OPL}}$)	**79.20%**	77.59%	**68.47%**
Student ($\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}} + \mathcal{L}_{\mathcal{OPL}} + \mathcal{L}_{\mathcal{ICV}}$)	78.40%	**77.99%**	68.15%

Table 2. The table shows top-1 accuracy comparisons for distinct subclasses of the entire class for Resnet. $\mathcal{L}_{\mathcal{CE}}$ represents the cross-entropy loss in the experiment. The hyperparameter of each loss term (e.g., $\mathcal{L}_{\mathcal{RKD}}$, $\mathcal{L}_{\mathcal{OPL}}$, and $\mathcal{L}_{\mathcal{ICV}}$) is 0.1.

We follow the same experimental settings as [34] to conduct our ablation experiments. For CIFAR10 and CIFAR100, we pad zeros to input images to

have 40 × 40 images and randomly crop 32 × 32 cropped images. Additionally, we use random horizontal flipping for data augmentation. We utilize SGD with batch size 128, momentum 0.9, and weight decay 5 × 10-4 to train networks for 200 epochs. We also apply learning rate decay starting with 0.1 by multiplying by 0.2 at 60, 120, and 160 epochs. Finally, we use WRN-40-2 and WRN-16-1 for a teacher and a student network. For Tiny ImageNet, we resize input images to 256 × 256 and randomly crop the input images to generate 224 × 224 cropped images for network training. We also use random color jittering and horizontal flipping for data augmentation. We train the teacher and student network for 300 epochs and apply adjustable learning rates from 0.1 to small values by multiplying by 0.2 at 60, 120, 160, 200, and 250 epochs. In addition, we replicate these same experimental conditions for Resnet34 and Resnet18 network architectures as a teacher and a student network, respectively.

4.2 Ablation Study Experiment: Subclass Numbers for the Student

Model: (Teacher, Student)	**(WRN-40-2, WRN-16-1)**					
Benchmark	CIFAR10			CIFAR100		
Teacher ($\mathcal{L}_{\mathcal{CE}}$)	94.98%	94.98%	94.98%	81.60%	81.60%	81.60%
Student ($\mathcal{L}_{\mathcal{CE}}$)	93.68%	93.68%	93.68%	93.20%	93.20%	93.20%
Hyperparameters	0.1	0.15	0.2	0.1	0.15	0.2
Student ($\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}}$)	93.82%	94.14%	94.00%	93.59%	94.00%	94.19%
Student ($\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}} + \mathcal{L}_{\mathcal{OPL}}$)	94.42%	94.30%	94.46%	93.99%	94.39%	94.79%
Student ($\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}} + \mathcal{L}_{\mathcal{OPL}} + \mathcal{L}_{\mathcal{ICV}}$)	**94.66%**	94.36%	94.62%	**95.19%**	94.99%	**95.19%**

Table 3. Top-1 accuracy comparisons for a variety of hyperparameters of every penalty loss (e.g., $\mathcal{L}_{\mathcal{RKD}}$, $\mathcal{L}_{\mathcal{OPL}}$, and $\mathcal{L}_{\mathcal{ICV}}$) for WRN. All experimental results are conducted by 5 subclasses. $\mathcal{L}_{\mathcal{CE}}$ represents the cross-entropy loss in the experiment.

We test our method using randomly selected subclasses of varying sizes (e.g., 5, 10, 50) to better understand the effects of different subclasses for top-1 classification accuracy for task specialization on CIFAR10, CIFAR100, and Tiny ImageNet. We list empirical evaluation results in Table 1 and Table 2. As we can see, most student networks trained using CE achieve higher top-1 accuracy for small subclasses (e.g., 5, 10, and 50) compared with the teacher networks. For CIFAR10 and CIFAR100 with 50 subclasses, the teacher has better top-1 accuracy than the top-1 accuracy of the students. We observe that the effectiveness of class specialization through knowledge distillation is sensitive to the relative number of subclasses (i.e., in proportion to the total number of teacher classes). Notably, the student performance is generally comparable to that of the

teacher when the relative number of subclasses is large (e.g., 50%); however, in the case of a considerably small relative number of subclasses (e.g., $5-10\%$), the student performance is often dramatically better. We believe that these experimental results indicate the strong *data efficiency* potential encapsulated by the CSKD framework. Despite an extreme scarcity of "specialized" training data, it is nevertheless possible to successfully train a student network that substantially outperforms a fully-trained teacher on class specialized tasks (in one case we train a student network on only 2.5% of the Tiny ImageNet training data, however the student model renders a 40% relative improvement over the teacher).

