mathematics

MDPI

Article

Effective Online Knowledge Distillation via Attention-Based Model Ensembling

Diana-Laura Borza †, Adrian Sergiu Darabant *,†, Tudor Alexandru Ileni † and Alexandru-Ion Marinescu †

Computer Science Department, Babes Bolyai University, 400084 Cluj-Napoca, Romania
* Correspondence: sergiu.darabant@ubbcluj.ro
† These authors contributed equally to this work.

Abstract: Large-scale deep learning models have achieved impressive results on a variety of tasks; however, their deployment on edge or mobile devices is still a challenge due to the limited available memory and computational capability. Knowledge distillation is an effective model compression technique, which can boost the performance of a lightweight student network by transferring the knowledge from a more complex model or an ensemble of models. Due to its reduced size, this lightweight model is more suitable for deployment on edge devices. In this paper, we introduce an online knowledge distillation framework, which relies on an original attention mechanism to effectively combine the predictions of a cohort of lightweight (student) networks into a powerful ensemble, and use this as a distillation signal. The proposed aggregation strategy uses the predictions of the individual students as well as ground truth data to determine a set of weights needed for ensembling these predictions. This mechanism is solely used during system training. When testing or at inference time, a single, lightweight student is extracted and used. The extensive experiments we performed on several image classification benchmarks, both by training models from scratch (on CIFAR-10, CIFAR-100, and Tiny ImageNet datasets) and using transfer learning (on Oxford Pets and Oxford Flowers datasets), showed that the proposed framework always leads to an improvement in the accuracy of knowledge-distilled students and demonstrates the effectiveness of the proposed solution. Moreover, in the case of ResNet architecture, we observed that the knowledge-distilled model achieves a higher accuracy than a deeper, individually trained ResNet model.

Keywords: online knowledge distillation; ensemble learning; attention aggregation; deep learning

MSC: 68T01; 68T07; 68T20; 68T30

check for updates

Citation: Borza, D.-L.; Darabant, A.S.; Ileni, T.A.; Marinescu, A.-I. Effective Online Knowledge Distillation via Attention-Based Model Ensembling. *Mathematics* **2022**, *10*, 4285. https://doi.org/10.3390/math10224285

Academic Editor: Jakub Nalepa

Received: 11 October 2022
Accepted: 13 November 2022
Published: 16 November 2022

1. Introduction

State-of-the-art machine learning models considerably improve the performance of various image understanding tasks, but they still fail to meet the non-functional requirements necessary for deployment on real-world test scenarios (inference time, latency, performance, throughput). Model compression techniques—such as quantization, pruning, low-rank approximation, knowledge distillation, and neural architecture search—aim to control the inference cost of neural networks [1].

There are numerous situations in which a trained neural network should be deployed on mobile devices (take, for example, the concrete case of an object identification/classification applet, which instead of sending the input to a dedicated server, runs the classification task on the user's device). In order to achieve this, knowledge distillation serves the purpose of a compression tool. By means of KD, the "knowledge" of a large trained model is transferred to a smaller student model. In this way, the lightweight model's accuracy on the test set is boosted, while preserving a low computational and memory footprint.

Knowledge distillation (KD) [2] is an effective technique to boost the accuracy of a lightweight network, by training it under the guidance of a more powerful network or an ensemble of networks. In the context of machine learning, knowledge typically refers to the learned weights of a network. Various distillation techniques have been proposed, in which the student mimics different knowledge sources of the teacher: the decision boundary (logits), intermediate feature maps, or intra-data relationships.

The classical formulation of KD [2,3], offline KD, involves a pre-trained teacher model with fixed weights when distilling knowledge to the student network. Despite its simplicity, this method involves two training steps (one for the teacher and one for the student). A large-capacity model might not always be available, and its training is resource consuming and cumbersome. In addition, the knowledge transfer is one-way (from the teacher to the student), as the teacher's weights are "frozen" during the second training stage. Finally, two-stage KD involves more parameters, which results in higher computational costs.

Online distillation frameworks [4–7] propose an alternative solution to the monolithic, large-capacity teacher, and simplify the training process by simultaneously training several peer students and learning from their combined predictions. One issue with online KD is related to the strategy of building the ensemble based on the students' individual predictions. Simply aggregating the students' logits affects the diversity of the student peers and, therefore, limits the effectiveness of online learning [8].

In this paper, we propose an effective online knowledge distillation framework to improve the generalization and learning capacity of a neural network architecture, while avoiding the increase in inference cost. Our main contribution is an original aggregation strategy inspired by attention mechanisms: The individual predictions of the peer student networks are combined via an attention module that assigns a weight to each network, and the ensemble is computed on the fly as the weighted average of the students' predictions. This ensemble is used throughout the training process as a distillation signal.

The remainder of this manuscript is organized as follows: In Section 2, we discuss other knowledge distillation frameworks, and in Section 3, we present the details of the proposed solution. Next, the experimental results and ablation studies are reported in Sections 4 and 5. Finally, Section 6 concludes this work.

2. Related Work

Knowledge distillation [3] was proposed as an effective and elegant compression technique to derive a lighter and faster network (student) from a more complex one (teacher), by penalizing the difference between their logits. Later, this mechanism was formalized by [2] to distill the "dark-knowledge" from the teacher to the student. The authors noticed that a powerful and confident teacher does not bring more knowledge than ground truth data, as its prediction tends to be a narrow probability distribution with a single peak for the ground truth class. To alleviate this issue, the teacher's logits are "softened" by a *temperature* scaling factor of the softmax activation. In such a manner, the lightweight network can infer what other classes were found similar by the teacher network. More formally, this can be expressed as

$$\zeta_\tau(z_i) = \frac{e^{z_i/\tau}}{\sum_j e^{z_j/\tau}} \tag{1}$$

where τ is the temperature of the softmax function (equation from [2]). When τ is greater than 1, the small probabilities of the softmax function are increased and the output is "softened".

