SSSD: Self-Supervised Self Distillation

Wei-Chi Chen
National Cheng Kung University
Tainan, Taiwan
iphone31302@gmail.com

Wei-Ta Chu
National Cheng Kung University
Tainan, Taiwan
wtchu@gs.ncku.edu.tw

Abstract

With labeled data, self distillation (SD) has been proposed to develop compact but effective models without a complex teacher model available in advance. Such approaches need labeled data to guide the self distillation process. Inspired by self-supervised (SS) learning, we propose a self-supervised self distillation (SSSD) approach in this work. Based on an unlabeled image dataset, a model is constructed to learn visual representations in a self-supervised manner. This pre-trained model is then adopted to extract visual representations of the target dataset and generates pseudo labels via clustering. The pseudo labels guide the SD process, and thus enable SD to proceed in an unsupervised way (no data labels are required at all). We verify this idea based on evaluations on the CIFAR-10, CIFAR-100, and ImageNet-1K datasets, and demonstrate the effectiveness of this unsupervised SD approach. Performance outperforming similar frameworks is also shown.

1. Introduction

Knowledge distillation [14] has been proven very effective to transfer knowledge of a large, relatively complex teacher model to a smaller student model, so that a lightweight student model can achieve similar or even better performance than the teacher model. Generally, a complex teacher model pre-trained based on a large dataset is needed in advance. However, large-scale and labeled datasets are not always available, and thus the requirement of a pre-trained teacher model cannot be met in all cases.

Self distillation eliminates the requirement of a pre-trained teacher model. A single network plays both the roles of teacher and student. For example, Zhang et al. [27] proposed that, in a ResNet structure, deeper residual blocks could be viewed as the teacher of shallower blocks, which are viewed as student models. Once the shallow parts (students) get improved, the deeper parts (teachers) could get improved also. Yang et al. [24] proposed that knowledge in the earlier epochs of a network (teacher) is transferred into its later epochs (student) to supervise the training process. In [27] and [24], only one network (serving both as the teachers and the students) is needed in the self distillation process.

Although the self distillation (SD) technique removes the requirement of a pre-trained teacher model, data labels are still needed to supervise the training process. Without labeled data, current self distillation methods cannot work because there is no ground to train the teacher. Motivated by the current progress of self-supervised learning, in this work we propose self-supervised self distillation (SSSD) to enable SD based on the pseudo labels given by a model pre-trained based on a pretext task. In this way, we can proceed SD without data labels and expand the applicability of SD.

Specifically, based on an unlabeled dataset D_{ssl}, we train a model M_{ssl} to complete the instance discrimination task [21], so that its ability of feature extraction is built. For the target dataset D_{tar} that is used for self distillation, the pre-trained model M_{ssl} is adopted to extract features. Features are then clustered by the K-means algorithm, and data points belonging to the same cluster are labeled with the same pseudo label. The pseudo labels serve as the (weak) ground to guide learning, and thus the self distillation process can be proceeded. We call this approach self-supervised self distillation (SSSD) for that no data label is needed in the entire process.

A similar structure has been proposed in [1], where the self-supervised model shares the same backbone network with the self distillation model, and conceptually SSL and SD are trained together. The main target of [1] is to release SSL from the limitation of specifically-designed data augmentation. The SSL model provides soft labels for learning, while the SD model further provides self-supervisory signals to improve performance of SSL. That is, SD is used to assist SSL. In our work, we view from the opposite viewpoint, i.e., we use SSL to assist SD. The main target of our work is to release SD from the limitation of well-labeled data.

Contributions of this work are twofold. First, we propose a self-supervised self distillation approach that enables SD

even if well-labeled data are not available. Second, we verify this idea based on a comprehensive experimental study.

The rest of this paper is organized as follows. Sec. 2 presents literature surveys of self-supervised learning and self distillation. Sec. 3 presents the proposed self-supervised self distillation approach. Evaluation and ablation studies will be provided in Sec. 4, followed by the concluding remarks in Sec. 5.

