Efficient Online Subclass Knowledge Distillation for Image Classification

Maria Tzelepi, Nikolaos Passalis and Anastasios Tefas
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
Email: {mtzelepi, passalis, tefas}@csd.auth.gr

***Abstract*—Deploying state-of-the-art deep learning models on embedded systems dictates certain storage and computation limitations. During the recent few years *Knowledge Distillation* (KD) has been recognized as a prominent approach to address this issue. That is, KD has been effectively proposed for training fast and compact deep learning models by transferring knowledge from more complex and powerful models. However, knowledge distillation, in its conventional form, involves multiple stages of training, rendering it a computationally and memory demanding procedure. In this paper, a novel single-stage self knowledge distillation method is proposed, namely *Online Subclass Knowledge Distillation* (OSKD), that aims at revealing the similarities inside classes, so as to improve the performance of any deep neural model in an online manner. Hence, as opposed to existing online distillation methods, we are able to acquire further knowledge from the model itself, without building multiple identical models or using multiple models to teach each other, rendering the proposed OSKD approach more efficient. The experimental evaluation on two datasets validates that the proposed method improves the classification performance.**

I. Introduction

Deep learning models [1] have been utilized in order to resolve a plethora of visual analysis tasks, accomplishing superior performance and overthrowing prior solutions in recent years [2]–[5]. Generally, state-of-the-art deep learning models owe their exceptional performance to their depth and complexity, significantly inhibiting their applicability on devices with restricted computational resources, such as mobile and embedded systems. Thus, developing compact and effective models able to address storage and computational constraints of autonomous systems has become a challenging task.

Substantial amount of research works have been recently carried out towards this goal [6]. One solution constitutes in designing models that satisfy the memory and computation requirements without considerably sacrificing the accuracy [7]–[12]. Several other possibilities also exist, including the removal of redundant parameters of the model through parameter pruning, reducing in this way the complexity of the model [13], [14] and the reduction of the required bits for the parameter representation in order to compress the model [11], [15]. Finally, *Knowledge Transfer* (KT) [16]–[23] has arisen as a promising way to settle this issue proposing to transfer the knowledge from one, usually larger, model to a more compact model. *Knowledge Distillation* (KD) [24]–[26] widely propagated through [27] constitutes the most prominent offshoot of KT.

The conventional KD describes the process where the knowledge of a complex model, known as teacher, which accomplishes high performance, is transferred to a more compact and faster model, known as student. The student model is trained to regress the so-called *soft labels* generated by the teacher model by raising the temperature of the softmax activation function on the output layer of the network (known as *softening* the output distribution). These soft labels convey more information of the way the model learns to generalize, as compared to the hard labels, aiming at implicitly recovering similarities over the data. Apart from the aforementioned approach where the knowledge is transferred from a powerful model to a weaker model, there have also been proposed approaches where the knowledge is transferred from a simpler to a more powerful model [18] or the knowledge is transferred from teachers to students of identical capacity [24], [28]–[30]. The aforementioned process is known as *self-distillation*.

KD methods can be divided into two categories: *online* and *offline* KD. Offline KD stands for what we have already described as the multi-stage process of training first a high-capacity and more powerful teacher network and then distilling the knowledge to a weaker student network by training it to mimic the teacher network. Conventional KD is a research topic that has been flourishing in the recent years with a broad spectrum of applications [19], [31]–[35]. However, offline KD is an enduring, computationally and memory demanding procedure. To this aim, several online KD works have been proposed recently. Online KD describes the process where the teacher and student networks are trained simultaneously, without requiring a separate stage for pre-training the teacher network. Online KD includes works proposing to train multiple models mutually from each other [36], as well as works proposing to create ensembles of multiple identical branches of a target network in order to build a strong teacher and distill the knowledge from the teacher to the target network [37].

In the following we present the intuition of the proposed online distillation method considering a probabilistic view of KD. That is, deep neural models transform the probability distribution of the data, layer by layer, learning increasingly complex layer representations. Considering a multi-class classification problem, a conventional supervised loss forces the

data representations in the output layer of the model to become one-hot representations. However, trying to convert the complex data representations to one-hot representations usually leads to over-fitting and also requires deeper and more complicated models. Thus, in traditional KD methods it is manifested that it is advantageous for each sample to maintain the similarities with the other classes, instead of merely training with the hard labels.

In this paper, we propose a novel online self distillation method, namely *Online Subclass Knowledge Distillation* (OSKD), considering that inside each class there is also a set of sub-classes that share semantic similarities (e.g., blue cars, inflatable boats, etc.). Thus, we argue that it is useful to maintain the similarities of the sub-classes in order to further enhance the generalization ability of the model. Since the sub-classes inside each class are unknown and we are not able to follow a similar approach of softening their distribution as in the conventional KD, we propose to estimate them using the neighborhood of each sample. That is, we assume that the nearest neighbors of each sample inside a class belong to the same sub-class (i.e., share the same semantic similarities).

