Student-Teacher Oneness: A Storage-efficient approach that improves facial expression recognition

Zhenzhu Zheng, Christopher Rasmussen, Xi Peng
University of Delaware
{zzzheng, ras, xipeng}@udel.edu

Abstract

We present Student-Teacher Oneness (STO), a simple but effective approach for online knowledge distillation improves facial expression recognition, without introducing any extra model parameters. Stochastic sub-networks are designed to replace the multi-branch architecture component in current online distillation methods. This leads to a simplified architecture, and yet competitive performances. Under the "teacher-student" framework, we construct both teacher and student within the same target network. Student network is the sub-networks which randomly skipping some portions of the full (target) network. The teacher network is the full network, can be considered as the ensemble of all possible student networks. The training process is performed in a closed-loop: (1) Forward prediction contains two passes that generate student and teacher predictions. (2) Backward distillation allows knowledge transfer from the teacher back to students. Comprehensive evaluations show that STO improves the generalization ability of a variety of deep neural networks to a significant margin. The results prove our superior performance in facial expression recognition task on FER-2013 and RAF.

1. Introduction

Learning a good representation is important for facial expression recognition. Although deep neural networks have achieved great success in computer vision tasks such as image classification [31, 73, 66], object detection [51, 45, 6], segmentation [20, 15], human pose estimation [2] and person re-identification [22, 47]. But deep neural networks are often over-parameterized, which makes it not suitable for deployment, and easily suffering from over-fitting. To address this issue, one popular paradigm is Knowledge Distillation (KD), aiming at training *small* and *generalizable* models. The general idea is to transfer knowledge from a teacher (large) model to a student (small) model, where the student is trained to match the output of the teacher

Figure 1: Facial Expression Recognition is challenging, learning a generalizable feature representation is crucial.

[25, 46, 72]. However, classical knowledge distillation [25] relies on a pre-trained teacher, which might not always be available in practice. To solve this problem, *online* [48, 77, 78, 7] and *self*-distillation [16, 64, 76, 70] methods are proposed using different strategies. *Self*-distillation approaches [16, 64, 76, 70] typically take advantage of the model generations during the training trajectory [16, 64] or the intermediate flow within the network [76]. However, many approaches on this line come with a complex workflow or architecture design.

Online distillation [77, 78, 7], on the other hand, intends to build a strong teacher role by a group of (student) peers, which are typically constructed via a multi-branch architecture. However, the multi-branch architecture design has drawbacks: First, the number of branches (students) would be limited subject to the available storage. This is a *storage-heavy* consumption approach for training. Second, and more importantly, due to the limited number of branches, the model would not have sufficient power to cover a large degree of uncertainty/variety in the solution space.

We aim for a storage-efficient training scheme while maintaining competitive performance. To create student models without sourcing a multi-branch architecture, we propose to generate student (peers) within the same net-

work, which allow us to achieve a sufficient amount of student diversity, while without introducing any extra model parameters.

The teacher is the full network, while the students are the sampled sub-networks. Both the teacher and students share weights since they are inside the same network. The teacher can be considered as the *implicit* ensemble of all students. The analogy is that the students and teacher are *Oneness*, where students are the smaller individual and together form a more powerful larger collection. Individual (student) absorbs knowledge from the collection (teacher), and the teacher grows out from students.

The student network is sampled by randomly skipping some portions of the full network during the forward pass. In this case, there can be exponentially many student networks to be generated. By exploiting the dynamic architecture within the network, a certain degree of diversity can be achieved. This is different from approaches based on multi-branch [78, 7], where student diversity is limited to the static branching structure. To gain better performance, however, they require extra components such as gating or attention.

Inspired by [78], the whole training process is performed in a closed-loop: *forward prediction* and *backward distillation*. The **forward prediction** contains two passes: (1) one pass goes through the full network to generate the teacher prediction; (2) another pass goes through a randomly sampled sub-network to generate the student prediction. The **backward distillation** aims to transfer knowledge from the teacher to all students, which is the teacher itself.

The whole process can be considered as seamlessly incorporating distillation as a regularization into the training procedure.

Our contributions are summarized as follows:

- We tackle the online knowledge distillation problem from a **new aspect**: to achieve student model *diversity* within the target network, without sourcing a multi-branch architecture.
- Comprehensive experiments and ablation studies demonstrate the effectiveness of our proposed method, which improves the *generalization* performance of a variety of deep neural networks.
- Comprehensive evaluations and ablation studies prove our superior performance in facial expression recognition task.