2. Related Works

2.1. Self-Supervised Learning

Self-supervised learning is a paradigm for unsupervised learning. The main concept is to exploit information freely available besides or within label-free data to implicitly guide model learning, so that the model learns general-purpose features. Some typical ways to discover the weak supervisory information includes predicting the position of an image patch [6], predicting image rotation [8], and completing a jisaw puzzle [18].

Contrastive learning is a widely adopted formulation to do self-supervised learning. It encourages learning similiar representations for the data augmented from the same input (positive pairs), and learning distinct representations for the data augmented from different inputs (negative pairs). How to form positive/negative samples and how distributions of them influence learning is a key component. Wu et al. [21] formulated an instance discrimination task to facilitate learning features to be discriminative of individual instances. Chen et al. [3] showed that composition of multiple data augmentation operations is crucial in contrastive learning. He et al. [11] formulated contrastive learning as dictionary look-up, and the dictionary is dynamically built with a queue and a momentum encoder. By combining two designs of SimCLR [3] into the MoCo strategy [11], the MoCo v2 [4] further gets performance gain. In this paper, we mainly take MoCo v2 to construct a self-supervised model. More literature surveys on self-supervised learning or contrastive learning can be found in [15] and [16].

2.2. Self Distillation

To avoid the requirement of a complex teacher model and reduce the cost of knowledge distillation, self distillation methods have been proposed since 2019. Based on the same data with different distortions, Xu and Liu [22] proposed to construct a single network to extract features. This network is split into two branches to construct two classifiers, and these classifiers were guided to output similar posterior distributions for the same data with different distortions. Zhang et al. [27] proposed self distillation based on a ResNet framework. The deepest residual block serves as the teacher to distill knowledge to shallower residual blocks, which are viewed as students. Shallow and deep classifiers are constructed based on outputs of different residual blocks, and their ensemble result can provide promising performance. This "Be Your Own Teacher" (BYOT) framework largely reduces training overhead required by a complicated teacher model. In their follow-up work [26], the classifiers are attached with attention modules to boost performance, and a dynamic inference mechanism was proposed to better make ensemble results. In this paper, our SSSD is mainly based on the BYOT framework.

The concept of self distillation has been adopted in deep metric learning [20], graph neural network [5], and word segmentation [13]. Gong et al. [9] proposed to combine mutual information and self information to increase expressivity of extracted features and thus improve self distillation. Although the effectiveness of self distillation has been empirically observed, why it works still needs investigated. Mobahi et al. [17] provides the first theoretical analysis of self-distillation. In [2], self distillation is shown as implicitly combining ensemble and knowledge distillation to improve test accuracy. More surveys on knowledge distillation and variations of self distillation can be found in [10].

3. SSSD: Self-Supervised Self Distillation

Fig. 1 shows the overview of the proposed SSSD. The top part shows the self-supervised learning model pre-trained based on an unlabeled dataset D_{pre}. This SSL model is used to extract features from the target dataset D_{tar}, which are then clustered to generate a pseudo label to each cluster. The bottom part shows that we utilize the pseudo labels to guide the learning of the teacher model (the deepest block) and self distillation.

3.1. Self-Supervised Learning

Based on an unlabeled dataset D_{pre}, we adopt the MoCo v2 mechanism [11][4] to train a model $\mathcal{M}_{ssl}$ in a self-supervised way. Specifically in the pretext task, the ResNet-18 or ResNet-50 [12] are taken as the backbone networks of $\mathcal{M}_{ssl}$. Random crop, random color jittering, random horizontal flip, and random grayscale conversion are used for data augmentation. We follow the instance discrimination task [21] where a query image matches a key image if they are (augmented) from the same image.

Given the target dataset D_{tar}, the pre-trained model $\mathcal{M}_{ssl}$ extracts features from it, and then the K-means clustering algorithm is adopted to cluster these features. After clustering, data points clustered to the same cluster are labeled as the same pseudo label. These pseudo labels provide weak supervisory signals to the following self distillation process. Notice that the dataset D_{pre} for SSL and the target dataset D_{tar} for SD could be the same or different.