Thus, apart from the conventional classification objective, we introduce an additional distillation objective which encourages the data representations to come closer to the nearest representations of the same class and concurrently to move further away from the nearest representations of the other classes, ensuring in this way that the distillation objective will not encourage the representation entanglement. It is worth noting that subclass information has been successfully used to improve the accuracy of various learning problems [38]–[40] highlighting the importance of exploiting this information during the training process of powerful, yet prone to over-fitting, deep learning models.

Summing up, the model is trained synchronously both with the conventional supervised loss (hard labels) and the soft labels so as to maintain these sub-class similarities, without also the need of fine-tuning any other hyperparameter such as the temperature of the softmax activation function. Furthermore, as compared to the existing online distillation approaches, the proposed method is computationally more efficient, since it is capable of deriving additional knowledge from the data themselves, without requiring to create multiple copies or branches of the network or utilize multiple models. Finally, it should be highlighted that the proposed distillation method can be combined with any other method for developing effective and faster models, e.g., [8], [9].

The rest of this paper is structured as follows. Section II discusses related online distillation works, as well as self distillation works. The proposed method is presented in Section III. The experiments conducted to evaluate the proposed method are provided in Section IV, while the conclusions are drawn in Section V.

II. Related Work

In this section we first discuss prior works in the field of offline distillation with special emphasis on self distillation and subsequently we present recent works on online KD.

A. Offline Distillation

Offline KD is a research topic that has gained considerable research attraction during the recent year [19], [41]. Several works have also been emerged in the recent literature, proposing self distillation approaches. For example, in [42], KD is applied from a teacher model to a student model of identical architecture where the student accomplishes better performance being also much faster. The flow of solution procedure matrix is utilized in this approach for transferring the knowledge between the intermediate layers. A self distillation approach where a teacher model is initially trained, and then, after its convergence, an identical student model is trained with both the goals of the hard labels and matching the output of the teacher model, however without softening the logits (i.e., the inputs to the final softmax activation function) by raising the temperature, is proposed in [28]. Similarly, a target model is trained with a conventional supervised loss, the self-discovered knowledge is extracted, and in the second training stage, the model is trained with both the supervised and the distillation losses in [29].

B. Online Distillation

During the recent years, several online distillation have been also proposed. The so-called codistillation method [43] improves the accuracy by proposing to train c copies of a target model in parallel, by adding a distillation term to the loss function of the i-th model to regress the average prediction of the other models. A quite similar approach where an ensemble of students teach each other throughout the training process is proposed in [36]. That is, each student is trained with a conventional supervised learning loss, and a distillation loss that aligns each student's class posterior with the class probabilities of other students. In this way, each model acts as a teacher of the other models. In this approach, as opposed to the aforementioned codistillation method [43], different networks can be used for the mutual training.

Subsequently, an online distillation approach where a multi-branch version of the network is created by adding identical branches each of which constitutes an independent classification model with shared low level layers, and a strong teacher model is created using a gated logit ensemble of the multiple branches in [37]. Each branch is trained with the conventional classification loss and the distillation loss which matches the teacher's prediction distributions.

In this work, an online self distillation method is proposed. That is, as opposed to the existing self distillation approaches, the knowledge is distilled within the same model in an online manner. The proposed approach does not use the aforementioned multiple stages of the training pipeline, which renders it more efficient. Furthermore, as opposed to the existing online distillation methods, the proposed method allows synchronous model updating, without the need of building multiple identical models, or using multiple (possibly different) models to mutually teach each other, which comes

with additional computational cost.

III. Proposed Method

We consider a C-class classification problem, and the labeled data $\{\boldsymbol{y}_i, \boldsymbol{c}_i\}_{i=1}^N$, where $\boldsymbol{y}_i \in \Re^D$ an input vector and D its dimensionality, while $\boldsymbol{c}_i \in \mathcal{Z}^{\mathcal{C}}$ corresponds to its C-dimensional one-hot class label vector (hard label). For an input space $\mathcal{Y} \subseteq \Re^D$ and an output space $\mathcal{F} \subseteq \Re^C$, we consider as $\phi(\cdot\,;\mathcal{W}) : \mathcal{Y} \to \mathcal{F}$ a deep neural network with $n \in \mathbb{N}$ layers, and set of parameters $\mathcal{W} = \{\boldsymbol{W}_1, \ldots, \boldsymbol{W}_n\}$ where $\boldsymbol{W}_i$ refers to the weights of the i-th layer, which transforms its input vector to a C-dimensional vector containing the probabilities for each class. That is, $\phi(\boldsymbol{y}_i\,;\mathcal{W}) \in \mathcal{F}$ corresponds to the output vector of $\boldsymbol{y}_i \in \mathcal{Y}$ given by the network ϕ with parameters $\mathcal{W}$.