2. Related Works

2.1. Knowledge Distillation

Knowledge Distillation (KD) originated from [4], popularized by [25], now become is a hot research topic [18, 55] applied in many areas [61, 62, 14]. The key problem is how to transfer the knowledge from a large teacher model to a small student model. It contains two major components: **knowledge** and **distillation scheme**.

Knowledge. Depending on what information that the student model try to mimic from the teacher model, KD methods can be broadly categorized into three categories [18]: (1) **Response-based** knowledge refers to the final prediction of the teacher model. It is simple yet effective, and has been widely used in different tasks [9, 75, 37] and applications [50, 26]. The most popular form is also known as soft target [25, 1], which can be considered as label smoothing or regularization [30, 38, 13]. Our approach belongs to this category. (2) **Feature-based** knowledge is an extension of the response-based, which considered both the output of the last layer and the output of intermediate layers [46, 72, 29, 24, 43, 10, 56, 12, 23]. (3) **Relation-based** knowledge further explores the relationships between different layers [65, 74, 32, 41, 11, 35, 8] or data samples [35, 39, 40, 42, 53, 44].

Distillation Schemes. The distillation schemes can be directly divided into three main categories: **offline distillation**, **online distillation** and **self-distillation**.

While **offline** distillation requires a pre-trained teacher model, online and self-distillations aims to fulfill the absence of the teacher role from different aspects. Typically, **self**-distillation approaches take advantage of generation in the training trajectory [16, 64], the information flow within the network [76] or class information [70]. However, many approaches on this line come with a complex workflow or architecture design.

Online distillation [48, 77, 78, 7, 19, 59] allows both the teacher and student(s) study together from each other. The basic idea is to simultaneously training a group of student models by learning from peers predictions as an effective substitute for the static pre-trained Teacher. However, there are drawbacks. First, online ensemble KD simply aggregate students logits to form an ensemble teacher restrains the diversity of student peers, thus limiting the effectiveness of online learning learning. Second, existing approaches adopt a multi-branch architecture leading to storage-heavy consumption and also not flexible for ensemble in a more versatile or dynamic way. Our approach falls into this category. Different from traditional online distillation methods, we intends to generate **diversity** within the network instead of any auxiliary branches, leading to a *storage-efficient* solution.

Concurrent with our work, Mean Teacher [52] also construct the teacher model without extra parameters by using the average model weights of the training epochs. The difference is that [52] focus on the semi-supervised learning,

while ours belongs to supervised learning and explore on the architecture aspect.

2.2. Implicit Ensemble

An alternative to traditional ensembles, so-called "implicit" ensembles have high efficiency during both training and testing. From the *architecture* perspective, Dropout [49], DropConnect [54] and Stochastic Depth [27] can be considered as sampling sub-networks at different levels. Dropout [49] creates an ensemble out of a single model by "dropping" random sets of hidden nodes during each mini-batch. DropConnect [54] and Stochastic Depth [27] can be considered as specific cases of Dropout operating on the edge and layer level, respectively. In this work, we take advantage of implicit ensemble to generate student networks. This is different from one-shot architecture search [3] where the sub-network weights are dynamically generated, which requires a more complicated process.

Dropout Distillation [5] proposed to better approximate the averaging process for prediction in the original dropout. Different from [5], we leverage a "dropout-based" method [27] as a means to generate student/teacher networks achieving in-network knowledge transfer.

2.3. Adaptive Computation

Skipping layers. Stochastic Depth [27] can be considered as random layer-wise dropout in training. This idea can be extended to inference [58, 60].

Skipping channels. Slimmable networks [68, 67] proposed switchable batch normalization to dynamically adjust the channels for accuracy-efficiency trade-offs at *inference* time.

Concurrent with our work, "inplace distillation" [67] proposed knowledge transfer from full network to sub-networks in place. Despite conceptually similar, our problem, goal, method and strategy are different: The sub-networks in [67] operate on various different *width* for accuracy-efficiency trade-offs at *inference* time via adapting post-statistics of Batch Normalization [28]. On the contrary, we use sub-networks with different *depth* during *training* towards better generalization performance via increasing student model diversity.