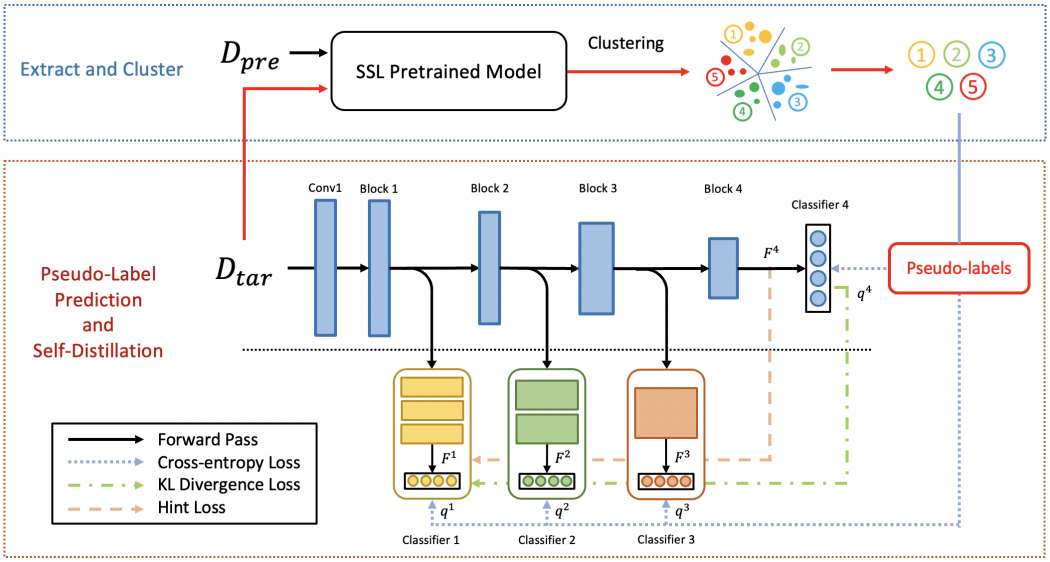

Figure 1. Overview of the proposed self-supervised self distillation.

3.2. Self Distillation

3.2.1 Network Architecture

The bottom part of Fig. 1 shows the network architecture of the self distillation approach [27]. Starting from a backbone network, based on the target dataset D_{tar}, this model $\mathcal{M}_{sd}$ takes pseudo labels as supervisory signals to conduct self distillation. In the following, we mainly implement the self distillation method by taking ResNet-18 [12] as the backbone. This network can be divided into four sections according to residual blocks. The deepest section (the 4th section) is viewed as the teacher of the shallower sections (the 1st to the 3rd sections).

To proceed self distillation, the output of each section is connected by a sequence of bottleneck layers, followed by a fully-connected layer and a softmax layer, so that each section can be viewed as a classifier. The 1st to the 3rd classifiers are constructed based on the knowledge distilled from the 4th classifier. The distillation is guided from three perspectives.

- The classification results of student classifiers should be similar to the pseudo labels. Quantitively, cross entropy between predicted labels and pseudo labels is calculated. The weakly-supervised knowledge hidden in pseudo labels are implicitly utilized by shallow sections.
- The classification results of student classifiers should be similar to that of the teacher classifier. Specifically, the KL divergence of softmax outputs between students and the teacher is calculated. The KL divergence measures similarity between the output distributions of the teacher classifier and student classifiers.
- Different sections make their classification based on different levels of feature maps, which represent the same image from different perspectives. Implicit knowledge of the deepest feature maps can be introduced to improve feature extraction in shallow sections. The L2 losses between feature maps of the deepest section and each shallow section are thus calculated.