In the classical setup, during training, a Kullback–Libeler divergence loss term is employed to ensure that the student network mimics the teacher's softened predictions. Other methods proposed using the root-mean-square error loss [9] or distilling from hard-labels [10].

2.1. Online KD

Online KD frameworks constitute an effective substitute for classical two-stage offline KD: instead of using a "good" pre-trained teacher network, a cohort of student peers are trained and share their knowledge.

In Deep Mutual Learning (DML) [5], several peer student networks mutually exchange information through a Kullback–Liebler loss term. In this framework, each student plays the role of the teacher for all the other peers. The main drawback of DML is that the predictions of the peer networks can conflict with each other (and even with the ground truth).

On-the-fly Native Ensemble (ONE) [4] is an online KD framework in which an ensemble is formed by adding several auxiliary branches over some shared low-level network layers. A gating component is used to assemble the knowledge of the branches into a more powerful prediction, which is, in turn, distilled back to all branches. This method is applicable only to branches with the same architecture, and the knowledge transfer occurs only at the branch layers.

In [6], the authors proposed a method for collaborative learning based on a hierarchical multiple branch network. The classifier heads provide different views on the data to improve generalization but also act as a regularization term. In addition, backpropagation rescaling was used to avoid gradient explosion and to provide supervision for the shared layers.

A similar approach to the proposed solution is the Knowledge Distillation method via Collaborative Learning (KDCL) [7]; it trains a pool of students together and aggregates their logits to generate soft targets for knowledge distillation. Four methods for assembling the students' predictions were proposed and compared. To ensure the diversity of the peers, each network applied a different set of augmentations to the training data. [11] combined online ensembling and network collaboration into a unified framework. The architecture consists in a multi-branch network, where each branch denotes a *peer*. To improve the quality of the KD, two teachers were computed online: the peer ensemble teacher, which distills knowledge from an online high-capacity teacher to each peer, and the peer mean teacher, which distills knowledge among peers. Random augmentations were performed multiple times on peer inputs.

In [12], Feature Fusion Learning (FFL) was proposed as an online distillation framework for intermediate feature maps. In this framework, several parallel sub-networks are trained together and a fusion module combines their feature maps into a more meaningful one. This is passed to a fused classifier, which performs the overall classification but also delivers its knowledge to each sub-network.

2.2. Attention-Based KD

Inspired by the human visual system, which can effectively focus on salient visual features of complex scenes, attention mechanisms have been integrated into various deep learning architectures, especially in the field of computer vision. The main idea is to redistribute the weights of a feature map according to a computed attention-mask.

SeNets [13] introduced an attention mechanism to perform a channel-wise feature re-calibration process by computing a weight for each channel in the feature maps. Inspired by this mechanism, in [14], the channel attention information is transferred from the teacher to the student. Channel attention weights are computed for the teacher's and student's intermediate feature maps, and the student is guided to learn the attention information of each channel.

In [15], the authors proposed an attention-based feature distillation mechanism, in which a meta-network employs a query-key attention component [16] to identify similarities between the student's and teacher's feature map. The resulting attention vector is used to transfer the teacher's knowledge selectively to student features. The query-key attention mechanism computes the similarities for all possible combinations between the teacher and student networks; so, the training process is computationally expensive.

Online Knowledge Distillation with Diverse peers (OKDDip) [17] applies a two-level KD using multiple auxiliary peers and one group leader. In the first level, group-based learning is achieved via an attention-based mechanism, while in the second level, the knowledge in the ensemble of peers is transferred to the group leader (the model used for inference). The main drawback of this method is that it involves a complex training strategy and more parameters are used during training.

3. Proposed Approach

The following notation will be used throughout this manuscript: N is the number of peer student networks trained within the proposed KD framework, C is the number of categories for the classification problem, $\hat{P}_i^k \in \mathbb{R}^C$ are the non-normalized logits of the i^{th} student network for the k^{th} image, G^k is the one-hot encoding of the ground truth data associated with the k^{th} sample, $\sigma(\cdot)$ is the sigmoid function, $\zeta(\cdot)$ is the softmax function, and $ReLU(\cdot)$ is the Rectified Linear Unit activation function. Table 1 organizes these notations in tabular form.

Table 1. Notations used throughout this manuscript.

Notation	Meaning
N	number of peer students
C	number of categories for the classification problem
$\sigma(\cdot)$	sigmoid function
$\zeta(\cdot)$	softmax function
$ReLU(\cdot)$	Rectified Linear Unit activation function
$\hat{P}_i^k$	non-normalized logits of the i^{th} student network for the k^{th} sample
G^k	one-hot encoding of the ground truth data of k^{th} sample

3.1. Research Methodology

Online KD methods are preferred to classical offline KD frameworks because they involve a simplified one-stage training process, in which all the models are treated as students that gain extra knowledge from each other's predictions or feature representations. This study investigates the problem of online KD and, more specifically, how the students' predictions can be combined to obtain an effective knowledge distillation signal. To this end, we propose a framework in which several student models are trained simultaneously, and an original attention-based aggregation mechanism is employed to combine their predictions into a powerful ensemble. The proposed method was tested on several image classification benchmarks using various network architectures. To demonstrate the effectiveness of the solution, we first train an individual model ("vanilla" model) on a classification benchmark using the classical cross-entropy loss function. Then, using the same training schedule and data processing techniques, we train several models with the same network architecture within the proposed KD framework. Throughout the training process, the predictions of the models are combined using the proposed attention mechanism and the "knowledge" of this ensemble guides the student models via an additional KD loss. For testing, a single knowledge-distilled student is selected—the one with the highest accuracy on the test set. To validate the proposed method, we compare the accuracy of the "vanilla" student with the accuracy of the knowledge-distilled student. The experimental results show that the accuracy of the knowledge-distilled model is always improved, regardless of the network architecture, classification benchmark, or training setup (from scratch or by using transfer learning).