2.4. NAS with Knowledge Distillation

Neural Architecture Search (NAS), aiming at automatically designing network architectures by machines. Recent work [33] distills the neural architecture knowledge from a teacher model to improve the effectiveness of NAS. [36] distill the teacher's knowledge into both the parameters and architecture of the student. Our approach is different from this line of research in that *no search involved*, instead, the sub-networks in our approach are generated by randomly skipped connections, yielding a simple but effective solution.

3. Student-Teacher Oneness

In this section, we introduce a specific solution that use Stochastic Depth [27] to generate the sub-sample networks during training. More discussions please see Sec. 5.

We formulate an online distillation training method based on the idea of constructing both teacher and student networks via implicit ensemble within the same target network. In another word, the network generates both teacher and student predictions.

For model training, we often have access to n labelled training samples $D = (x_i, y_i)_i^n$ with each belonging to one of C classes $y_i \in Y = \{1, 2, ..., C\}$. The network parameter outputs a probabilistic class posterior $p(c|x, \theta)$ for a sample x over a class c:

$$p(c|x,\theta) = f_{sm}(z) = \frac{exp(z^c)}{\sum_{j=1}^{C} exp(z^j)}, c \in Y \quad (1)$$

where z is the logits or unnormalized log probability outputted by the network θ. To train a multi-class classification model, we typically adopt the Cross-Entropy (CE) measurement between the predicted and ground-truth label distribution as the objective loss function:

$$L_{ce} = -\sum_{c=1}^{C} \delta_{c,y} log(p(c|x,\theta)) \quad (2)$$

where $\delta_{c,e}$ is Dirac delta which returns 1 if c is the ground-truth label, and 0 otherwise. With the CE loss, the network is trained to predict the correct class label in a principle of maximum likelihood.

Overview. An overview of our approach is depicted in Fig. 2. The training contains two phases:

In the **forward** phase, the teacher and student predictions are generated in two separate forward passes. The Student prediction is generated by the output of a sample sub-network from the full network. The teacher prediction is obtained by the input go through the full network and weighted by sample ("survival") probability for each block. It can be considered as an *approximate* ensemble of all Student predictions.

In the **backward** phase, knowledge distillation is performed to ensure all students get knowledge from the teacher. Distillation loss is used here to ensure the ensemble logit is as close to the Teacher as possible.

Our method is established based on the "collapsing version of multi-branches" design for model training with several *merits*: (1) Exponential number of Students can be gen-

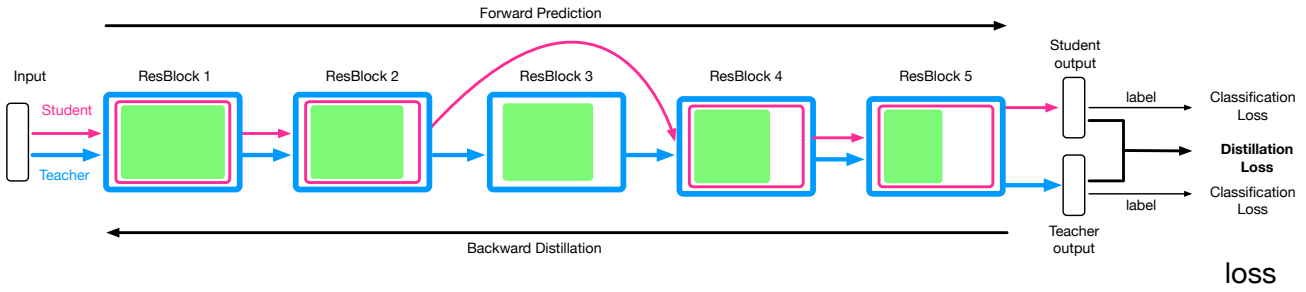


Figure 2: Overview. Teacher: full network. Student: sampled sub-network for each mini-batch. Other components are not shown for a simple illustration. Red: Student forward pass. Blue: Teacher forward pass. Green regions indicate the survival probability for the Residual Block (ResBlock), the larger the higher sample ("survival") probability is. Best view in color.

erated without additional storage expenses. (2) By the randomly sampling sub-network, each Student by itself has a stronger power against overfitting. It contributes to the generalization ability of the model as a whole.

Note that we do not use gating components or additional attention mechanisms as [77, 7] to further boost performance. This is because we can generate exponential many students and maintain sufficient diversity without additional computations.