3.2.2 Loss Functions

Here we formally define the loss functions mentioned above. Let $\Theta = \{\theta_{i/C}\}_{i=1}^{C}$ denote the classifiers in $\mathcal{M}_{sd}$, which is divided into C sections, and thus conceptually C classifiers are included. The softmax output of the ith classifier is denoted as q^i, $i = 1, 2, 3, 4$. The softmax output of the deepest classifier is especially denoted as q^C, i.e., $q^4 = q^C$. Given an input image x, the network $\mathcal{M}_{sd}$ finally outputs the predicted label.

The first item mentioned in Sec. 3.2.1 is defined as the summation of cross entropy between the pseudo labels $\tilde{\boldsymbol{y}}$ (obtained by $\mathcal{M}_{ssl}$) and the softmax outputs of classifiers:

$$\mathcal{L}_c = \sum_{i=1}^{3} CE(q^i, \tilde{\boldsymbol{y}}), \tag{1}$$

where $CE(\cdot)$ denotes cross entropy.

The second term is defined as the summation of the KL divergence between the softmax output of the Cth classifier and each shallow classifier:

$$\mathcal{L}_k = \sum_{i=1}^{3} KL(q^i, q^C), \tag{2}$$

where $KL(\cdot)$ denotes KL divergence. Notice that the cross entropy in eqn. (1) is calculated between the softmax output and the pseudo labels, while the KL divergence in eqn. (2) is calculated between softmax outputs of the teacher classifier and student classifiers.

The third term is defined as the summation of L2 distances between the deepest section and each shallow section:

$$\mathcal{L}_\ell = \sum_{i=1}^{3} \|F_i - F_C\|_2^2, \tag{3}$$

where F_i and F_C denote features (output by the bottleneck layers) fed to the classifier θ_i and θ_C, respectively.

Overall, these three losses are combined as:

$$\mathcal{L}_{all} = CE(q^C, \tilde{\boldsymbol{y}}) + (1-\alpha)\mathcal{L}_c + \alpha\mathcal{L}_k + \lambda\mathcal{L}_\ell, \tag{4}$$

where α and λ are balanced parameters set as 0.3 and 0.003, respectively. The first term $CE(q^C, \tilde{\boldsymbol{y}})$ is the cross entropy between the softmax ouput of the teacher classifier and the pseudo labels. This term is weighted more as compared to losses from student classifiers.

4. Experiments

4.1. Performance Evaluation

Taking CIFAR-10, CIFAR-100, and ImageNet-1K datasets as the main bases, we first verify that SSL aids SD. In the following, the pre-trained dataset D_{pre} is the same as the target dataset D_{tar} if not specially noted. We adopt ResNet-18 or ResNet-50 as the backbones to construct the SSL model $\mathcal{M}_{ssl}$, in order to evaluate how different SSL models influence performance of SD. We also try the pre-trained models trained for different numbers of epochs. Conceptually training more epochs makes the SSL model learn more knowledge. The backbone of SD is keeping as ResNet-18. Visual features are extracted by $\mathcal{M}_{ssl}$ and then clustered into K classes to get pseudo labels. The self distillation process then learns student classifiers and a teacher classifier.

Table 1 shows the classification accuracy of the SD models assisted by SSL, on the CIFAR-10, CIFAR-100, and ImageNet-1K datasets, respectively. The number of clusters for generating pseudo labels is set as 1000, 500, and 5000 for the CIFAR-10, the CIFAR-100, and the ImageNet-1K datasets. In addition, we intentionally implement a random labeling approach as the baselines, where we randomly assign one of 1000/500/5000 labels to each data point. Based on such random labels, we train the SD model and get the classification accuracy shown in the random label rows.

From Table 1 three observations can be made. First, a well-trained SSL model really aids SD, compared to the random labeling approach. This shows the value of the proposed SSSD, where no data labels are available. Second, an SSL model trained for more epochs really provides more accurate pseudo labels and richer knowledge to the SD model, so that accuracy of the corresponding counterparts is higher. Third, a more complex $\mathcal{M}_{ssl}$ (ResNet-50) provides richer knowledge than a simpler $\mathcal{M}_{ssl}$ (ResNet-18) to the SD model.