In the typical classification problem, we seek for the parameters $\mathcal{W}^*$ that minimize the cross entropy loss, J_{ce}, between the output vector $\boldsymbol{y}_i$ and one-hot class label vector $\boldsymbol{c}_i$:

$$\mathcal{W}^* = \arg\min_{\mathcal{W}} \sum_{i=1}^{N} J_{ce}(\boldsymbol{c}_i, \phi(\boldsymbol{y}_i\,;\mathcal{W})), \tag{1}$$

The cross entropy loss for a sample i is formulated as:

$$J_{ce}(\boldsymbol{c}_i, \boldsymbol{y}_i) = \sum_{m=1}^{C} c_i^m log(z_i^m), \tag{2}$$

where c_i^m is the m-th element of $\boldsymbol{c}_i$ one-hot label vector, and z_i^m refers to the m-th element of the output of the network:

$$z_i^m = \frac{\exp(\phi(\boldsymbol{y}_i\,;\mathcal{W})^m)}{\sum_{j=1}^{C} \exp(\phi(\boldsymbol{y}_i\,;\mathcal{W})^j)}. \tag{3}$$

In this work, we propose to distill additional knowledge online from the model itself throughout the network's training. To this end, considering that there are sub-classes inside each class that share semantic similarities, we propose to maintain these similarities, which are ignored during the network's training only with the hard labels.

Thus, for each representation $\phi(\boldsymbol{y}_i\,;\mathcal{W}) \in \mathcal{F}$ we also define the set $\mathcal{R}^i$ containing the nearest representations belonging to the same class, $\phi(\boldsymbol{y}_i\,;\mathcal{W})$, and a set $\mathcal{V}^i$ containing the nearest representations belonging to different classes to the representation. Then, the distillation objective forces the representation to come closer to the nearest neighbors belonging to the same class. Furthermore, apart from the aforementioned criterion that encodes the subclass knowledge, we add a disentanglement criterion, that is each representation is also forced to move further away from the nearest representations belonging to different classes, so as to ensure that the distillation objective will not encourage the representation entanglement. That is, the overall distillation objective is formulated as:

$$\min_{\mathcal{W}} \mathcal{J}_1 = \min_{\mathcal{W}} \sum_{\boldsymbol{y}_i, \boldsymbol{y}_j \in \mathcal{R}^i} \|\phi(\boldsymbol{y}_i\,;\mathcal{W}) - \phi(\boldsymbol{y}_j\,;\mathcal{W})\|_2^2, \tag{4}$$

and

$$\max_{\mathcal{W}} \mathcal{J}_2 = \max_{\mathcal{W}} \sum_{\boldsymbol{y}_i, \boldsymbol{y}_l \in \mathcal{V}^i} \|\phi(\boldsymbol{y}_i\,;\mathcal{W}) - \phi(\boldsymbol{y}_l\,;\mathcal{W})\|_2^2. \tag{5}$$

As it has also been proven in [44], equations eq. (4) and (5) can be reformulated as:

$$\min_{\mathcal{W}} \mathcal{J}_1 = \min_{\mathcal{W}} \sum_{\boldsymbol{y}_i \in \mathcal{R}^i} \|\phi(\boldsymbol{y}_i\,;\mathcal{W}) - \boldsymbol{\mu}_r^i)\|_2^2, \tag{6}$$

and

$$\max_{\mathcal{W}} \mathcal{J}_2 = \max_{\mathcal{W}} \sum_{\boldsymbol{y}_i \in \mathcal{V}^i} \|\phi(\boldsymbol{y}_i\,;\mathcal{W}) - \boldsymbol{\mu}_v^i)\|_2^2 \tag{7}$$

respectively, where $\boldsymbol{\mu}_r^i = \frac{1}{|\mathcal{R}^i|} \sum_{\boldsymbol{y}_j \in \mathcal{R}^i} \phi(\boldsymbol{y}_j\,;\mathcal{W})$, and $\boldsymbol{\mu}_v^i = \frac{1}{|\mathcal{V}^i|} \sum_{\boldsymbol{y}_l \in \mathcal{V}^i} \phi(\boldsymbol{y}_l\,;\mathcal{W})$. This formulation allows for a simpler implementation of the distillation objective. Thus, the overall distillation loss is formulating as: $J_{oskd} = \mathcal{J}_1 + (1 - \mathcal{J}_2)$.

Consequently, in the proposed distillation training procedure, we seek for the parameters $\mathcal{W}^*$ that minimize the overall loss of cross entropy, J_{ce} and distillation, J_{oskd}:

$$\mathcal{W}^* = \arg\min_{\mathcal{W}} \sum_{i=1}^{N} [J_{ce}(\boldsymbol{c}_i, \phi(\boldsymbol{y}_i\,;\mathcal{W})) + \lambda J_{oskd}(\boldsymbol{\mu}_r^i, \boldsymbol{\mu}_v^i, \phi(\boldsymbol{y}_i\,;\mathcal{W}))], \tag{8}$$

where λ balances the importance between predicting the hard labels and regressing the soft labels. Simple SGD is utilized to train the model:

$$\Delta\mathcal{W} = -\eta \frac{\vartheta J}{\vartheta \mathcal{W}}, \tag{9}$$

where J corresponds to the overall loss. In this way, the network, concurrently to the cross entropy loss, is trained to match the soft labels forcing the representations to maintain the similarities inside each class.