3.2. Solution Outline

The outline of the proposed solution is depicted in Figure 1. A group of lightweight student peers is simultaneously trained into an online distillation framework and their predictions are aggregated into a more powerful ensemble based on an attention mechanism inspired by [18].

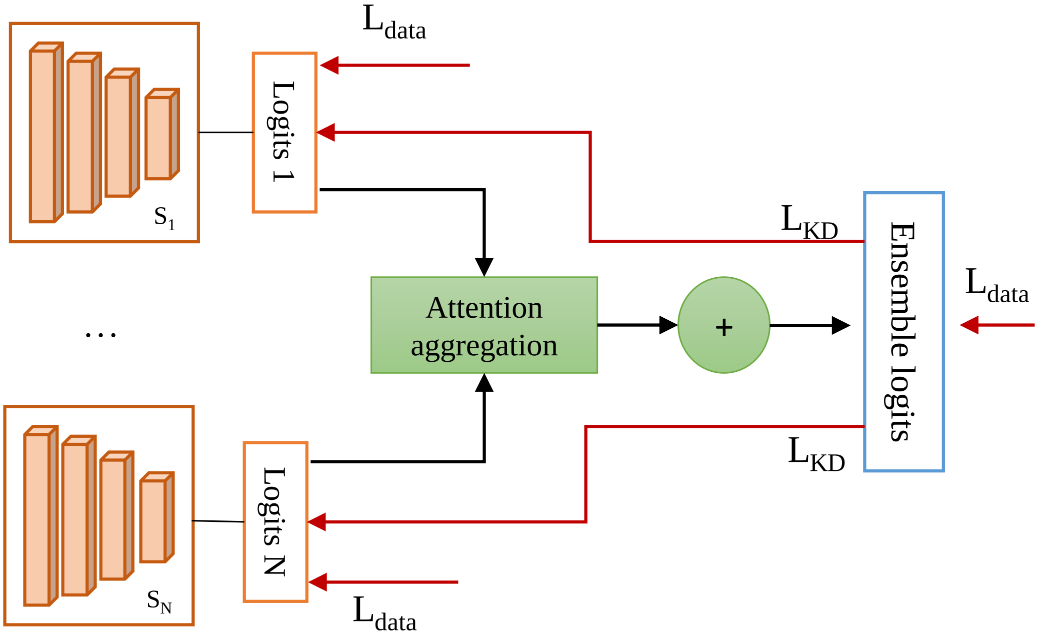

Figure 1. Solution outline. The proposed framework simultaneously trains a group of student peers and learns from the peers' predictions. The red arrows indicate the loss terms applied to each output: L_{data}—the standard cross-entropy loss; L_{KD}—the knowledge distillation loss.

The output of the attention aggregation module is a list of weights w (one for each peer network), used to ensemble the predictions ($\hat{E}$):

$$\hat{E} = \frac{\sum_{i=1}^{N} w_i \cdot P_i}{\sum_{i=1}^{N} w_i}. \tag{2}$$

The ensemble output ($\hat{E}$) will be used as a distillation signal throughout the training process.

All the models are trained in an end-to-end manner, with a multi-task loss function (L):

$$L = L_{data}(G, \hat{E}) + \sum_{i=1}^{N}(L_{data}(G, \hat{P}_i) + \lambda L_{KD}(\hat{E}, \hat{P}_i, \tau)) \tag{3}$$

where λ is the distillation strength and τ is the softmax temperature. The students and the ensemble's output logits are trained with the standard cross-entropy loss [19] (L_{data}):

$$L_{data}(G, \hat{P}) = -\sum_{x \in \chi} G(x) \cdot log(\zeta(\hat{P}(x))). \tag{4}$$

In addition, the ensemble's output is used as a distillation signal for all of the students, through a Kullback–Liebler [20] loss term (L_{KD}):

$$L_{KD}(G, \hat{P}, \tau) = \sum_{x \in \chi} \tau^2 D_{KL}(\zeta_\tau(\hat{E}), \zeta_\tau(\hat{P})), \tag{5}$$

where τ is the softmax temperature parameter; $\hat{E}$ and $\hat{P}$ are the ensemble's and student-softened predictions, respectively; and D_{KL} is the Kullback–Liebler divergence between two distributions p_1 and p_2 defined on the probability space χ:

$$D_{KL}(p_1, p_2) = -\sum_{x \in \chi} p_1(x) \cdot log \frac{p_2(x)}{p_1(x)}. \tag{6}$$

During the inference phase, a single knowledge-distilled student is used; in all the experiments, we report the accuracy of the most accurate student.

3.3. Ensembling Strategy

The question that remains is as follows: *"how could the individual predictions of the student networks be combined into a more powerful ensemble used as a distillation signal?"* To this end, we extract three features based on the predictions of the students, which are then fed to a channel attention mechanism (Figure 2) inspired by [18] to establish the ensembling weights (Equation (2)).

The proposed solution is directly influenced by the attention mechanism proposed in [18]; therefore, it will be described in detail. In [18], the authors proposed the Convolutional Block Attention Module (CBAM), an effective attention mechanism that computes attention maps across the channel and spatial dimensions of intermediate feature maps. The channel attention mechanism is used to ensure the model's focus on relevant features in the input volume; it starts with two pooling operations to aggregate the spatial information of each channel. These pooled features are then forwarded to a shared multi-layer perceptron with a single hidden layer. The output layer uses sigmoid activation to compute the weights for each channel in the input volume. Mathematically speaking, the channel attention mechanism [18] determines the weights w as follows:

$$w = \sigma(MLP(GAP(F)) + MLP(GMP(F))) \tag{7}$$

where MLP denotes the shared multi-layer perceptron, $GAP(\cdot)$ and $GMP(\cdot)$ represent the Global Average and Max pooling operations, F is the input feature map, and σ is the sigmoid activation function.