Model: (Teacher, Student)	**(Resnet34, Resnet18)**		
Benchmark	Tiny ImageNet		
Teacher ($\mathcal{L}_{\mathcal{CE}}$)	56.00%	56.00%	56.00%
Student ($\mathcal{L}_{\mathcal{CE}}$)	73.60%	73.60%	73.60%
Hyperparameters	0.1	0.15	0.2
Student ($\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}}$)	76.80%	76.80%	75.20%
Student ($\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}} + \mathcal{L}_{\mathcal{OPL}}$)	**79.20%**	79.20%	76.00%
Student ($\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}} + \mathcal{L}_{\mathcal{OPL}} + \mathcal{L}_{\mathcal{ICV}}$)	78.40%	76.80%	78.40%

Table 4. Top-1 accuracy comparisons for a variety of hyperparameters of every penalty loss (e.g., $\mathcal{L}_{\mathcal{RKD}}$, $\mathcal{L}_{\mathcal{OPL}}$, and $\mathcal{L}_{\mathcal{ICV}}$) for Resnet. All experimental results are conducted by 5 subclasses. $\mathcal{L}_{\mathcal{CE}}$ represents the cross-entropy loss in the experiment.

4.3 Ablation Study Experiment: Hyperparameters for Loss Terms

We report the effects of varying hyperparameter values for the CIFAR10, CIFAR100, and Tiny ImageNet datasets in Table 3 and Table 4. All experimental results are conducted on five target subclasses. Setting the hyperparameter of all three loss terms (e.g., $\mathcal{L}_{\mathcal{RKD}}$, $\mathcal{L}_{\mathcal{OPL}}$, and $\mathcal{L}_{\mathcal{ICV}}$) to 0.1 achieves consistently competitive top-1 classification accuracy compared with other hyperparameter values. Specifically, the student model with all three hyperparameter values set to 0.1 achieves classification accuracy of 94.66% and 95.19% for CIFAR10 and CIFAR100, respectively. In addition, we see that most of the students utilizing all three loss functions in tandem have the highest top-1 classification accuracy compared with the student with only one or two losses. These results help validate our hypothesis that RKD, OPL, and ICV can effectively be leveraged in tandem to enhance class specialized KD performance by striking an agreeable balance between data-efficient knowledge distillation and model regularization.

In order to provide a qualitative analysis of our loss functions, we visualize the feature embeddings of the penultimate student layer by t-SNE [28] in figure

2 and figure 3. In particular, in both figures, (b) illustrates the latent structure inculcated to the student model through RKD loss; and (d) shows the effect of dense class embeddings induced by ICV.

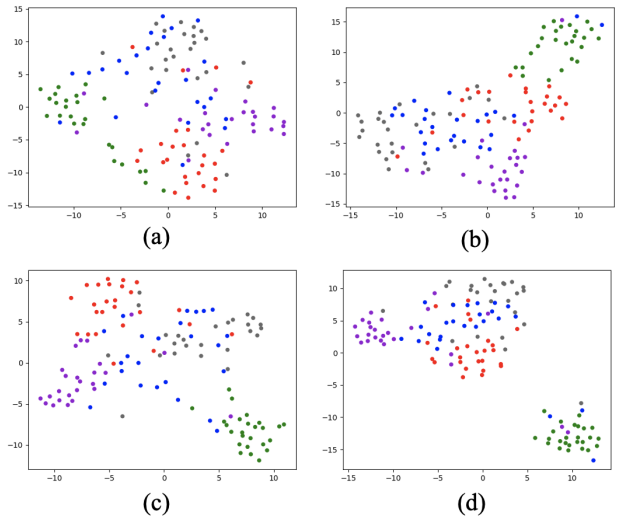

Fig. 2. Feature embedding t-SNE visualization of the penultimate layer for Tiny ImageNet with 5 target subclasses. Each plot represents: (a) $\mathcal{L}_{\mathcal{CE}}$, (b) $\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}}$, (c) $\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}} + \mathcal{L}_{\mathcal{OPL}}$, and (d) $\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}} + \mathcal{L}_{\mathcal{OPL}} + \mathcal{L}_{\mathcal{ICV}}$