IV. Experiments

Three datasets were used to evaluate the performance of the proposed distillation method. The descriptions of the datasets and the utilized model architecture follow bellow. First, we utilized the MNIST dataset for building a binary classification problem (even digits against odd digits) which naturally offers known subclasses, for visualizing the effect of the proposed method. Subsequently, we performed four sets of experiments utilizing four different number of nearest neighbors (which in turn define the size of the subclasses) on Cifar-10 and SVHN datasets. Finally, an ablation study is conducted on Cifar-10 dataset in order to validate the effectiveness of subclass knowledge distillation. Test accuracy is used as evaluation metric. Each experiment is executed five times, and the mean value and the standard deviation are reported, considering the maximum value of test accuracy for each experiment. The curves of mean test accuracy are also provided.

A. Datasets and Experimental Setup

Three datasets were used for the experiments conduncted in this paper: the Cifar-10 dataset [45] the Street View House Numbers (SVHN) dataset [46] and the MNIST dataset [47]. The Cifar-10 dataset, [45] consists of 60,000 images of size 32×32 divided into 10 classes with 6,000 images per class. 50,000 images are used as the train set and 10,000 images as the test set. The Street View House Numbers (SVHN) dataset [46] obtained from house numbers in Google Street View images. It contains 73,257 train images and 26,032 test images, divided into 10 classes, 1 for each digit from 0 to 9. Input images are of size 32×32. The *MNIST* dataset, [47], of handwritten digits, has a train set of 60,000 images, and a test set of 10,000 images, divided into 10 classes, 1 for each digit from 0 to 9. Images are of size 28×28.

For all the conducted experiments we used a simple CNN model consisting of five layers; unless otherwise stated. The employed CNN model is composed of two convolutional layers with 6 filters of size 5×5 and 16 filters of size 5×5 respectively, followed by a Rectified Linear Unit (ReLU) [48] activation, and three fully connected layers ($128 \times 64 \times 10$). The convolutional layers are followed by a 2×2 max-pooling layer with a stride of 2. In the first two fully connected layers the activation function is the ReLU, while the last output layer is a 10-way softmax layer which produces a distribution over the 10 class labels of the utilized datasets. Finally, for comparison purposes against previous online KD approaches, we also utilize Wide-Res 16-2 [49] to perform experiments on Cifar-10 dataset.

All the experiments conducted using the Pytorch framework. The mini-batch gradient descent is used for the networks' training. In our experiments we set mini-batch size to 32. The learning rate is set to 10^{-3}, and the momentum is 0.9. The models are trained on an NVIDIA GeForce GTX 1080 with 8GB of GPU memory for 100 epochs. In order to select the weight factor λ in eq. (8) for controlling the balance between the contributing losses, we fix the number of nearest neighbors (i.e., we use 4 nearest neighbors) and we perform experiments for different values of the weight factor λ. The experimental results are presented in Fig. 1. As we can see, better results are achieved for $\lambda = 0.001$, and thus we use this value in the rest experiments. We should finally note, that better results could be accomplished through a more extended search for the optimal weight factor.

B. Experimental Results

First, a toy example is constructed in order to illustrate the effect of the proposed distillation method. More specifically, we use the MNIST dataset, and we build a binary classification problem for discriminating between even and odd digits. For each of the two classes we use three different digits, that is 0, 2, and 4 for the even class, and 1, 3, and 5 for the odd class. The train set consists of 36,018 samples, while the test set consists of 6,032 samples. In this way, we are able to acknowledge in retrospect that there are three distinct subclasses in each of the two classes. That is, even class consists in digits 0, 2, and 4, while odd class consists in digits 1, 3, and 5. Then, we train a simple CNN consisting of two convolutional and two fully connected layers with and without the proposed distillation objective. For the proposed distillation method, we consider 10 nearest neighbors for each sample inside each class, for a mini-batch of 60 samples. Then, we use the t-distributed Stochastic Neighbor Embedding (t-SNE) [50] Linear Discriminant Analysis (LDA) [51] and Locally Linear Embedding [52] algorithms to visualize the data representations in the penultimate layer. Experimental results for the test set are illustrated in Figs. 2-4. For visualization purposes, even we deal with a binary classification problem, we utilize different colors for each subclass. Thus, it is evident that the proposed distillation objective achieves to reveal the three subclasses inside each class, and force them to preserve their consistency preventing the samples collapse, allowing thus for conveying additional useful information, while also maintaining their discrimination ability.