In the proposed method, the extracted features rely on global pooling operators and are then fed to a multi-layer perceptron, summed together, and passed through a sigmoid activation to determine the attention weights. The aggregation process is also guided by the ground truth data, as it is *solely* used during training. In the beginning, the individual predictions of the students are stacked into a single tensor $P_k \in \mathbb{R}^{N \times C}$. The entire process is detailed in Figure 2.

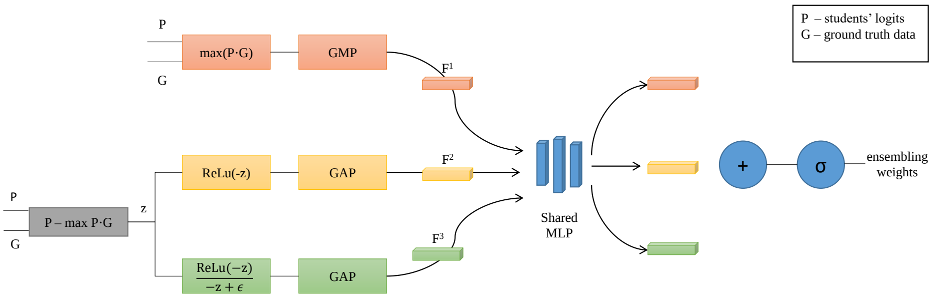

Figure 2. Ensembling strategy. The logits of the peer networks are assembled by an attention mechanism. P are the stacked logits of the sub-networks, and G represents the one-hot encoded ground truth data. Three features (F^1, F^2, and F^3) are computed and then passed through a multi-layer perceptron to compute the assembling weights of the ensemble.

The first extracted feature $F_k^1 \in \mathbb{R}^N$ is related to each student's prediction confidence for the ground truth class:

$$F_k^1 = GMP(P_k \cdot G_k) \tag{8}$$

where *GMP* is the Global Maximum Pooling operator. By multiplying the predictions with the one-hot encoding of ground truth, all except the ground truth class predictions will be set to zero. After applying the global maximum pooling operator, each network will be assigned either a strictly positive value (its confidence on the ground truth class) or a 0 (if its confidence for the ground truth class is less than 0).

The other features account for the students' predictions for the other classes. Let us define $z_k^i = (P_k^i - \max(P_k^i \cdot G_k^i))$ as difference between the i^{th} student's prediction and its prediction for the ground truth class. If the value on position j in z_k^i vector is positive, then the student's prediction for the j^{th} class is larger than for the ground truth class (i.e., the network is more confident on the j^{th} class than on the actual class).

The second feature F_k^2 accounts for the magnitude of the differences between the student's confidence in the actual class versus the other classes. By applying the ReLu activation function on the negative of z_k, all the classes on which the classifier was more confident than on the actual class will be assigned to zero, while for the other classes, the difference between the classifier's confidence in the actual class and the current class will be preserved. More formally, the second feature F_k^2 can be expressed as

$$F_k^2 = GAP(ReLU(-z_k)) \tag{9}$$

where *GMP* is the Global Maximum Pooling operator and $ReLU(\cdot)$ is the Rectified Linear Unit activation function.

Lastly, the third feature F_k^3 is related to the number of classes that have smaller confidence than the actual class. Similarly to F_k^2, we take the ReLU on the negative of z_k but we also divide the result by $-z_k$. In this way, all classes with smaller confidence than the ground truth class will be assigned to one.

$$F_k^3 = GAP\frac{ReLU(-z_k)}{-z_k + \epsilon} \tag{10}$$

where ϵ is a small constant to avoid division by zero.

Finally, as in [18], each of these descriptors (F^1, F^2, and F^3) are forwarded through a multi-layer perceptron with a single hidden layer. Then, to compute the combination weights, the outputs are merged using element-wise summation and passed through a sigmoid function. The ensemble's output is computed as the weighted average of the students' logits and these weights.

4. Experimental Results

In this section, we report the results of a series of experiments conducted to evaluate the proposed knowledge distillation framework on several image classification datasets. We evaluate our method on CIFAR-10, CIFAR-100, and TinyImageNet image classification benchmarks.

As the purpose of this study is to improve the accuracy of a lightweight model via knowledge distillation, without increasing its number of parameters, we employed the following evaluation strategy for a model. We first obtain the "vanilla" version of the model by training independently and evaluate it using the accuracy metric. Then, we train several models with the same architecture within the proposed knowledge distillation framework. For inference/deployment, we select the knowledge-distilled student with the highest accuracy on the test set and evaluate it. As we are interested in the improvement obtained after knowledge distillation, the metric that we are interested in is the gain in accuracy, which we compute as follows:

$$KD\_Gain = ACC_{KD} - ACC_{Vanilla} \tag{11}$$

where ACC_{KD} is the accuracy of the knowledge-distilled student and $ACC_{Vanilla}$ is the accuracy of the "vanilla", independently trained student.

The CIFAR-10, CIFAR-100, and TinyImageNet datasets are generic image classification benchmarks and have balanced testing sets; so, we report only the accuracy metric.

4.1. CIFAR-10 and CIFAR-100 Datasets

CIFAR-10 and CIFAR-100 datasets [21] comprise 60000 RGB-images with 32 × 32 image resolution, split into training (50,000 images) and validation (10,000 images) subsets. The images of CIFAR-10 are divided into 10 classes, while in CIFAR-100 each image is annotated with a "coarse" label (20 super-classes) and the actual class label (100 categories).