Student Prediction. In Residual networks [21], a ResBlock output is as follows:

$$H_l = \text{ReLU}(f_l(H_{l-1}) + H_{l-1}) \tag{3}$$

where H_l denotes the output of the l^{th} layer, $f_l(\cdot)$ represents a typical convolutional transformation from layer $l-1$ to l. And we assume a ReLU activation function.

We use Stochastic Depth [27] to sample sub-networks via randomly dropping entire ResBlocks and bypassing their transformations through skip connections. Let $b_l \in \{0, 1\}$ denote a Bernoulli random variable indicating whether the l^{th} ResBlock is active ($b_l = 1$) or not ($b_l = 0$). The sample ("survival") probability of the l^{th} ResBlock is denoted as $p_l = Pr(b_l = 1)$.

Based on Eq. 3, the update rule of ResBlock in a student network can be formed as

$$H_l^s = \text{ReLU}(b_l f_l(H_{l-1}^s) + H_{l-1}^s) \tag{4}$$

where superscript s indicates student network. If $b_l = 1$, Eq. 4 functions as a ResBlock. If $b_l = 0$, the ResBlock reduces to a skip connection.

Teacher Prediction. The update rule of ResBlock in a teacher network is the combination of all possible student networks where each block is weighted by its survival probability. It can be considered an *approximate* ensemble of all sub-networks.

$$H_l^t = \text{ReLU}(p_l f_l(H_{l-1}^t) + H_{l-1}^t) \tag{5}$$

where superscript t indicates teacher network. "Survival" probability p_l is a hyper-parameter, we follow the preferred setting mentioned in [27] throughout.

Knowledge Distillation. Given the teacher's logits of each training sample, we distill this knowledge back to all students in a closed-loop form. For facilitating knowledge transfer, we computer soft probability distributions [25] at a temperature of T for the teacher and student respectively:

$$\tilde{p^s}(c|x, \theta^s) = \frac{exp(z_c^s/T)}{\sum_{j=1}^{C} exp(z_j^s/T)}, c \in Y \tag{6}$$

$$\tilde{p^t}(c|x, \theta^t) = \frac{exp(z_c^t/T)}{\sum_{j=1}^{C} exp(z_j^t/T)}, c \in Y \tag{7}$$

Higher values of T lead to more softened distributions.

To quantify the alignment between student and the teacher in their predictions, we use Kullback Leibler divergence from the student to the teacher written as:

$$\mathcal{L}_{kl} = \sum_{j=1}^{C} \tilde{p^t}(j|x, \theta^t) log \frac{\tilde{p^t}(j|x, \theta^t)}{\tilde{p^s}(j|x, \theta^s)} \tag{8}$$

Overall Loss Function. We obtain the overall loss function as the combination of classification loss and distillation loss:

$$\mathcal{L} = \mathcal{L}_{ce}^s + \mathcal{L}_{ce}^t + \lambda * T^2 * \mathcal{L}_{kl} \tag{9}$$

where classification loss is calculated by $\mathcal{L}_{ce}^s$ and $\mathcal{L}_{ce}^t$ which are the conventional CE loss terms associated with the Student and Teacher, respectively. $\mathcal{L}_{kl}$ indicates distillation loss. Following [78], the gradient magnitudes produced by the soft target $\tilde{p}$ are scaled by $\frac{1}{T^2}$, so we multiply the distillation loss term by a factor T^2 to ensure that the relative

Algorithm 1: Student-Teacher Oneness

Input: Labelled training data $\mathcal{D}$; Training epoch number τ;
Output: Trained model θ^t (Teacher network);
/* Training */
Initialization: $i = 1$; Randomly initialize θ^t; Assign survival probability p_l to each Residual Block.
while $i \leq \tau$ **do**
  for *each mini-batch* **do**
    Randomly sample a sub network $\theta^s \in \theta^t$ (Student network);
    Compute Student prediction. Eq. 4;
    Compute Teacher prediction. Eq. 5;
    Compute soft targets of Student and Teacher Eq. 6 and Eq. 7;
    Update full network parameter θ^t by SGD algorithm. Eq. 9.
  end
end
/* Testing */
Deployment: Use θ^t and Teacher prediction. Eq. 5.

contributions of ground-truth and teacher probability distributions remain roughly unchanged. λ is the trade-off between loss terms.