4.2. Performance Comparison

4.2.1 Distillation Schemes

We compare the proposed method with SEED [7] and ClusterFit [23]. In SEED [7], a larger teacher network is leveraged to transfer its knowledge into a smaller student network in a self-supervised manner. Without requiring labeled data, the teacher network is pre-trained by a self-supervised learning process. Given test samples, the student network is guided to output a score distribution similar to the one output by the teacher network, based on losses commonly used in knowledge distillation. The target of SEED is to boost performance of small networks in the self-supervised learning scheme. In ClusterFit [23], a network is pre-trained to learn visual representations of an image dataset. It then clusters the images based on the extracted representations and generates pseudo labels. Next, a new network is trained from scratch based on the pseudo labels, which is then used to complete the downstream task. The target of ClusterFit is to reduce overfit of the pre-trained network and improve robustness of the learnt visual representations.

Our method seems to be similar to the processes of SEED and ClusterFit, but we need to emphasize the difference as follows. SEED clearly has the framework of a teacher and a student, and works in the standard procedure of knowledge distillation. Our pre-trained network merely provides pseudo labels, and we only have one model (as the teacher and the student at the same time) to self-distillate itself. In ClusterFit, the pre-trained network provides pseudo labels, which is the same as ours. But in our work, the network for downstream tasks is a self-distillation model (BYOT [27]). Overall, SEED adopts distillation to improve self-supervised learning in small models, while our work adopts self-supervised learning to enable unsupervised self distillation. ClusterFit utilizes the concept of clustering (and thus pseudo labels) to avoid learning overfitted representations, while our work adopts the concept of clustering to guide self distillation.

Table 2 shows the comparison of classification accuracy on the ImageNet-1K dataset. The pre-trained models are ResNet-50 or ResNet-101 trained for 200 epochs. The student models used in SEED and ClusterFit, and the backbone of the proposed SSSD are all ResNet-18, no matter the pre-trained models are ResNet-50 or ResNet-101. As can be seen in this table, the proposed SSSD slightly outperforms

Table 1. Classification accuracy of the SD model on the CIFAR-10, CIFAR-100, and ImageNet-1K datasets.

CIFAR-10						
$\mathcal{M}_{ssl}$	Epochs	Classifier 1	Classifier 2	Classifier 3	Classifier 4	Ensemble
Random labels	–	42.34	42.10	41.92	44.03	45.99
ResNet-18	200	78.54	79.47	79.71	80.00	82.05
	800	81.96	83.32	83.96	84.10	85.75
$\mathcal{M}_{ssl}$	Epochs	Classifier 1	Classifier 2	Classifier 3	Classifier 4	Ensemble
ResNet-50	200	78.93	79.75	80.08	80.22	82.20
	800	**83.15**	**84.72**	**85.81**	**85.94**	**87.36**
CIFAR-100						
$\mathcal{M}_{ssl}$	Epochs	Classifier 1	Classifier 2	Classifier 3	Classifier 4	Ensemble
Random labels	–	18.05	17.88	18.66	21.06	21.34
ResNet-18	200	52.78	53.49	53.51	53.24	57.01
	800	54.44	55.72	56.24	55.85	59.71
$\mathcal{M}_{ssl}$	Epochs	Classifier 1	Classifier 2	Classifier 3	Classifier 4	Ensemble
ResNet-50	200	51.33	52.17	52.08	52.03	55.76
	800	**57.37**	**58.90**	**60.54**	**60.25**	**63.60**
ImageNet-1K						
$\mathcal{M}_{ssl}$	Epochs	Classifier 1	Classifier 2	Classifier 3	Classifier 4	Ensemble
Random labels	–	12.07	12.25	13.19	16.83	16.24
ResNet-18	200	44.19	45.68	48.53	51.95	52.35
	800	45.86	48.05	51.38	54.82	54.82
$\mathcal{M}_{ssl}$	Epochs	Classifier 1	Classifier 2	Classifier 3	Classifier 4	Ensemble
ResNet-50	200	48.41	50.97	55.24	58.45	58.11
	800	**50.23**	**53.74**	**58.63**	**61.86**	**61.12**

Table 2. Performance comparison of different distillation schemes on the ImageNet-1K datasets.