All the models were trained from scratch for 200 epochs, using a batch size of 32, with Adam optimizer and different variants of the ResNet architecture [22]. The initial learning rate was set to 10^{-3} and decayed by 0.1 at epochs 80, 140, and 170. In addition, a learning rate reducer was applied to reduce the learning rate by a factor of $\sqrt{0.1}$ if learning stagnates for 5 epochs. For experiments, $N = 3$ peer networks were trained, the softmax temperature τ was set to 3, and the knowledge distillation strength was set to $\lambda = 1$. (See the ablation studies in Section 5 for more information on varying this hyper-parameter.)

Table 2 reports the results on the CIFAR-10 and CIFAR-100 image classification benchmarks. *Vanilla* refers to the accuracy of the independently trained student and *KD* to the accuracy of the knowledge-distilled student. *MFLOPS* represents the number of mega FLOPS (floating point operations per second) of the model, and *Params.* represents the number of parameters of the model.

Table 2. Results on CIFAR-10 and CIFAR-100 datasets.

Dataset	Model	MFLOPS	Params.	Vanilla	KD	KD Gain
CIFAR-10	ResNet-20	82.298	274,442	92.1%	92.96%	0.86
	ResNet-32	139.325	470,218	92.77%	93.88%	1.11
CIFAR-100	ResNet-20	82.309	280,292	67.54%	69.73%	2.19
	ResNet-32	139.337	476,068	69.60%	72.76%	3.16
	ResNet-50	224.877	769,732	71.35%	73.93%	2.58

The knowledge-distilled network always surpasses an independently trained network. Moreover, the results show that the knowledge-distilled student has a higher accuracy than the immediately larger ResNet version (i.e., a knowledge-distilled ResNet-20 student surpasses an individually trained ResNet-32 network). As a detailed example, in the CIFAR-100 setup, a 0.19 accuracy improvement ($0.19 = 92.96 - 92.77$) is attained with a 41.12% decrease in parameters (from 470,218 to 274,442) and 40.93% decrease in the number of FLOPS (from 139.325 MFLOPS to 82.298 MFLOPS).

4.2. TinyImageNet Dataset

TinyImageNet [23] is a subset of the ImageNet [24] benchmark with 200 class categories; each category has 200 training images, 50 validation images, and 50 test images. In addition, the resolution of the images was reduced to 64 × 64.

For this setup, the networks were trained from scratch for 100 epochs with Adam optimizer. The initial learning rate was set to 10^{-3} and decayed by 0.1 at epochs 30, 60, and 90. To prevent overfitting, several geometrical augmentation techniques (horizontal flips, width and height shifts, rotations) as well as cutout [25] were applied to the training data. Similar to the CIFAR training setup, the hyper-parameters of the framework were set to $N = 3$, $\tau = 3$ and $\lambda = 1$.

Table 3 reports the results on the TinyImageNet classification benchmark. A 1.43% gain in accuracy is achieved when training the network in the proposed framework.

Table 3. Results on TinyImageNet dataset.

Model	MFLOPS	Params.	Vanilla	KD	KD Gain
ResNet-20	329.277	325,192	52.92%	54.35%	1.43%

4.3. Transfer Learning

Collecting a large-scale dataset required for training an accurate deep model is a challenging and cumbersome task, if not impossible for some tasks for which a limited amount of data are available. Nowadays, as a wide variety of pre-trained models are publicly available, transfer learning is the norm. In this section, we experiment with transfer learning on two classification benchmarks (Oxford Pets and Oxford Flowers datasets) with a limited amount of training data and higher resolution images than CIFAR and TinyImageNet datasets.

Oxford pets [26] contains 37 cat and dog breed categories, with approximately 200 images per class. The images feature large variations in size, lighting, and pose. Oxford flowers [27] is a 102 flower category dataset, and each class contains between 40 and 258 images. Both datasets are already split into training, validation, and test sets. For this experiment, we used the same splits as provided in the datasets. Oxford Flowers comprises 1020 training images, 1020 validation images, and 6149 test images, while Oxford Pets comprises 3680 training images and 3699 test images.

For the transfer learning setup, all peer student networks share the same "frozen" backbone (weights remain fixed during training) initialized with ImageNet weights [24], and only their final classification layers were trained. The weights of the architectures trained on the ImageNet dataset were retrieved from the *tensorflow* machine learning framework https://www.tensorflow.org/api_docs/python/tf/keras/applications, accessed on 25 August 2022. The final trainable classification layers were initialized using Xavier uniform initialization [28]. This process is depicted in Figure 3.