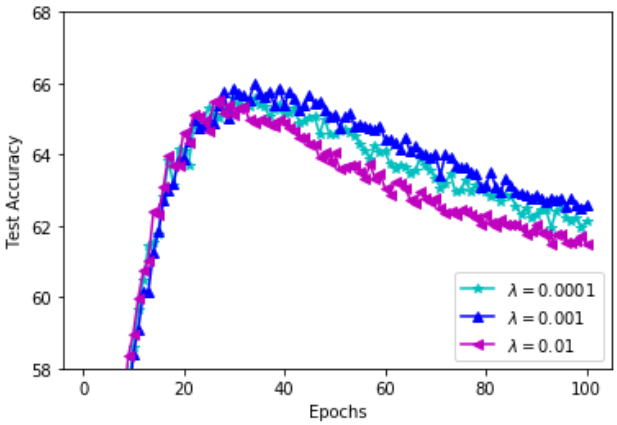
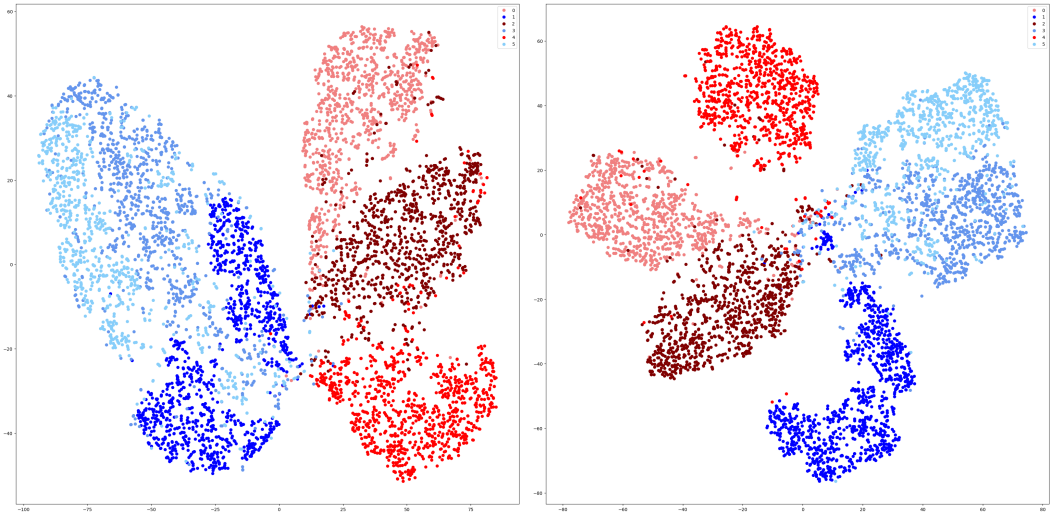


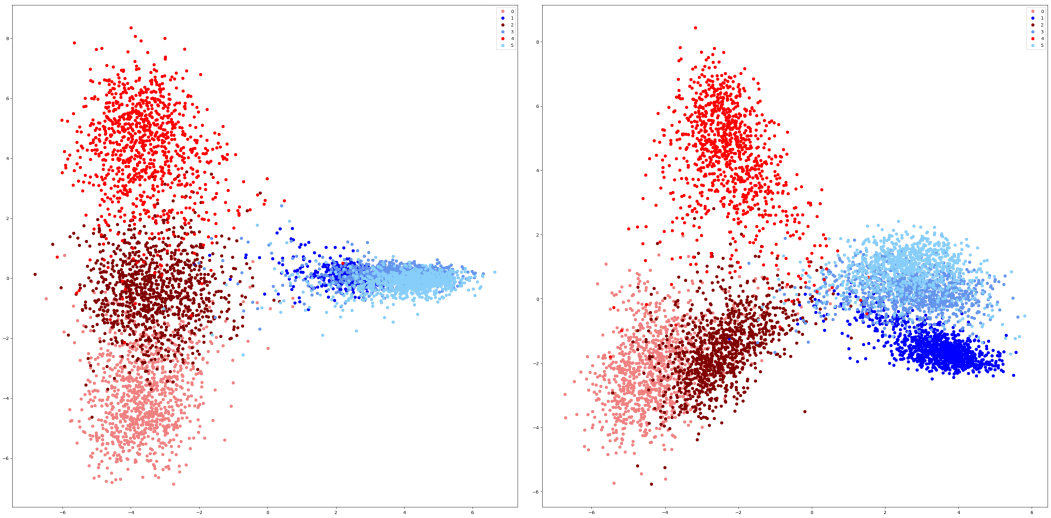
Fig. 1: Cifar-10: OSKD weight factor λ in eq. (8)

Subsequently, four sets of experiments performed, for four different numbers of nearest neighbors, in order to validate the proposed online distillation on both the utilized datasets. That is, we use 2, 4, 8 and 12 nearest neighbors for each sample (abbreviated as "OSKD - 2NN", "OSKD - 4NN", "OSKD - 8NN", and "OSKD - 12NN" respectively), and we compare the performance of the proposed method against the baseline performance, that is without distillation (abbreviated as "W/o Distillation"). The experimental results are presented in Table I. Best results are printed in bold. As we can observe from the reported results the proposed method considerably improves the baseline performance in all the considered cases. We can also observe that better results are reported for 12 nearest neighbors in both the considered cases. Correspondingly, in Figs. 5 and 6 the mean test accuracy of the proposed method for the three different number of nearest neighbors against the baseline method are reported for the Cifar-10 and SVHN datasets respectively, validating the enhanced performance of the proposed method. Furthermore, it is observed that the test accuracy decreases as the training progresses. This is attributed to over-fitting. However, it is evident that the aforementioned