4.4 Classification Performance Comparisons

We evaluate our proposed method for image classification with specialized image classes using four distinct benchmarks (CIFAR10, CIFAR100, Tiny ImageNet, and CUB200 [52]). In order to assess specialized image classification performance on our student networks, we randomly select five subclasses from the entire set of classes for each of the benchmark datasets. For the CIFAR10, CIFAR100, and Tiny ImageNet, datasets, the teacher network is trained on the *entire* training dataset (i.e., all classes) and tested on five selected subclasses. By contrast, the student networks are trained and tested using only the five selected subclasses. Because the CUB200 benchmark dataset consists of only 11,788 images for 200 different bird species, we acquire a pre-trained Resnet34 model on the ImageNet benchmark [41] and fine-tune this model on the CUB200 benchmark for the teacher network. For the student, we train the student network (e.g., Resnet18) on the images associated with five random subclasses *from scratch* and test the trained student network on the same subclasses. We evaluate our proposed losses and several state-of-the-art knowledge distillation approaches (e.g., FitNet [40], Attention [57], VID [1], and the Relational Knowledge Distillation (ReKD) loss

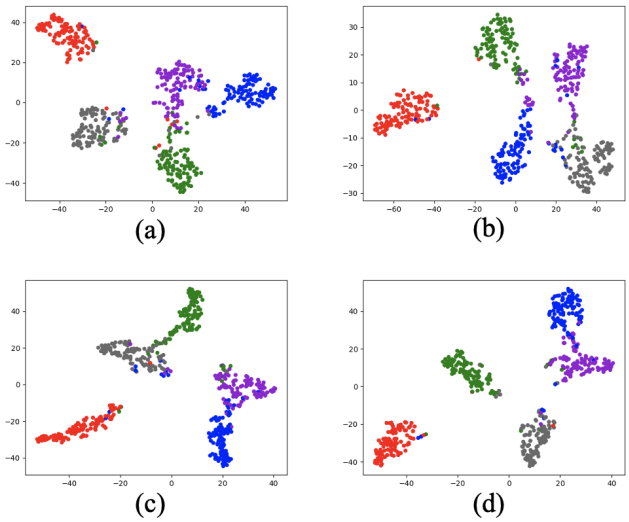

Fig. 3. Feature embedding t-SNE visualization of the penultimate layer for CIFAR100 with 5 target subclasses. Each plot represents: (a) $\mathcal{L}_{\mathcal{CE}}$, (b) $\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}}$, (c) $\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}} + \mathcal{L}_{\mathcal{OPL}}$, and (d) $\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}} + \mathcal{L}_{\mathcal{OPL}} + \mathcal{L}_{\mathcal{ICV}}$

	CIFAR10	**CIFAR100**	**Tiny ImageNet**	**CUB200**
Class Fraction (%)	50	5	2.5	2.5
Teacher	94.98%	81.60%	56.00%	63.96%
Student ($\mathcal{L}_{\mathcal{CE}}$)	93.68%	93.20%	73.60%	51.40%
Student (FitNet) [40]	93.86%	93.59%	75.20%	57.04%
Student (Attention) [57]	94.54%	94.40%	78.40%	55.63%
Student (VID) [1]	93.82%	93.79%	78.40%	59.15%
Student (ReKD) [34]	94.06%	92.79%	78.40%	59.15%
Student ($\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}}$) (Ours)	93.82%	93.59%	76.80%	58.45%
Student ($\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}} + \mathcal{L}_{\mathcal{OPL}}$) (Ours)	94.42%	93.99%	**79.20%**	57.74%
Student ($\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}} + \mathcal{L}_{\mathcal{OPL}} + \mathcal{L}_{\mathcal{ICV}}$) (Ours)	**94.66%**	**95.19%**	78.40%	**61.97%**

Table 5. Top-1 classification accuracy (%) for image classification on CIFAR10, CIFAR100, Tiny ImageNet, and CUB200. We select hyperparameter 0.1 to the three losses in our proposed framework.