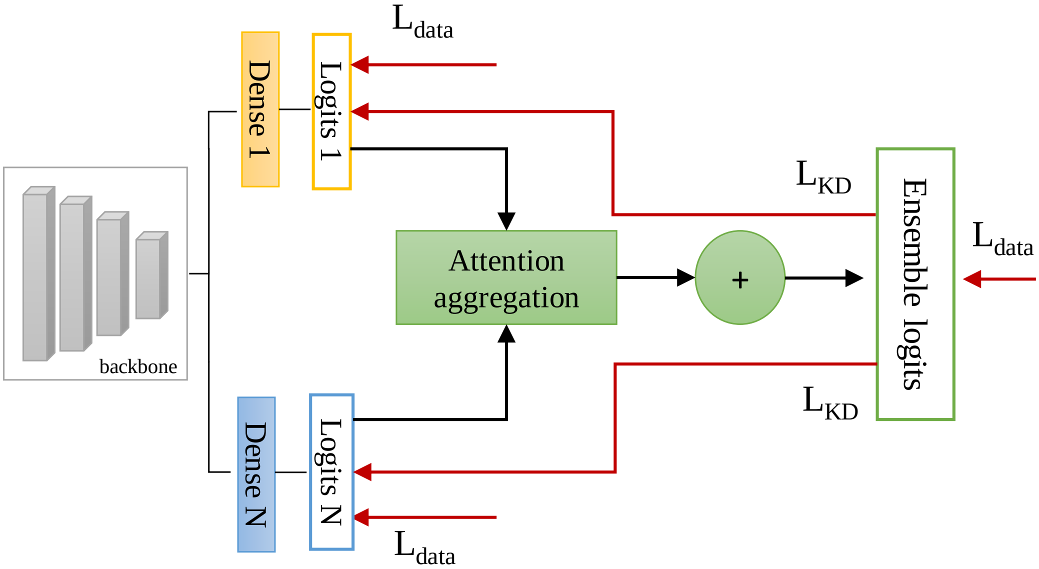

Figure 3. Transfer learning setup: The peer student networks share the same "frozen" backbone, initialized with ImageNet weights.

All the models were trained for 32 epochs, using RMSProp optimizer [29] with a learning rate of 10^{-3}. The data pre-processing step involved padding the images to square shape by duplicating the edge features and then resizing them to 224 × 224. Table 4 reports the results obtained when using transfer learning on Oxford Pets and Oxford Flowers datasets. We experimented with several neural network architectures: DenseNet [30], ResNet [22], NASNet [31], and MobileNet [32]. In this table, the column *Vanilla* indicates the accuracy of an independently trained, "vanilla" student, the column *KD* reports the accuracy of a knowledge-distilled student, and *KD Gain* is the accuracy improvement obtained when training a network in the proposed KD framework.

Table 4. Transfer learning results on Oxford Pets [26] and Oxford Flowers datasets [27].

Model	Oxford Pets			Oxford Flowers		
	Vanilla	KD	KD Gain	Vanilla	KD	KD Gain
DenseNet	90.51	90.89	0.38	85.66	86.11	0.45
ResNet-50	87.66	88.87	1.21	85.31	85.51	0.20
NASNet	87.44	87.63	0.19	69.36	69.85	0.49
MobileNet	88.43	88.98	0.55	82.04	82.68	0.64

The knowledge-distilled student always surpasses the accuracy of the vanilla-trained network; however, in this case, the improvement is lower than by training the networks from scratch. This is expected, as only the final layer benefits from knowledge distillation because the rest of the weights in the peer student networks are frozen.

4.4. Comparison with State of the Art

In this section, we provide a comparison of the proposed KD framework with similar methods for the literature. It is noteworthy that a direct numerical comparison is not always relevant, as other methods use different training schedules, optimizers, and model architectures, all of which have an impact on the networks' performance. However, as we are dealing with the process of KD (in which we want to improve an individually trained model by making it benefit from the knowledge of a more powerful model), we are actually interested in the gain of the knowledge distillation process (i.e., difference between the accuracy of the knowledge-distilled student and the individually trained model). Consequently, when interpreting the results, we rely on the accuracy improvement of a knowledge-distilled student over an independently trained student.

Table 5 compares the results obtained on CIFAR-10 classification benchmark.

Table 5. Comparison with state-of-the-art works on CIFAR-10 database using ResNet-32 [22] as student network.(best gain represented in bold).

Method	Vanilla	KD	KD Gain
ONE [4]	93.07%	94.01%	0.94%
CLCNN [6]	93.17%	94.14%	0.97%
OKDDip net. [17]	93.66%	94.38%	0.72%
OKDDip br. [17]	93.66%	94.42%	0.76%
PCL [11]	93.26%	94.33%	1.07%
Proposed	92.77%	93.88%	**1.11%**

The improvement of the knowledge-distilled students over the independently trained student is marginal (around 1% for all the methods); however, these accuracy levels are close to the reported human accuracy on CIFAR-10 [33].

When compared with other works, the proposed method achieves a higher accuracy gain after knowledge distillation. As opposed to the proposed method, in ONE [4] and CLCNN [6], the low-level layers of the student networks are shared, which could limit the discriminative power and diversity of peer networks.

In Table 6, we compare the proposed solution with other knowledge distillation frameworks on the CIFAR-100 dataset.

Table 6. Comparison with state-of-the-art works for ResNet-32 architecture trained on CIFAR-100 dataset.

Method	Vanilla	KD	KD Gain
DML [5]	68.99%	71.19%	2.20%
KDCL [7]	71.28%	73.76%	2.48%
OKDDip net. [17]	71.24%	74.60%	3.36%
OKDDip br. [17]	71.24%	74.37%	3.13%
SAD [15]	75.32%	77.47%	2.15%
PCL [11]	71.28%	74.14%	2.86%
Proposed	69.6%	72.76%	3.16%

For a fair comparison, the results of KDCL [7] were retrieved from [11], in which the KDCL framework was trained on three parallel peer networks.

The proposed method surpasses DML [5] by almost 0.96% accuracy gain. In DML, a cohort of student networks is trained together with mutual distillation and the parameters of the network are updated in a multi-stage setting. Furthermore, we surpass SAD [15], an offline attention-based KD framework by over 1% accuracy gain. In SAD, a pre-trained teacher network is used together with an attention module that learns the similarities between the teacher's and the student's feature maps, and then applies them to control the distillation intensities of all possible pairs.

Compared with other online KD frameworks [7,11,17], the proposed method attains comparable if not better results. OKDDip [17] can be implemented either in a branch-based setting (the low-level layers of the students are shared)—denoted *br.* in Table 6, or in a network-based setting (each student is an individual network)—denoted *net.* in the table. The OKDDip in-network approach slightly surpassed the proposed method by 0.2%; however, it uses more parameters for training and a more complex training strategy.