Connections with Stochastic Depth [27]. Stochastic Depth [27], a training strategy that in statistics approximately combines exponential numbers of sub-networks via dropping certain blocks during each forward pass. We borrow this idea by using the smaller sub-networks as students, while the ensemble naturally performs as teacher.

There are differences: Stochastic Depth [27]'s objective does not include distillation loss, and update weights of the sub-network only. Our approach adopts distillation loss and update weights of the full network instead.

Model Training and Deployment. Details for model training and deployment are summarized in Alg 1. Unlike traditional online distillation methods which build Student peers via auxiliary components, Ours construct both the student and teacher model within exactly the same network. The student model diversity can be achieved by varying sampled network architecture during each mini-batch. Thus there is no extra complexity for model architecture as that required by ONE [78, 7].

One difference from typical model training is that our approach has two forward passes[1], in order to generate student and teacher predictions. Once the model is trained, we can simply use the teacher prediction for deployment. And only one forward pass is needed as the model testing normally does.

4. Experiments

4.1. Facial Expression Recognition

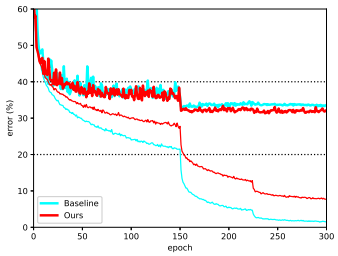
Datasets. We used two benchmarks facial expression datasets with 7 human facial expressions. (1) FER-2013 [17]: It consists of 28,709 gray-scale images for training and 3,589 for testing. (2) RAF [34] is a real-world facial expression recognition dataset, which contains 12,271 RGB images for training and 3,068 for testing. Here we use the basic 7 expression categories. Sample images are shown in Fig. 1.

Setup. Similar to CIFAR, we resize images to 32x32. For FER-2013, image channels are duplicated to make them RGB images. For all datasets, we adopted the same experimental settings as for making fair comparisons [27, 78]. We used the SGD with Nesterov momentum and set the momentum to 0.9 with weight decay 1e-4. Batch size is 128, training epoch is 300. We deployed a standard learning rate schedule that drops from 0.1 to 0.01 at 50 % training and to 0.001 at 75%. Following [25], we set T = 3 in all the experiments. Cross-validation of hyper-parameters[2] may give better performance but at the cost of extra model tuning. Trade-off parameter λ is set to be 0.25. We adopted the common top-1 classification error rate.

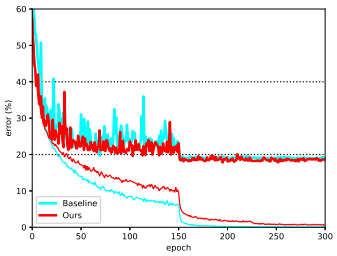
Results. (1) **Improve generalization ability.** Tab. 1 shows the classification results on FER-2013 and RAF on ResNets as the target networks with a variety of depths. It shows that our approach consistently improves the performances to a significant margin on depth 32, 50, and 110. We observe that the peak performance comes from depth 50, and degrade when depth increases to 110. With our approach, we are able to train ResNet-110 with 3.09% improvement, indicating a strong ability to prevent overfitting. Fig. 3 shows the training curves on ResNet-32 and ResNet-110 on FER-2013 and RAF, respectively. (2) **Storage-efficiency.** Tab. 2 show the comparison with state-of-the-art online distillation methods. It shows that our approach reaches on-par or even better performances without introducing extra parameters, while other methods typically use 2.5 to 3.25 times parameters, due to the multi-branch architecture. This indicates our approach is storage efficiency while maintaining competitive performance.

[1]Our approach has an approximately 1.24x training time of standard training, and this overhead seems to be acceptable in real-world practice.

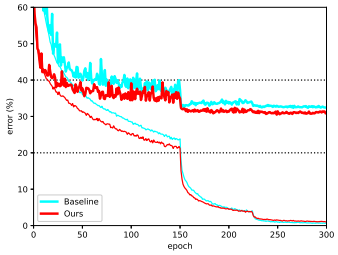
[2]For student network generation, we follow the same hyper-parameter setting as [27].



