applied sciences

MDPI

Article

Improving Deep Mutual Learning via Knowledge Distillation

Achmad Lukman * and Chuan-Kai Yang

Department of Information Management, School of Management, National Taiwan University of Science and Technology, Taipei City 106335, Taiwan

* Correspondence: mecaman@gmail.com or d10509802@mail.ntust.edu.tw

Abstract: Knowledge transfer has become very popular in recent years, and it is either based on a one-way transfer method used with knowledge distillation or based on a two-way knowledge transfer implemented by deep mutual learning, while both of them adopt a teacher–student paradigm. A one-way based method is more simple and compact because it only involves an untrained low-capacity student and a high-capacity teacher network in the knowledge transfer process. In contrast, a two-way based method requires more training costs because it involves two or more low-cost network capacities from scratch simultaneously to obtain better accuracy results for each network. In this paper, we propose two new approaches, namely full deep distillation mutual learning (FDDML) and half deep distillation mutual learning (HDDML), and improve convolutional neural network performance. These approaches work with three losses by using variations of existing network architectures, and the experiments have been conducted on three public benchmark datasets. We test our method on some existing KT task methods, showing its performance over related methods.

Keywords: image classification; knowledge distillation; mutual learning; convolutional neural network

check for updates

Citation: Lukman, A.; Yang, C.-K. Improving Deep Mutual Learning via Knowledge Distillation. *Appl. Sci.* **2022**, *12*, 7916. https://doi.org/10.3390/app12157916

Academic Editor: José Salvador Sánchez Garreta

Received: 24 July 2022
Accepted: 4 August 2022
Published: 7 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Research on the modification of deep convolutional neural networks has been very popular in recent years because it can improve the performance of many computer vision tasks to solve many problems, such as object detection and semantic segmentation [1], image classification [2], or image recognition [3], such as neural network model for diagnostic chest X-ray images [4]. This research uses a hybrid CNN whose method is a combination of inception V4 and multiclass SVM. Both of these methods have their respective tasks, namely inception V4 functions to perform feature extraction of chest X-ray images while multiclass SVM is a classifier. Another study on the identification and classification of cassava leaf disease [5] used an enhanced convolutional neural network that applied its novelty in the form of the global average election polling layer (GAEPL) to replace the fully connected layer. They claim that it can reduce the dimensionality from three-dimensional to one-dimensional, which can overcome over fitting. Further, the implementation of convolutional neural networks is widely applied, including in their research in the form of customized CNN for natural language recognition [6]. They apply the CNN bilinear method, which consists of two CNN branches functioning as feature extractors, while the output vectors are pooled bilinearly via an outer function. Although in general, many proposed methods run slowly, mostly because their developed networks are associated with many parameters, leading to wider or deeper networks [7]. Several research methods have emerged to overcome the slow and large network on completing tasks. One of them has become very popular recently, being knowledge distillation (or KD for short) proposed by Hinton et al. [8], which offers the concept of the teacher–student paradigm to carry out knowledge transfer (or KT for short) from the teacher model. (i.e., cumbersome networks) to an untrained small student network using KD. This technique is a development by Ba and Caruana [9]. They demonstrated the ability of a shallow net compression model to approximate a trained SOTA deep model using the same parameters to mimic the original

model. This model is not trained directly on the original labeled data, but is trained to approximate the functions that have been studied by a more complex and larger model or a high-capacity model. In addition, recent research in [10] tries to improve the accuracy of KD by using a new approach with contrastive objectives for representation learning. They show better accuracy results than some previous methods by using different variation network architectures.

Some previous research approaches originally used method [8] to transfer knowledge from the teacher model, perhaps formed from several complex networks to a small student network by minimizing the KL divergence outputs between the teacher and student models. They later tried to solve this knowledge transfer problem with different techniques, one of which is DML [11], combining two or more networks that learn from each other simultaneously to solve common tasks without the help of a network teacher. This technique uses two losses in the training process: a conventional supervised learning loss for each network and a KLD-based one as the mimicry loss to align the probability estimates of each pair. The test accuracy of this technique is better than that of the KD using the experimental setting in [12], but the accuracy is much worse when using the experimental setting in [10]. To overcome this problem, we use the concept of (i) two or more student networks, each of which has a transfer of knowledge from a teacher network to teach each other simultaneously (FDDML), and (ii) one student network has a transfer of knowledge obtained from a powerful (wide and/or deep) teacher network that can teach each other student network simultaneously (HDDML) using the concept of mutual learning. Both of our approaches still use the teacher–student paradigm.

In this paper, we propose a combination of both KD and DML methods to improve the accuracy of student networks in carrying out the tasks to solve KT problems. We offer two techniques, namely full deep distillation mutual learning and half deep distillation mutual learning, and we measure the network accuracy with a learning metric in the form of a confusion matrix.