5. Ablation Studies

In this section, we perform a series of ablation studies to investigate further properties of the proposed method. All the experiments reported in this section use ResNet-20 [22] as peer student networks trained on the CIFAR-100 dataset. For all studies—except Section 5.1, where N varies from 2 to 5—the number of student networks is set to $N = 3$.

5.1. Number of Student Networks

In the proposed KD framework, the "teacher" is computed on the fly as the weighted average of the N peer student networks. Table 7 reports the impact of this hyper-parameter N over the accuracy of the knowledge-distilled student and the ensemble. The first row from the table accounts for the accuracy of a vanilla-trained student.

Table 7. The effect of the number of students N on the distillation performance.(best results represented in bold).

N	KD	Ensemble	KD Gain
1	67.54%	N/A	N/A
2	69.30%	72.28%	1.76
3	69.73%	73.32%	2.19
4	69.01%	73.61%	1.47
5	**69.9**%	73.69%	**2.36**

It should be noted that the ensemble is used only as a training component (the "teacher" from which the knowledge is distilled) and cannot be used for inference on real-world data because the ground truth data are required when computing the attention features (Section 3.3). Nevertheless, we report its accuracy on the test data to analyze the relationship between the teacher's accuracy and the accuracy of the knowledge-distilled students.

As the number of peer student networks N increases, the accuracy of the ensemble increases as well. The accuracy of the knowledge-distilled students follows the trend, but there is an exception for the case of $N = 4$ students, where we obtained a slightly lower and out-of-order gain in accuracy than the other configurations. Still, in all cases, when training a network in the proposed KD framework we obtain a higher accuracy (with an accuracy boost ranging from 1.47 when $N = 4$ to 2.36 when $N = 5$) than by independently training it. Continuously increasing the number of students would not continually improve the performance of the best student as their knowledge absorption is not boundless.

5.2. Ensembling Features

The main contribution of this paper is the attention-based ensembling strategy that is used to compute the distillation signal across the training process. To compute the ensembling weights, we extracted three features based on the students' logits and the ground truth data (as described in Section 3.3). In Table 8, we analyze the impact of these features on the accuracy of the knowledge-distilled student and the ensemble.

Table 8. Ensembling features.(best results represented in bold).

Features	KD	Ensemble	KD Gain
Vanilla	67.54%	N/A	N/A
F^1	67.89%	71.66%	0.35
F^2	68.76%	72.74%	1.22
F^3	67.29%	71.57%	-0.25
F^1F^2	69.59%	75.5%	2.05
F^1F^3	68.34%	72.39%	0.80
F^2F^3	68.98%	72.64%	1.44
$F^1F^2F^3$	69.73%	73.32%	**2.19**

The first column indicates which features are used by the attention mechanism when computing the ensembling weights. Feature F^2 has the most discriminative power. The experiments show that the best results are obtained when using all the proposed features F^1, F^2, and F^3. The ensemble with the highest accuracy is the one obtained using features (F_1, F_2) at 75.5%. However, this is not the setup yielding the best knowledge-distilled student. Feature F_3 used individually brings a negative gain. However, when combined with either F_1 or F_2, it increases their respective induced gains behaving like a catalyst. When used in combination (F_1, F_2, F_3), F_3 keeps its catalytic effect, helping this combination of features achieve the largest improvement.

5.3. Distillation Strength

Finally, we analyze the impact of the knowledge distillation loss weight λ (Equation (3)) over the accuracy of the knowledge-distilled students (Table 9). The first row from the table represents the accuracy of an independently trained student.

Table 9. Impact of the KD loss weight.(best results represented in bold).

λ	KD	KD Gain
Vanilla	67.54%	N/A
0	68.01%	0.47
0.1	68.69%	1.15
1	69.73%	**2.19**
2	69.41%	1.87
3	68.94%	1.40

Even when the knowledge distillation weight is disabled ($\lambda = 0$), we observe a slight improvement over the independently trained "vanilla" student; so, just by training a model in the proposed framework without any KD loss (Equation (5)), a small boost in accuracy is observed. This is due to the regularizing effect of training the cohort of students together. As we increase the value of λ towards 1, the accuracy of the knowledge-distilled students increases; however, for larger values, the accuracy starts to decrease. Within the proposed framework, the student networks are influenced by two loss functions (Equation (3)): the classification loss L_{data} and the knowledge distillation loss (Kullback–Liebler divergence) L_{KD}. As λ increases, the weight of the latter also increases, while the weight of the classification loss remains the same. These results indicate that the classification loss still plays an important role in the proposed framework and that the students benefit from the ground truth data. Moreover, the classification loss is important because the teacher ensemble relies on the predictions of the individual students.

As future work, we also plan to experiment by varying the λ parameter during the training process.

6. Conclusions and Future Work

This work tackled the problem of knowledge distillation, a teacher–student training setup in which a lightweight student network is guided by the knowledge of a teacher network formed from lightweight students to improve its accuracy while maintaining a low computational cost. As opposed to offline KD, where a pre-trained powerful teacher is required, the proposed method simultaneously trains a group of student models and learns from peers' predictions, which are aggregated into an ensemble via an attention mechanism. The attention mechanism is employed solely throughout the training process and, for inference, a single lightweight network is selected and used. The main advantage of this work is that by training a model in the proposed online KD framework, its accuracy is boosted, without adding any additional parameters. In addition, the learning process is end-to-end, and the teacher's (ensemble) logits are computed on the fly using an original attention-based mechanism.