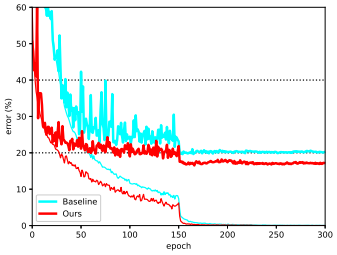
(a) ResNet-32 on FER-2013



(b) ResNet-32 on RAF



(c) ResNet-110 on FER-2013



(d) ResNet-110 on RAF

Figure 3: Training and testing error for ResNet-32, 110 on FER-2013 and RAF, repectively. Thin lines indicate training error, **bold** lines indicate **test** error.

Method	FER-2013	RAF	Params
ResNet-32 [21]	32.85	18.55	0.46M
ResNet-32 + **Ours**	**31.33**	**17.80**	0.46M
	(+1.52)	(+0.75)	
ResNet-50 [21]	31.83	17.67	0.76M
ResNet-50 + **Ours**	**30.38**	**16.43**	0.76M
	(+1.45)	(+1.24)	
ResNet-110 [21]	32.35	19.62	1.15M
ResNet-110 + **Ours**	**30.48**	**16.53**	1.15M
	(+1.87)	(+3.09)	
WRN-20-2 [71]	31.72	17.54	1.08M
WRN-20-2 + **Ours**	**31.27**	**17.33**	1.08M
	(+0.45)	(+0.21)	

Table 1: Facial Expression Recognition Results. error rates (Top-1, %) on FER-2013 and RAF.

Method	FER-2013	RAF	Params
baseline [21]	32.19	18.55	0.47M
DML [77]	31.91	18.48	1.4M
ONE [78]	31.90	17.85	1.18M
OKDDip [7]	<u>31.40</u>	**17.42**	1.53M
Ours	**31.33**	<u>17.80</u>	**0.47M**

Table 2: Comparison with online distillation methods. Facial Expression Recognition error rates (Top-1, %) on FER-2013 and RAF. Target Network: ResNet-32 [21]. **Bold**: best result. <u>Underline</u>: second best.

4.2. Ablation Study

Model Component Analysis. Table 3 shows the benefits of individual components of IN on CIFAR100 using ResNet-110 as target network. We have these observations: (1) **W/O Online Distillation** by setting trade-off $\lambda = 0$, IN can be considered as Stochastic Depth [27] but with a small difference: for each backpropagation, Stochastic Depth update the weights of the sub-network, while IN updates weights of the full network. It shows a 1.77 % performance drop from the full method. (2) **W/O Backward full network** yields a degraded performance with a large deviation. This indicates knowledge transfer to all students is important. This is because all student networks are shared weights, updating the full network leads to a stronger student in the next forward pass. An alternative explanation is that the Teacher can be considered as All students together. Updating weights for the full network will lead to a stronger Teacher in the next iteration. This suggests IN achieves the efficacy of knowledge transfer between the teacher and student in an online manner.

Configuration	Error (%)
Baseline [21]	25.33
Stochastic Depth [27]	22.61
W/O Online Distillation ($\lambda = 0$)	22.63
W/O Backward Full network	23.15
Full	**21.60**

Table 3: Model component analysis of ResNet-110 as target network on CIFAR100.

5. Discussions

Connections with Self-distillation. STO shares the same spirit with the self-distillation approaches, but work on a different perspective. Typically self-distillation methods focus on manipulating different forms of information, such as internal representations within the network [76], model weights in the training trajectory [16, 64], training data/label [69, 57, 70, 63], etc. Our approach aims to explore the direction of network architecture, specifically achieving storage-efficient online distillation via taking advantage of the redundancy in the neural networks. From a fundamental perspective, our goal is to unravel more potentials of the neural network. We believe that combining these lines of research will lead to even better performance.

Training Time Cost. The 2-forward pass of IN will generate a small overhead to the training time compared to standard training. However, in practice, this overhead can almost be ignored. This is because the extra forward pass goes through a portion of the network, which costs less time than a normal forward pass. Also, a significant amount of training computations come from the backpropagation. So adding one forward pass will contribute to a relatively small portion in terms of the total training time. In our experiments, we observe IN has an approximately 1.24x training time of standard training, and this overhead seems to be acceptable in real-world practice. Besides, distillation can leads to fast convergence, which allows fewer training epochs in practice.

6. Conclusions

We proposed Student-Teacher Oneness (STO), a simple but effective training scheme that improves facial expression recognition. STO naturally integrates the properties of implicit ensemble and knowledge distillation, which leads to a storage-efficient training strategy with both higher performance and memory efficiency.