Pre-trained models	Methods	Accuracy
ResNet-50	ClusterFit [23]	56.55
	SEED [7]	57.90
	SSSD (our)	58.45
ResNet-101	ClusterFit [23]	59.02
	SEED [7]	58.90
	SSSD (our)	60.46

SEED and ClusterFit. This shows that the proposed self-supervised self distillation is not only feasible, but also effective in the standard downstream task.

4.2.2 Pseudo Labels

Another work very similar to our work is DACSD [1]. In DACSD, a ResNet-18 is jointly used to be the pre-trained model that learns and clusters visual representations, as well as the backbone for self distillation. As the training proceeds, the ResNet-18 gradually learns better visual representations and generates better pseudo labels. These pseudo labels reversely guide learning of the ResNet-18 through back propagation. The main difference between ours and DACSD is that, in our framework, we complete training of the pre-trained model first and thus can generate relatively good pseudo labels at the beginning of the self distillation process. On the other hand, DACSD generates relatively weak pseudo labels at the beginning of the self distillation process.

Table 3 shows performance comparison of pseudo label generation schemes on the ImageNet-1K dataset. The backbone of the self distillation models in both DACSD and ours are ResNet-18. The self-supervised model to give our results in Table 3 is ResNet-50 ($\mathcal{M}_{ssl}$). As can be seen, performance of ours outperforms the counterpart of DACSD based on this experimental setting.

4.3. Semi-Supervised Evaluation

To evaluate effectiveness of the representations learnt by the self distillation model, we connect a fully-connected (FC) layer to the last residual block of the SD model after distillation, and train the FC layers based on the labeled ImageNet-1K dataset. Following the settings in [25] and [3], we intentionally subsample only 1%, 10%, 20%, ..., 50% of the labeled ImageNet-1K dataset and use them to fine-tune the residual blocks as well as FC layers on the labeled data without regularization. This is to simulate the semi-supervised learning scheme. If the learnt visual representations are robust, only few labeled data are needed to

Table 3. Performance comparison of pseudo label generation schemes on the ImageNet-1K datasets.

Backbone of $\mathcal{M}_{sd}$	Methods	Classifier 1	Classifier 2	Classifier 3	Classifier 4	Ensemble
ResNet-18	DACSD [1]	45.59	48.14	51.63	49.77	53.62
	SSSD (our)	48.41	50.97	55.24	58.45	58.11

Table 4. Classification accuracy obtained based on different settings of semi-supervised learning on the ImageNet-1K dataset.

Pre-trained models	1%	10%	20%	30%	40%	50%
ResNet-50, 200 epochs	40.16	57.41	61.18	63.19	64.54	65.53
ResNet-101, 200 epochs	42.58	58.48	62.05	64.09	65.36	65.94

Table 5. Performance of object detection by the Faster R-CNN with the backbones replaced by different $\mathcal{M}_{sd}$'s.

Methods	Pre-trained models	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
SEED [7]	ResNet-50, 200 ep	46.1	74.8	49.1
	ResNet-101, 200 ep	46.8	75.8	49.3
SSSD (our)	ResNet-50, 200 ep	53.3	79.2	58.4
	ResNet-101, 200 ep	53.4	79.3	58.5

construct robust classifiers.