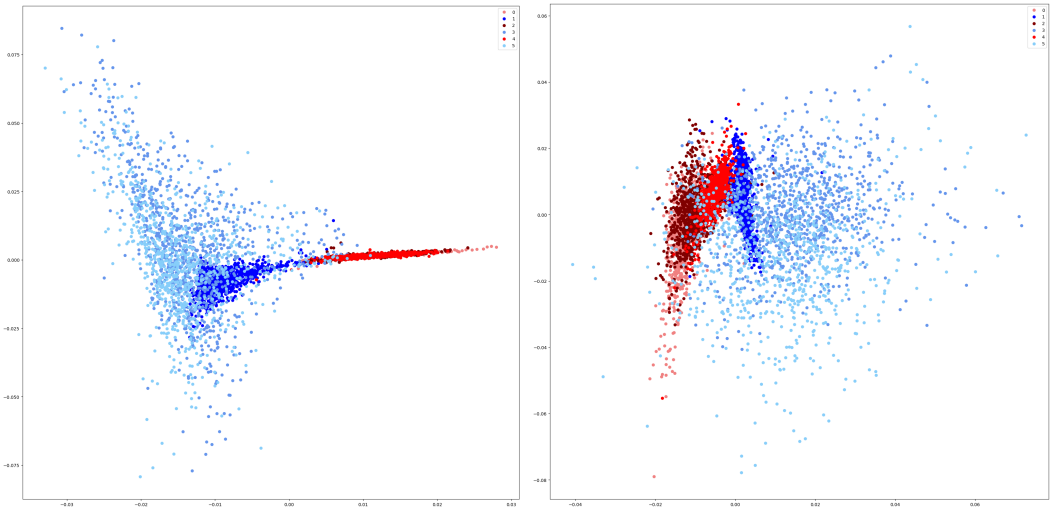
(a) W/o Distillation (b) OSKD

Fig. 2: T-SNE Visualization



(a) W/o Distillation (b) OSKD

Fig. 3: LDA Visualization



(a) W/o Distillation (b) OSKD

Fig. 4: LLE Visualization

reduction is significantly lower in the proposed OSKD method, as compared to models trained without distillation training. These results further highlight the regularization effect of the proposed online distillation method.

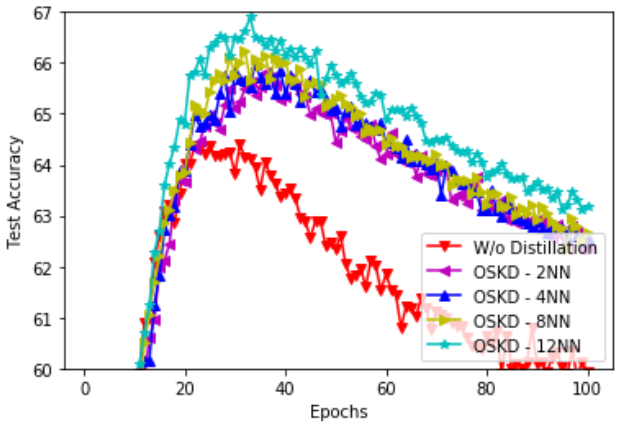


Fig. 5: Cifar-10: Test accuracy for different numbers of nearest neighbors inside each class

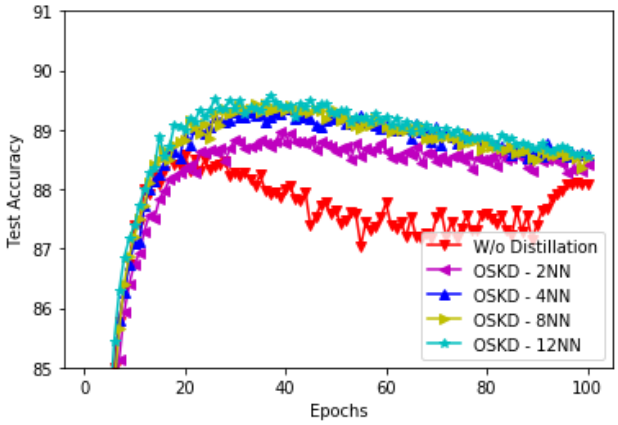


Fig. 6: SVHN: Test accuracy for different numbers of nearest neighbors inside each class

Method	Cifar-10	SVHN-10
W/o Distillation	64.83% ± 0.57%	88.82% ± 0.21%
OSKD - 2NN	66.16% ± 0.76%	89.00% ± 0.14%
OSKD - 4NN	66.39% ± 0.77%	89.52% ± 0.23%
OSKD - 8NN	66.59% ± 0.78%	89.61% ± 0.29%
OSKD - 12NN	**67.36% ± 0.82%**	**89.67% ± 0.28%**

TABLE I: Test accuracy

Subsequently, we compare the proposed method with ONE [37] and FFL [53] methods. We should note that for as much as possible fair comparisons, we use only two sub-networks in all the competitive approaches, similar to [53], since the proposed method does not utilize multiple branches of the network. Thus, we compare the OSKD method with ONE distillation method, considering the average performance of the two branches, and correspondingly with the FFL-S distillation method considering the average performance of the two sub-networks. We should also note that the number of parameters in both FFL-S and ONE methods is identical to the OSKD case, since the additional branches in both cases, as well as the fusion module in FFL-S, are discarded in the test phase. Furthermore, even we do not follow an ensembling methodology, we also compare the performance of the proposed OSKD method with the ensembling methods, that is ONE-E and FFL. It is noteworthy that the number of parameters in ONE-E is 1.24M, and 1.29M in FFL, while the number of parameters of OSKD is 0.70M, considering WRN 16-2. Evaluation results are presented in Table II. As it is shown, the proposed OSKD method achieves superior performance over competitive online distillation methods, as well as over ensebmling methods.