References

[1] Lei Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *NeurIPS*, 2014. 2

[2] Qian Bao, Wu Liu, Jun Hong, Lingyu Duan, and Tao Mei. Pose-native network architecture search for multi-person human pose estimation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 592–600, 2020. 1

[3] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Smash: one-shot model architecture search through hypernetworks. *arXiv preprint arXiv:1708.05344*, 2017. 3

[4] Cristian Bucilua, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006. 2

[5] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. Dropout distillation. In *International Conference on Machine Learning*, pages 99–107. PMLR, 2016. 3

[6] Qi Cai, Yingwei Pan, Yu Wang, Jingen Liu, Ting Yao, and Tao Mei. Learning a unified sample weighting network for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14173–14182, 2020. 1

[7] Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. Online knowledge distillation with diverse peers. In *AAAI*, 2020. 1, 2, 4, 5, 6

[8] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7028–7036, 2021. 2

[9] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Advances in neural information processing systems*, 2017. 2

[10] Hanting Chen, Yunhe Wang, Chang Xu, Chao Xu, and Dacheng Tao. Learning student networks via feature embedding. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):25–35, 2020. 2

[11] Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Darkrank: Accelerating deep metric learning via cross sample similarities transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 2

[12] Inseop Chung, SeongUk Park, Jangho Kim, and Nojun Kwak. Feature-map-level online adversarial knowledge distillation. In *International Conference on Machine Learning*, pages 2006–2015. PMLR, 2020. 2

[13] Qianggang Ding, Sifan Wu, Hao Sun, Jiadong Guo, and Shu-Tao Xia. Adaptive regularization of labels. *arXiv preprint arXiv:1908.05474*, 2019. 2

[14] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. Seed: Self-supervised distillation for visual representation. In *ICLR*, 2021. 2

[15] Yang Fu, Sifei Liu, Umar Iqbal, Shalini De Mello, Humphrey Shi, and Jan Kautz. Learning to track instances without video annotations. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 8680–8689, 2021. 1

[16] Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *PMLR*, 2018. 1, 2, 7

[17] Ian J Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al. Challenges in representation learning: A report on three machine learning contests. In *International conference on neural information processing*, pages 117–124. Springer, 2013. 5

[18] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021. 2

[19] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11020–11029, 2020. 2

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4, 6, 7

[22] Lingxiao He, Yinggang Wang, Wu Liu, He Zhao, Zhenan Sun, and Jiashi Feng. Foreground-aware pyramid reconstruction for alignment-free occluded person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8450–8459, 2019. 1

[23] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1921–1930, 2019. 2

[24] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3779–3787, 2019. 2

[25] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Workshop*, 2015. 1, 2, 4, 5

[26] Hanzhe Hu, Shuai Bai, Aoxue Li, Jinshi Cui, and Liwei Wang. Dense relation distillation with context-aware aggregation for few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10185–10194, 2021. 2

[27] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016. 3, 4, 5, 6, 7

[28] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 3

[29] Jangho Kim, SeongUk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. *arXiv preprint arXiv:1802.04977*, 2018. 2

[30] Seung Wook Kim and Hyo-Eun Kim. Transferring knowledge to smaller network with class-distance loss. In *ICLRW*, 2017. 2

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[32] Seunghyun Lee and Byung Cheol Song. Graph-based knowledge distillation by multi-head attention network. In *BMVC*, 2019. 2

[33] Changlin Li, Jiefeng Peng, Liuchun Yuan, Guangrun Wang, Xiaodan Liang, Liang Lin, and Xiaojun Chang. Block-wisely supervised neural architecture search with knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1989–1998, 2020. 3

[34] Shan Li, Weihong Deng, and JunPing Du. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 2584–2593. IEEE, 2017. 5

[35] Yufan Liu, Jiajiong Cao, Bing Li, Chunfeng Yuan, Weiming Hu, Yangxi Li, and Yunqiang Duan. Knowledge distillation via instance relationship graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7096–7104, 2019. 2

[36] Yu Liu, Xuhui Jia, Mingxing Tan, Raviteja Vemulapalli, Yukun Zhu, Bradley Green, and Xiaogang Wang. Search to distill: Pearls are everywhere but not the eyes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7539–7548, 2020. 3