Table 4 shows classification accuracy obtained based on different settings of semi-supervised learning on the ImageNet-1K datasets. When the pre-trained model is ResNet-50 trained for 200 epochs, the classification accuracy significantly boosts from 40.16 to 57.41 when the fraction of training data increases from 1% to 10%. This shows that the representations learnt by the proposed SSSD are effective because similar performance can be obtained even if only 10% of training data are utilized. More training data can help improve performance, but we found the improvement gets saturated when more than 40% of the training data are sampled. A similar trend can be seen when the pre-trained model is ResNet-101 trained for 200 epochs.

4.4. Object Detection

In addition to image classification, here we want to verify that the representations after distillation can be transferred to other tasks like object detection. Specifically, we replace the backbone of a Faster R-CNN [19] by the ResNet-18 $\mathcal{M}_{sd}$ after self distillation. It is then fine-tuned on the VOC-07+12 train+val set and evaluated on VOC-07 test split. The model $\mathcal{M}_{sd}$ used in this experiment is obtained by self distillation guided by the 16000 pseudo labels, which are generated by the SSL model $\mathcal{M}_{ssl}$ pre-trained based on the ImageNet-1K dataset (with data labels).

Table 5 shows performance of object detection in terms of bounding box average precisions AP^{bb}, AP^{bb}_{50}, and AP^{bb}_{75}. As can be seen, the proposed SSSD and its learnt model $\mathcal{M}_{sd}$ can be effective backbone for Faster R-CNN. Comparing with the similar experiments provided in SEED [7], the performance superiority is clear.

4.5. Different Pre-trained Datasets

The experiments shown from Table 1 to Table 5 are conducted by setting the dataset for SSL the same as that for SD, i.e., $D_{pre} = D_{tar}$. Here we further evaluate how performance changes if $D_{pre} \neq D_{tar}$. Specifically, we compare the setting of D_{pre} = ImageNet1K, D_{tar} = CIFAR, with D_{pre} = D_{tar} = CIFAR. Due to different image sizes in ImageNet1K and CIFAR, we resize the images of ImageNet1K into 32×32 and train the $\mathcal{M}_{ssl}$ model for 200 epochs.

Table 6 shows performance variations when the pseudo labels are generated by pre-trained models learnt based on different D_{pre}'s. As can be seen, for both D_{tar} = CIFAR-10 and D_{tar} = CIFAR100, using ImageNet1K as D_{pre} can yield better performances than using CIFAR10 and CIFAR-100 as D_{pre}. This shows that pseudo labels generated by models trained based on a larger-scale dataset can provide richer information than that trained based on a small-scale dataset. Our proposed SSSD can utilize better pseudo labels to learn better representations for SD models.

5. Conclusion

We have presented using self-supervised learning to enable unsupervised self distillation. A model is first pre-trained based on a dataset D_{pre} to complete the instance discrimination task in a self-supervised learning scheme. This pre-trained model is viewed to have the ability to extract satisfactory visual representations for general purpose. It is used to extract visual features from the target dataset D_{tar}, and then the K-means clustering algorithm is adopted to cluster features and thus generate pseudo labels. These pseudo labels guide the learning of a self distillation model, which learns effective representations for D_{tar} by self learning from the deepest part of the network, without requiring a teacher model trained in the supervised way. Therefore, the whole framework does not need labeled data, and the proposed self-supervised self distillation breaks the limitation of previous supervised self distillation. We verify the effectiveness of SSSD by evaluating on the CIFAR-10, CIFAR-100, and ImageNet-1K datasets. Performance superior to similar frameworks is shown.

Acknowledgement. This work was funded in part by

Table 6. Classification accuracy on CIFAR-10 and CIFAR-100 using pseudo labels generated by pre-trained models learnt based on different D_{pre}'s.