To sum up, our contributions of this paper are follows:

- Developing a new approach (FDDML and HDDML) that combines the two methods DML and KD into a formula to improve the performance of DML with adopting three losses by using variations of existing network architectures to improve the network performance;
- Exploring the effect of variations in the number of batch size on knowledge transfer from a teacher model has been trained with the original sample size in the TinyImageNet dataset to several untrained students with a downsampled size of 32×32;
- We show the effectiveness of our approach with Cinic-10 that two different batch size includes 64 and 128.

There are many related methods that have been proposed regarding knowledge distillation using knowledge transfer and mutual learning, which are summarized as follows.

Knowledge transfer. Since Ba and Caruana [9] published the results of their research on the compression model, their method has become popular. The idea is to find a way to improve the accuracy of simple network architecture at a low cost so that it can approximate a cumbersome network or a network architecture of larger or more complex architecture requiring a larger cost. Furthermore, the idea of the compression model is improved by Hinton et al. [8] via a new approach, namely knowledge distillation. It uses soft probabilities to transfer knowledge from a teacher network (i.e., larger or more cumbersome or complex model architecture) to a simpler and less expensive untrained student network using a temperature (T) that can be changed. Then, Zhang et al. [11] proposed a new method, inspired by the method from [8], namely "attention map" which is a response pattern obtained from the teacher and student feature maps, with better results reported than KD. Tung et al. [13] proposed another form of KD using $b \times b$ similarity matrices from the activation maps, generated from a teacher and student networks. Here b is the dimension of the input mini-batch of images whose accuracy performance has surpassed several previous methods using a variety of network architectures, for the teacher and student models. Later,

Peng et al. [14] proposed a new distillation framework method using correlation congruence to transfer correlation knowledge between instances to the student network. In addition to computer vision, KD is also used in the field of automatic speech recognition [8]. Their research can improve frame classification accuracy using the WER dataset. Meanwhile, [15] used KD to enhance the accuracy of the acoustic model performance ensemble for CTC-attention shared end-to-end speech recognition.

Mutual learning. Zhang et al. [11] has proposed the idea of mutual learning (or ML in short) as an alternative for learning to keep pace with the recent popularity on knowledge distillation neural networks. The distillation method works in the one-way direction where knowledge from a large network as a teacher can be transferred to a simple or compact network as a student. On the other hand, ML works on two or more student networks collaboratively during the training process. Two losses are used for each student, namely a cross entropy loss as a supervised loss and a KL divergence loss as a mimicry loss. Yao and Sun [16] developed a KT research in the form of a dense cross, based on ML by inserting an auxiliary classifier during the training process between a teacher and a student. The well-designed auxiliary classifier function makes it easier for this framework to work optimally by not only considering the probabilistic predications in the last layer but also the hidden layers for each network. Another experiment performed by Park et al. [17] shows that the transfer of mutual data examples can increase the accuracy of the student network significantly.

In contrast to all existing methods, we design a novelty approach which is similar to the deep mutual learning (or DML in short) method, that is, apart from training two or more untrained students collaboratively, we also add knowledge distillation from a high-capacity teacher to teach the untrained students during the training process, which can further improve the accuracy achieved by the student networks. Table 1 summarizes of recent related works.

Table 1. Comparison of recent related works.

Ref No.	Dataset Name	Number of Images Used	Method	GPU's
[8]	MNIST, JFT,	60,000, 100 million labeled images,	KD	No Information
[9]	TIMIT phoneme recognition, CIFAR-10	6300 sentences(5.4 h) 60 K images	Model Compression	Nvidia GTX 580 GPUs
[10]	CIFAR-100, ImageNet, STL-10, TinyImageNet	60 K images, 1.2 million images from 1 K classes for training and 50 K for validation, Training set of 5 K labeled images from 10 classes and 100 K unlabeled images and a test set 8 K images, 120 K images	Contrastive Representation Distillation	Two Titan-V GPUs
[17]	CUB-200-2011, Cars 196	200 images, 196 images	Relational Knowledge distillation	
[14]	CIFAR-100, ImageNet	60 K images, 1.2 million images from 1 K classes for training and 50 K for validation	Correlation Congruence for Knowledge Distillation	16 TiTAN X
[13]	CINIC-10, CIFAR-10	270 K images, 60 K images	Similarity-Preserving Knowledge Distillation	No information
[16]	CIFAR-100, ImageNet	60 K images, 1.2 million images from 1 K classes for training and 50 K for validation	Knowledge Transfer via Dense Cross-Layer Mutual-Distillation	GPUs
[11]	CIFAR-100, Market-1501	60 K images, 1501 images	DML	Single NVIDIA GeForce 1080 GPU
Ours	CIFAR-100, TinyImageNet, Cinic-10	60 K images, 120 K images 270 K images	FDDML HDDML	Single NVIDIA GeForce 1080 GPU

2. Materials and Methods

In this section, we describe the detailed formulation and implementation of our approach, which is different from previous methods.