The proposed method was evaluated on several image classification benchmarks, both by training the networks from scratch and by transfer learning. We proved that the proposed method can be generically used to improve the accuracy of image classification problems. The results show that by training a network in this framework, the performance of the knowledge-distilled student is consistently improved when compared with the vanilla-trained networks. Moreover, for the ResNet architecture, a knowledge-distilled student exceeds the performance of the immediately larger ResNet model trained on the vanilla configuration. Our method was compared with several state-of-the-art works on multiple image classification benchmarks, and the experimental results shows that it attains comparable if not better results. Although the "gain" is not significant in absolute terms, its magnitude is on par with the results from the state of the art. The main contribution is to achieve better accuracy using the same network architectures by slightly changing the training process. Thus, the same known network architectures will obtain better results using KD techniques on the same datasets.

As for all online knowledge distillation methods, one limitation of the present study is the exact mechanism that leads to the accuracy improvement of knowledge distillation over other approaches has not been fully understood [8]. Moreover, the training process has a higher computational resource footprint, as it involves optimizing several peer students at once. However, as opposed to offline KD methods, the training process occurs in a single step.

As future work, we plan to incorporate attention mechanisms to also distill knowledge from intermediate feature maps and to extend the framework for other vision tasks.

Author Contributions: Conceptualization, D.-L.B. and A.S.D.; methodology, D.-L.B.; software, D.-L.B.; validation, D.-L.B., A.S.D., T.A.I. and A.-I.M.; investigation, D.-L.B., A.S.D., T.A.I. and A.-I.M.; resources, D.-L.B. and A.-I.M.; writing—review and editing, T.A.I.; visualization, T.A.I. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a grant of the Romanian Ministry of Education and Research—UEFISCDI, project number PN-III-P2-2.1-PED-2021-3120.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FLOPS	Floating Point Operations Per Second
GAP	Global Average Pooling
GMP	Global Max Pooling
KD	Knowledge distillation
KL	Kullback–Leibler divergence
ReLU	Rectified Linear Unit

References

1. Cai, H.; Lin, J.; Lin, Y.; Liu, Z.; Tang, H.; Wang, H.; Zhu, L.; Han, S. Enable deep learning on mobile devices: Methods, systems, and applications. *ACM Trans. Des. Autom. Electron. Syst. (TODAES)* **2022**, *27*, 1–50. [CrossRef]
2. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
3. Buciluă, C.; Caruana, R.; Niculescu-Mizil, A. Model compression. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PE, USA, 20–23 August 2006; pp. 535–541.
4. Lan, X.; Zhu, X.; Gong, S. Knowledge distillation by on-the-fly native ensemble. *arXiv* **2018**, arXiv:1806.04606.
5. Zhang, Y.; Xiang, T.; Hospedales, T.M.; Lu, H. Deep mutual learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4320–4328.
6. Song, G.; Chai, W. Collaborative learning for deep neural networks. *arXiv* **2021**, arXiv:1805.11761.
7. Guo, Q.; Wang, X.; Wu, Y.; Yu, Z.; Liang, D.; Hu, X.; Luo, P. Online knowledge distillation via collaborative learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11020–11029.
8. Wang, L.; Yoon, K.J. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *arXiv* **2018**, arXiv:2004.05937.
9. Kim, T.; Oh, J.; Kim, N.; Cho, S.; Yun, S.Y. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. *arXiv* **2021**, arXiv:2105.08919.
10. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training data-efficient image transformers & distillation through attention. In Proceedings of the International Conference on Machine Learning. PMLR, Virtual, 18–24 July 2021; pp. 10347–10357.
11. Wu, G.; Gong, S. Peer collaborative learning for online knowledge distillation. *Proc. Aaai Conf. Artif. Intell.* **2021**, *35*, 10302–10310. [CrossRef]
12. Kim, J.; Hyun, M.; Chung, I.; Kwak, N. Feature fusion for online mutual knowledge distillation. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 4619–4625.
13. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
14. Zhou, Z.; Zhuge, C.; Guan, X.; Liu, W. Channel distillation: Channel-wise attention for knowledge distillation. *arXiv* **2020**, arXiv:2006.01683.
15. Ji, M.; Heo, B.; Park, S. Show, attend and distill: Knowledge distillation via attention-based feature matching. *Proc. Aaai Conf. Artif. Intell.* **2021**, *35*, 7945–7952. [CrossRef]
16. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
17. Chen, D.; Mei, J.P.; Wang, C.; Feng, Y.; Chen, C. Online knowledge distillation with diverse peers. *Proc. Aaai Conf. Artif. Intell.* **2020**, *34*, 3430–3437. [CrossRef]
18. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
19. Good, I. Some terminology and notation in information theory. *Proc.-IEE-Part Monogr.* **1956**, *103*, 200–204. [CrossRef]
20. Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [CrossRef]

21. Krizhevsky, A.; Hinton, G. Learning multiple Layers of Features from Tiny Images. Technical Report, University of Toronto, Toronto, ON, Canada. 2009. Available online: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf (accessed on 1 October 2022).
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
23. Le, Y.; Yang, X. Tiny imagenet visual recognition challenge. *CS 231N* **2015**, *7*, 3.
24. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
25. DeVries, T.; Taylor, G.W. Improved regularization of convolutional neural networks with cutout. *arXiv* **2017**, arXiv:1708.04552.
26. Parkhi, O.M.; Vedaldi, A.; Zisserman, A.; Jawahar, C. Cats and dogs. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3498–3505.
27. Nilsback, M.E.; Zisserman, A. Automated flower classification over a large number of classes. In Proceedings of the 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, Bhubaneswar, India, 16–19 December 2008; pp. 722–729.
28. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; JMLR Workshop and Conference Proceedings; pp. 249–256.
29. Hinton, G.; Srivastava, N.; Swersky, K. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited On* **2012**, *14*, 2.
30. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
31. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8697–8710.
32. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
33. Ho-Phuoc, T. CIFAR10 to compare visual recognition performance between deep neural networks and humans. *arXiv* **2018**, arXiv:1811.07270.