[34]) on the same four benchmarks. We select these four approaches as the baseline techniques since these KD methods transfer the feature embeddings rather than the logits from the teacher to the student. These approaches thus avoid the issue of dimensionality misalignment between the teacher and student logits for class specialization tasks. For the hyperparameters of these approaches, we set $\lambda_{Attention}$ to 50. Additionally, we set λ_{ReKD-D} to 25and λ_{ReKD-D} to 50 in the ReKD loss. Then, we set λ_{FitNet} and λ_{VID} to 0.1. We use the same setting mentioned in Section 4.1 for model training and use the same experiment setting for Tiny ImageNet for the CUB200 dataset. Table 5 shows top-1 classification accuracy on the four benchmarks using five target subclasses. As we can see, a number of student networks outperform the teacher networks on *well-calibrated specialized classification tasks*, particularly when the class fraction percentage (i.e., number of subclasses divided by the total number of classes) is relatively low and the class representation is balanced. From these results, CSKD either achieves competitive classification performance or outperforms existing state-of-the-art knowledge distillation approaches for the tested class specialization tasks. Moreover, in each experiment, CSKD demonstrates significant performance improvements over a "naive" student network trained solely using $\mathcal{L}_{\mathcal{CE}}$. In the case of the challenging CUB200 dataset, despite the absence of pre-trained features, the more compact CSKD model utilizing our total loss function ($\mathcal{L}_{\mathcal{CE}} + \mathcal{L}_{\mathcal{RKD}} + \mathcal{L}_{\mathcal{OPL}} + \mathcal{L}_{\mathcal{ICV}}$) yields over 20% relative performance improvement over a naive student, and performs only marginally worse than the larger, pre-trained teacher model for the class specialization task. These results empirically validate the essential thrust of of our CSKD approach. In place of greedily transferring the knowledge of feature embeddings of intermediate layers from the teacher to the student, our proposed approach instead (1) only transfers specific teacher logits to the student network and (2) regularizes the student network by concurrently enforcing orthogonality in between the classes and minimizing intra-class variance.

5 Conclusion

In this paper, we presented the CSKD framework combining RKD and ICV loss functions to render a compact and performant student network for the class specialization problem setting. The proposed RKD loss improves the efficiency of KD for class specialized tasks by transferring only the relevant portion of the teacher output of the teacher to a specialized student network. Additionally, ICV loss enforces spatially dense feature embeddings by minimizing class-wise variance. CSKD consistently outperforms other knowledge distillation approaches for specialized student networks.

Acknowledgments We thank all the paper reviewers who provided constructive and knowledgeable feedback on our work for making our manuscript publishable.