Method	Test Accuracy
WRN 16-2	93.55% ± 0.11%
ONE [37]	93.76%± 0.16%
FFL-S [53]	93.79% ± 0.12%
ONE-E [37]	93.84%± 0.20%
FFL [53]	93.86% ± 0.11%
OSKD	**93.96% ± 0.13%**

TABLE II: Comparisons against online distillation methods on Cifar-10 utilizing the WRN 16-2 architecture.

Furthermore, we evaluate the complexity of the proposed online distillation method using the sum of floating point operations (FLOPs) in one forward pass on a fixed input size. Model size, represented by the model's parameters, is also reported for each of the utilized models. To this aim, we utilize Wide ResNet 16-2 model on the Cifar-10 dataset, and we compare the complexity with the most famous offline KD method [27]. In this case, we use as teacher the stronger Wide ResNet 40-2 model (abbreviated as WRN 40-2). Evaluation results are provided in Table III. From the demonstrated results, it is validated the proposed online method is significantly more efficient as compared to the conventional offline methodology. Furthermore, we should note that even the student models under both online and offline procedures are similar, since the distillation concerns the training procedure, online self distillation comes with certain advantages against offline methodologies. That is, online methodologies can generally achieve superior performance over offline [37], and even more online self distillation methodology apart from the aforementioned significant gains in terms of computation and memory cost, achieves enhanced performance, and it is also guaranteed no compatibility issues between the student and teacher models will arise [35], since the additional knowledge derives from the model itself. We should finally note that competitive online distillation methods that utilize multiple branches or copies of a given network, require at least two times more FLOPs than the proposed one. That is, the proposed online distillation method is also more efficient as compared to competitive online methods, too.

Method	Teacher	Student	Complexity
KD [27]	WRN 40-2 (2.26M parameters)	WRN 16-2 (0.7M parameters)	0.43 GFLOPs
OSKD	-	WRN 16-2 (0.7M parameters)	0.10 GFLOPs

TABLE III: Complexity of the proposed OSAKD and KD [27] methods using the sum of floating point operations (FLOPs) in one forward pass on a fixed input size utilizing the Cifar-10 dataset. Model size, represented by the model's parameters, is also reported inside parentheses for each of the utilized models.

Finally, an ablation study is conducted in order to validate that the effectiveness of the proposed method derives from the subclass knowledge rather than the additional criterion that forces the data representations of each class to move further away from the nearest representations of the other classes so as to ensure that the distillation objective will not encourage the representation entanglement. To this aim, we perform experiments utilizing only the subclass objective, that is forcing each representation to come closer to the nearest neighbors belonging to the same class, without the aforementioned disentanglement criterion (denoted as Only Class), as well as utilizing only the disentanglement criterion, without the subclass objective (denoted as Only Non-Class). Experimental results on the Cifar-10 dataset are provided in Table IV. The expected number of samples of the same class is 2 to 4 for batches of 32 samples. Indeed, using these numbers of neighbors for estimating the subclasses leads to the best accuracy. On the other hand, the rest of the in-batch samples are expected to belong to a different class, and as a result, using a larger number of neighbors from different classes leads to better accuracy for the disentanglement criterion. We should highlight that the subclass criterion improves the performance in any case, confirming the subclass knowledge hypothesis. Furthermore, the best performance is accomplished by the combined objective.

NN	Only Class	Only Non-Class	Both
2	65.40% $\pm$ 1.18%	65.24% $\pm$ 0.84%	**66.16% $\pm$ 0.76%**
4	65.95% $\pm$ 0.63%	65.72% $\pm$ 0.73%	**66.39% $\pm$ 0.77%**
8	65.30% $\pm$ 0.53%	66.44% $\pm$ 0.45%	**66.59% $\pm$ 0.78%**
12	65.20% $\pm$ 0.67%	66.42% $\pm$ 0.67%	**67.36% $\pm$ 0.82%**

TABLE IV: Cifar-10 - Mini Batch Size: 32 (Baseline:64.83% $\pm$ 0.57%)

V. Conclusions

In this paper a novel single-stage self knowledge distillation method is proposed, namely *Online Subclass Knowledge Distillation*, that aims at recovering the similarities inside classes, improving the performance of any deep neural model in an online manner. As opposed to existing online distillation methods, the proposed method is capable of obtaining further knowledge from the model itself, without building multiple identical models or using multiple models to teach each other, rendering the OSKD method more effective. The experimental evaluation on two datasets indicates effectiveness of the proposed method in improving the classification performance.

Acknowledgement

This research is co-financed by Greece and the European Union (European Social Fund - ESF) through the Operational Programme "Human Resources Development, Education and Lifelong Learning 2014-2020" in the context of the project "Lighweight Deep Learning Models for Signal and Information Analysis" (MIS 5047925).

References

[1] L. Deng, "A tutorial survey of architectures, algorithms, and applications for deep learning," *APSIPA Transactions on Signal and Information Processing*, vol. 3, p. e2, 2014.