[37] Zhong Meng, Jinyu Li, Yong Zhao, and Yifan Gong. Conditional teacher-student learning. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6445–6449. IEEE, 2019. 2

[38] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? In *NeurIPS*, 2019. 2

[39] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, pages 3967–3976, 2019. 2

[40] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *ECCV*, pages 268–284, 2018. 2

[41] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas. Heterogeneous knowledge distillation using information flow modeling. In *CVPR*, pages 2339–2348, 2020. 2

[42] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas. Probabilistic knowledge transfer for lightweight deep representation learning. *IEEE Transactions on Neural Networks and Learning Systems*, 32(5):2030–2039, 2020. 2

[43] Peyman Passban, Yimeng Wu, Mehdi Rezagholizadeh, and Qun Liu. Alp-kd: Attention-based layer projection for knowledge distillation. *arXiv preprint arXiv:2012.14022*, 2020. 2

[44] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5007–5016, 2019. 2

[45] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[46] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015. 1, 2

[47] Weijian Ruan, Chao Liang, Yi Yu, Zheng Wang, Wu Liu, Jun Chen, and Jiayi Ma. Correlation discrepancy insight network for video re-identification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 16(4):1–21, 2020. 1

[48] Guocong Song and Wei Chai. Collaborative learning for deep neural networks. In *NeurIPS*, 2018. 1, 2

[49] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *PMLR*, 2014. 3

[50] Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*, 2020. 2

[51] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. 2013. 1

[52] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, pages 1195–1204, 2017. 2

[53] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1365–1374, 2019. 2

[54] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *ICML*, 2013. 3

[55] Lin Wang and Kuk-Jin Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[56] Xiaobo Wang, Tianyu Fu, Shengcai Liao, Shuo Wang, Zhen Lei, and Tao Mei. Exclusivity-consistency regularized knowledge distillation for face recognition. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 325–342. Springer, 2020. 2

[57] Xinshao Wang, Yang Hua, Elyor Kodirov, and Neil M Robertson. Proselflc: Progressive self label correction for training robust deep neural networks. In *CVPR*, 2021. 7

[58] Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 409–424, 2018. 3

[59] Guile Wu and Shaogang Gong. Peer collaborative learning for online knowledge distillation. In *AAAI*, 2021. 2

[60] Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S Davis, Kristen Grauman, and Rogerio Feris. Blockdrop: Dynamic inference paths in residual networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8817–8826, 2018. 3

[61] Yibo Hu Xiang Wu, Ran He and Zhenan Sun. Learning an evolutionary embedding via massive knowledge distillation. *International journal of Computer Vision*, 2020. 2

[62] Guodong Xu, Ziwei Liu, Xiaoxiao Li, and Chen Change Loy. Knowledge distillation meets self-supervision. In *ECCV*, pages 588–604. Springer, 2020. 2

[63] Ting-Bing Xu and Cheng-Lin Liu. Data-distortion guided self-distillation for deep neural networks. In *AAAI*, 2019. 7

[64] Chenglin Yang, Lingxi Xie, Chi Su, and Alan L Yuille. Snapshot distillation: Teacher-student optimization in one generation. In *CVPR*, 2019. 1, 2, 7

[65] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*, 2017. 2

[66] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014. 1

[67] Jiahui Yu and Thomas S Huang. Universally slimmable networks and improved training techniques. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1803–1811, 2019. 3

[68] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks. In *ICLR*, 2019. 3

[69] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *CVPR*, 2020. 7

[70] Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. Regularizing class-wise predictions via self-knowledge distillation. In *CVPR*, 2020. 1, 2, 7

[71] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 6

[72] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017. 1, 2

[73] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 1

[74] Chenrui Zhang and Yuxin Peng. Better and faster: knowledge transfer from multiple self-supervised learning tasks via graph distillation for video classification. *arXiv preprint arXiv:1804.10069*, 2018. 2

[75] Feng Zhang, Xiatian Zhu, and Mao Ye. Fast human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3517–3526, 2019. 2

[76] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Im-

prove the performance of convolutional neural networks via self distillation. In *CVPR*, 2019. 1, 2, 7

[77] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *CVPR*, 2018. 1, 2, 4, 6

[78] Xiatian Zhu, Shaogang Gong, et al. Knowledge distillation by on-the-fly native ensemble. In *NeurIPS*, 2018. 1, 2, 4, 5, 6