References

1. Ahn, S., Hu, S.X., Damianou, A., Lawrence, N.D., Dai, Z.: Variational information distillation for knowledge transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9163–9171 (2019)
2. Ba, J., Caruana, R.: Do deep nets really need to be deep? Advances in neural information processing systems **27** (2014)
3. Bargoti, S., Underwood, J.: Deep fruit detection in orchards. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 3626–3633. IEEE (2017)
4. Bjerge, K., Nielsen, J.B., Sepstrup, M.V., Helsing-Nielsen, F., Høye, T.T.: An automated light trap to monitor moths (lepidoptera) using computer vision-based tracking and deep learning. Sensors **21**(2), 343 (2021)
5. Chan, T.H., Jia, K., Gao, S., Lu, J., Zeng, Z., Ma, Y.: Pcanet: A simple deep learning baseline for image classification? IEEE transactions on image processing **24**(12), 5017–5032 (2015)
6. Choi, Y., Choi, J., El-Khamy, M., Lee, J.: Data-free network quantization with adversarial knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 710–711 (2020)
7. Cinbis, R.G., Verbeek, J., Schmid, C.: Weakly supervised object localization with multi-fold multiple instance learning. IEEE transactions on pattern analysis and machine intelligence **39**, 189–203 (2016)
8. Dias, P.A., Tabb, A., Medeiros, H.: Apple flower detection using deep convolutional networks. Computers in Industry **99**, 17–28 (2018)
9. Dietterich, T.G., et al.: Ensemble learning. The handbook of brain theory and neural networks **2**(1), 110–125 (2002)
10. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. IEEE transactions on pattern analysis and machine intelligence **35**(8), 1915–1929 (2012)
11. Frankle, J., Carbin, M.: The lottery ticket hypothesis: Finding sparse, trainable neural networks. arXiv preprint arXiv:1803.03635 (2018)
12. Gabbay, F., Shomron, G.: Compression of neural networks for specialized tasks via value locality. Mathematics **9**, 2612 (2021)
13. Ghorbani, A., Ouyang, D., Abid, A., He, B., Chen, J.H., Harrington, R.A., Liang, D.H., Ashley, E.A., Zou, J.Y.: Deep learning interpretation of echocardiograms. NPJ digital medicine **3**(1), 1–10 (2020)
14. Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., Lew, M.S.: Deep learning for visual understanding: A review. Neurocomputing **187**, 27–48 (2016)
15. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. Advances in neural information processing systems **28** (2015)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
17. Hinton, G., Vinyals, O., Dean, J., et al.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 **2**(7) (2015)
18. Huang, Z., Wang, N.: Like what you like: Knowledge distill via neuron selectivity transfer. arXiv preprint arXiv:1707.01219 (2017)
19. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. arXiv preprint arXiv:1602.07360 (2016)

20. Kang, D., Emmons, J., Abuzaid, F., Bailis, P., Zaharia, M.: Noscope: optimizing neural network queries over video at scale. arXiv preprint arXiv:1703.02529 (2017)
21. Kao, W.C., Xie, H.X., Lin, C.Y., Cheng, W.H.: Specific expert learning: Enriching ensemble diversity via knowledge distillation. IEEE Transactions on Cybernetics (2021)
22. Kosaian, J., Phanishayee, A., Philipose, M., Dey, D., Vinayak, R.: Boosting the throughput and accelerator utilization of specialized cnn inference beyond increasing batch size. In: International Conference on Machine Learning. pp. 5731–5741. PMLR (2021)
23. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N **7**, 3 (2015)
24. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)
25. Lee, K., Shrivastava, A., Kacorri, H.: Hand-priming in object localization for assistive egocentric vision. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3422–3432 (2020)
26. Lee, S.H., Chan, C.S., Mayo, S.J., Remagnino, P.: How deep learning extracts and learns leaf features for plant classification. Pattern Recognition **71**, 1–13 (2017)
27. Luo, J.H., Wu, J., Lin, W.: Thinet: A filter level pruning method for deep neural network compression. In: Proceedings of the IEEE international conference on computer vision. pp. 5058–5066 (2017)
28. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
29. Malinin, A., Mlodozeniec, B., Gales, M.: Ensemble distribution distillation. arXiv preprint arXiv:1905.00076 (2019)
30. Mirzadeh, S.I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., Ghasemzadeh, H.: Improved knowledge distillation via teacher assistant. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 5191–5198 (2020)
31. Morgado, P., Vasconcelos, N.: Nettailor: Tuning the architecture, not just the weights. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3044–3054 (2019)
32. Müller, R., Kornblith, S., Hinton, G.: Subclass distillation. arXiv preprint arXiv:2002.03936 (2020)
33. Papandreou, G., Zhu, T., Chen, L.C., Gidaris, S., Tompson, J., Murphy, K.: Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In: Proceedings of the European conference on computer vision (ECCV). pp. 269–286 (2018)
34. Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3967–3976 (2019)
35. Perez, L., Wang, J.: The effectiveness of data augmentation in image classification using deep learning. arXiv preprint arXiv:1712.04621 (2017)
36. Polikar, R.: Ensemble learning. In: Ensemble machine learning, pp. 1–34. Springer (2012)
37. Ranasinghe, K., Naseer, M., Hayat, M., Khan, S., Khan, F.S.: Orthogonal projection loss. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12333–12343 (2021)
38. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. In: European conference on computer vision. pp. 525–542. Springer (2016)
39. Ravoor, P.C., Sudarshan, T.: Deep learning methods for multi-species animal re-identification and tracking–a survey. Computer Science Review **38**, 100289 (2020)

40. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550 (2014)
41. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
42. Salehi, M., Sadjadi, N., Baselizadeh, S., Rohban, M.H., Rabiee, H.R.: Multiresolution knowledge distillation for anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14902–14912 (2021)
43. Shen, C., Wang, X., Song, J., Sun, L., Song, M.: Amalgamating knowledge towards comprehensive classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 3068–3075 (2019)
44. Shen, C., Xue, M., Wang, X., Song, J., Sun, L., Song, M.: Customizing student networks from heterogeneous teachers via adaptive knowledge amalgamation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3504–3513 (2019)
45. Shen, H., Han, S., Philipose, M., Krishnamurthy, A.: Fast video classification via adaptive cascading of deep models. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3646–3654 (2017)
46. Shrestha, A., Mahmood, A.: Review of deep learning algorithms and architectures. IEEE access **7**, 53040–53065 (2019)
47. Singh, K.K., Lee, Y.J.: Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In: 2017 IEEE international conference on computer vision (ICCV). pp. 3544–3553. IEEE (2017)
48. Syafrudin, M., Alfian, G., Fitriyani, N.L., Rhee, J.: Performance analysis of iot-based sensor, big data processing, and machine learning model for real-time monitoring system in automotive manufacturing. Sensors **18**(9), 2946 (2018)
49. Teh, E.W., Rochan, M., Wang, Y.: Attention networks for weakly supervised object localization. In: BMVC. pp. 1–11 (2016)
50. Tsai, Y.H., Zhong, G., Yang, M.H.: Semantic co-segmentation in videos. In: European Conference on Computer Vision. pp. 760–775. Springer (2016)
51. Tung, F., Mori, G.: Similarity-preserving knowledge distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1365–1374 (2019)
52. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
53. Wang, Y., Zhou, W., Jiang, T., Bai, X., Xu, Y.: Intra-class feature variation distillation for semantic segmentation. In: European Conference on Computer Vision. pp. 346–362. Springer (2020)
54. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4133–4141 (2017)
55. Yu, X., Liu, T., Wang, X., Tao, D.: On compressing deep models by low rank and sparse decomposition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7370–7379 (2017)
56. Yun, S., Park, J., Lee, K., Shin, J.: Regularizing class-wise predictions via self-knowledge distillation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13876–13885 (2020)
57. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928 (2016)

58. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
59. Zaras, A., Passalis, N., Tefas, A.: Improving knowledge distillation using unified ensembles of specialized teachers. Pattern Recognition Letters **146**, 215–221 (2021)
60. Zhang, J., Gajjala, S., Agrawal, P., Tison, G.H., Hallock, L.A., Beussink-Nelson, L., Lassen, M.H., Fan, E., Aras, M.A., Jordan, C., et al.: Fully automated echocardiogram interpretation in clinical practice: feasibility and diagnostic accuracy. Circulation **138**(16), 1623–1635 (2018)
61. Zhang, L., Bao, C., Ma, K.: Self-distillation: Towards efficient and compact neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
62. Zheng, Z., Peng, X.: Self-guidance: Improve deep neural network generalization via knowledge distillation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3203–3212 (2022)
63. Zhong, Z., Li, J., Luo, Z., Chapman, M.: Spectral–spatial residual network for hyperspectral image classification: A 3-d deep learning framework. IEEE Transactions on Geoscience and Remote Sensing **56**(2), 847–858 (2017)
64. Zhu, M., Gupta, S.: To prune, or not to prune: exploring the efficacy of pruning for model compression. arXiv preprint arXiv:1710.01878 (2017)
65. Zhu, Y., Wang, Y.: Student customized knowledge distillation: Bridging the gap between student and teacher. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5057–5066 (2021)