D_{tar} = CIFAR-10							
D_{pre}	M_{ssl}	Epochs	Classifier 1	Classifier 2	Classifier 3	Classifier 4	Ensemble
CIFAR-10	R-18	200	78.54	79.47	79.71	80.00	82.05
	R-50		78.93	79.75	80.08	80.22	82.20
ImageNet-1K	R-18	200	80.53	82.21	83.44	83.47	84.97
	R-50		83.19	85.05	86.58	86.54	87.60
D_{tar} = CIFAR-100							
D_{pre}	M_{ssl}	Epochs	Classifier 1	Classifier 2	Classifier 3	Classifier 4	Ensemble
CIFAR-100	R-18	200	52.78	53.50	53.51	53.24	57.01
	R-50		51.33	52.17	52.08	52.03	55.76
ImageNet-1K	R-18	200	53.88	55.90	57.32	57.21	60.76
	R-50		57.35	59.85	61.14	60.83	64.12

Qualcomm through a Taiwan University Research Collaboration Project and in part by the National Science and Technology Council, Taiwan, under grants 111-3114-8-006-002, 110-2221-E-006-127-MY3, 108-2221-E-006-227-MY3, 107-2923-E-006-009-MY3, and 110-2634-F-006-022.

References

[1] Mohammed Adnan, Yani A. Ioannou, Chuan-Yung Tsai, and Graham W. Taylor. Domain-agnostic clustering with self-distillation. In *Proceedings of NeurIPS Workshop on Self-Supervised Learning: Theory and Practice*, 2021.

[2] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. In *arXiv:2012.09816*, 2021.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of International Conference on Machine Learning*, 2020.

[4] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. In *arXiv:2003.04297*, 2020.

[5] Yuzhao Chen, Yatao Bian, Xi Xiao, Yu Rong, Tingyang Xu, and Junzhou Huang. On self-distilling graph neural network. In *Proceedings of International Joint Conference on Artificial Intelligence*, 2021.

[6] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of International Conference on Computer Vision*, 2015.

[7] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. Seed: Self-supervised distillation for visual representation. In *Proceedings of International Conference on Learning Representations*, 2021.

[8] Zeyu Feng, Chang Xu, and Dacheng Tao. Self-supervised representation learning by rotation feature decoupling. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2019.

[9] Yu Gong, Ye Yu, Gaurav Mittal, Greg Mori, and Mei Chen. Muse: Feature self-distillation with mutual information and self-information. In *Proceedings of British Machine Vision Conference*, 2021.

[10] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021.

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2020.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2016.

[13] Rian He, Shubin Cai, Zhong Ming, and Jialei Zhang. Weighted self distillation for chinese word segmentation. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, 2022.

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Proceedings of NIPS Deep Learning Workshop*, 2014.

[15] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4037–4058, 2021.

[16] Phuc H. Le-Khac, Graham Healy, and Alan F. Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934, 2020.

[17] Hossein Mobahi, Mehrdad Farajtabar, and Peter L. Bartlett. Self-distillation amplifies regularization in hilbert space. In *Proceedings of Conference on Neural Information Processing Systems*, 2020.

[18] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proceedings of European Conference on Computer Vision*, 2016.

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of Advances in Neural Information Processing Systems*, 2015.

[20] Karsten Roth, Timo Milbich, Bjorn Ommer, Joseph Paul Cohen, and Marzyeh Ghassemi. Simultaneous similarity-based

self-distillation for deep metric learning. In *Proceedings of International Conference on Machine Learning*, 2021.

[21] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2018.

[22] Ting-Bing Xu and Cheng-Lin Liu. Data-distortion guided self-distillation for deep neural networks. In *Proceedings of AAAI Conference on Artificial Intelligence*, 2019.

[23] Xueting Yan, Ishan Misra, Abhinav Gupta, Deepti Ghadiyaram, and Dhruv Mahajan. Clusterfit: Improving generalization of visual representations. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2020.

[24] Chenglin Yang, Lingxi Xie, Chi Su, and Alan L. Yuille. Snapshot distillation: Teacher-student optimization in one generation. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2019.

[25] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of IEEE International Conference on Computer Vision*, 2019.

[26] Linfeng Zhang, Chenglong Bao, and Kaisheng Ma. Self-distillation: Towards efficient and compact neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[27] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of IEEE International Conference on Computer Vision*, 2019.