[2] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[3] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2016.

[4] A. Graves, A. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," *CoRR*, vol. abs/1303.5778, 2013.

[5] M. Tzelepi and A. Tefas, "Graph embedded convolutional neural networks in human crowd detection for drone flight safety," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2019.

[6] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," *arXiv preprint arXiv:1710.09282*, 2017.

[7] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[8] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6848–6856.

[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[10] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[11] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," in *ICLR*, 2016.

[12] G. Huang, S. Liu, L. Van der Maaten, and K. Q. Weinberger, "Condensenet: An efficient densenet using learned group convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2752–2761.

[13] S. Srinivas and R. V. Babu, "Data-free parameter pruning for deep neural networks," *arXiv preprint arXiv:1507.06149*, 2015.

[14] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," *arXiv preprint arXiv:1611.06440*, 2016.

[15] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4820–4828.

[16] T. Chen, I. Goodfellow, and J. Shlens, "Net2net: Accelerating learning via knowledge transfer," *arXiv preprint arXiv:1511.05641*, 2015.

[17] W. Chan, N. R. Ke, and I. Lane, "Transferring knowledge from a RNN to a DNN," *CoRR*, vol. abs/1504.01483, 2015. [Online]. Available: http://arxiv.org/abs/1504.01483

[18] Z. Tang, D. Wang, and Z. Zhang, "Recurrent neural network training with dark knowledge transfer," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 5900–5904.

[19] N. Passalis and A. Tefas, "Learning deep representations with probabilistic knowledge transfer," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 268–284.

[20] N. Passalis and A. Tefas, "Unsupervised knowledge transfer using similarity embeddings," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 946–950, 2019.

[21] J. Kim, S. Park, and N. Kwak, "Paraphrasing complex network: Network compression via factor transfer," in *Advances in Neural Information Processing Systems*, 2018, pp. 2760–2769.

[22] N. Passalis, M. Tzelepi, and A. Tefas, "Probabilistic knowledge transfer for lightweight deep representation learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[23] ——, "Multilayer probabilistic knowledge transfer for learning image representations," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2020, pp. 1–5.

[24] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *arXiv preprint arXiv:1412.6550*, 2014.

[25] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '06, 2006.

[26] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014, pp. 2654–2662.

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

[28] T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again neural networks," in *ICML*, 2018.

[29] X. Lan, X. Zhu, and S. Gong, "Self-referenced deep learning," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 284–300.

[30] S. Ahn, S. X. Hu, A. Damianou, N. D. Lawrence, and Z. Dai, "Variational information distillation for knowledge transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9163–9171.

[31] B. Pan, Y. Yang, H. Li, Z. Zhao, Y. Zhuang, D. Cai, and X. He, "Macnet: Transferring knowledge from machine comprehension to sequence-to-sequence models," in *Advances in Neural Information Processing Systems*, 2018, pp. 6092–6102.

[32] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang, "Structured knowledge distillation for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2604–2613.

[33] J. Mun, K. Lee, J. Shin, and B. Han, "Learning to specialize with knowledge distillation for visual question answering," in *Advances in Neural Information Processing Systems*, 2018, pp. 8081–8091.

[34] X. Wang, R. Zhang, Y. Sun, and J. Qi, "Kdgan: knowledge distillation with generative adversarial networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 775–786.

[35] S. Mirzadeh, M. Farajtabar, A. Li, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher," *CoRR*, vol. abs/1902.03393, 2019.

[36] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[37] x. lan, X. Zhu, and S. Gong, "Knowledge distillation by on-the-fly native ensemble," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018, pp. 7517–7527.

[38] S. Nikitidis, A. Tefas, N. Nikolaidis, and I. Pitas, "Subclass discriminant nonnegative matrix factorization for facial image analysis," *Pattern Recognition*, vol. 45, no. 12, pp. 4080–4091, 2012.

[39] S. Nikitidis, A. Tefas, and I. Pitas, "Projected gradients for subclass discriminant nonnegative subspace learning," *IEEE transactions on cybernetics*, vol. 44, no. 12, pp. 2806–2819, 2014.

[40] A. Maronidis, A. Tefas, and I. Pitas, "Subclass graph embedding and a marginal fisher analysis paradigm," *Pattern Recognition*, vol. 48, no. 12, pp. 4024–4035, 2015.

[41] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *arXiv preprint arXiv:2006.05525*, 2020.

[42] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[43] R. Anil, G. Pereyra, A. T. Passos, R. Ormandi, G. Dahl, and G. Hinton, "Large scale distributed neural network training through online distillation," 2018. [Online]. Available: https://openreview.net/pdf?id=rkr1UDeC-

[44] M. Kyperountas, A. Tefas, and I. Pitas, "Salient feature and reliable classifier selection for facial expression classification," *Pattern Recognition*, vol. 43, no. 3, pp. 972–986, 2010.

[45] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[46] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[47] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[48] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[49] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[50] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[51] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[52] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[53] J. Kim, M. Hyun, I. Chung, and N. Kwak, "Feature fusion for online mutual knowledge distillation," *arXiv preprint arXiv:1904.09058*, 2019.