2.1. DML and KD

In this section, we are the first to combine both DML [11] and KD [8]. In Deep Convolutional neural network, Given the N samples of training data $D = \{x_i\}_{i=1}^{N}$ from a collection of C classes, with $Y = \{y_i\}_{i=1}^{N}$ as a label set and $y_i \in \{1, 2, 3, \ldots, C\}$. The idea of this DML is to use two simpler networks to learn together as a knowledge transfer technique between one another without using a pre-trained teacher to improve the classification accuracy of the two or more networks. In addition to using cross-entropy loss as a supervised learning loss to predict the correct labels for training instances, this techniquealso uses KL-divergence loss to equalize the estimated probability between pairs and measure [11] the match of predictions p_1 and p_2. The supervised learning loss formula [18] that we use is:

$$L_s = -\sum_{i=1}^{N} y_i \log(p(x_i)) + (1 - y_i) \log(1 - p(x_i)) \tag{1}$$

where p is the probability of the class and y_i is the corresponding class label. Figure 1 shows that both G_1 and G_2 networks produce outputs in the softmax layer in the form of logit z^c. So, the probability of class C from sample x_i is defined as:

$$p^c(x_i) = \frac{e^{(z^c)}}{\sum_{m=1}^{M} e^{(z^c)}} \tag{2}$$

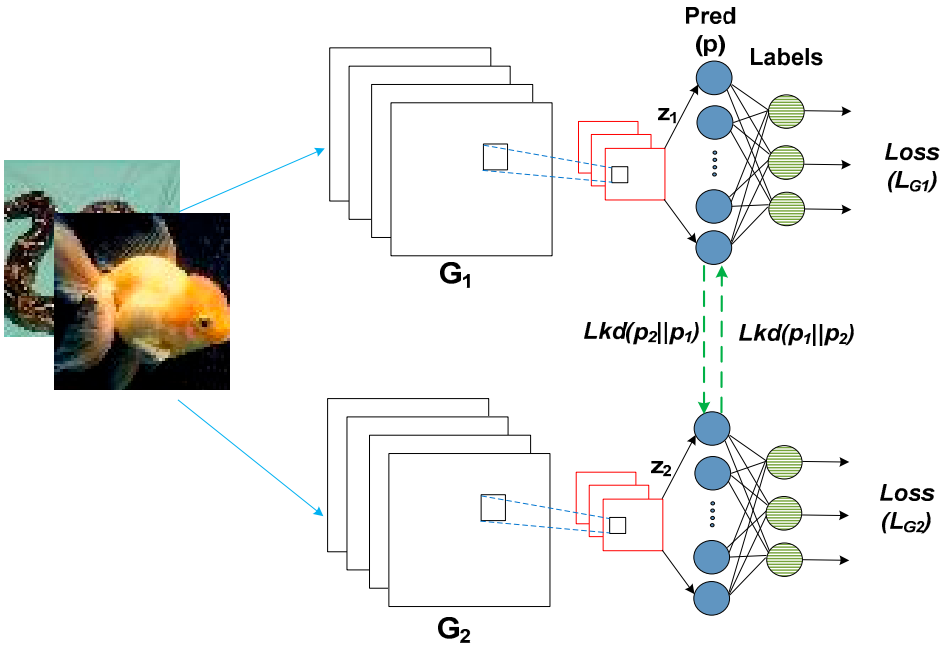


Figure 1. Deep mutual learning (DML) diagram [11] that the network G_1 and G_2 trained simultaneously together, and using a KLD-based.

Then, the KL divergence formula is used to calculate the mimicry loss to denote p_1 and p_2 predictions of each network, which is defined as:

$$L_{kd}(p_2 \parallel p_1) = \sum_{N=1}^{N} \sum_{c=1}^{C} p_2^c(x_i) \log \frac{p_2^c(x_i)}{p_1^c(x_i)} \tag{3}$$

Thus, the entire loss function for the G_1 network using the formula can be calculated as:

$$L_{G1} = L_{s1} + \lambda L_{kd}(p_2 \parallel p_1) \tag{4}$$

where $\lambda = 1$, is a weight factor. Similarly, for network G_2 that the entire loss function can be calculated as:

$$L_{G2} = L_{s2} + \lambda L_{kd}(p_1 \parallel p_2) \tag{5}$$

Knowledge distillation in [8] is one of the most popular knowledge transfer methods today, and it uses a teacher–student framework. The basic idea of this method is that a pre-trained teacher network (i.e., a cumbersome network or the biggest network) using certain hyperparameters are then used to train an untrained student network (i.e., small network) for the purpose of transferring knowledge. This process uses a distillation knowledge equation where a temperature (T) is involved and can be varied to obtain a soft probability output from a class C image which can be calculated as:

$$P^c(x_i) = \frac{e^{(z_i^c/T)}}{\sum_{c=1}^{C} e^{(z_i^c/T)}} \tag{6}$$

Suppose the teacher network is marked as G_t and the student network is G_s, then the distillation loss can be defined as:

$$L_{kld}(P_t, P_s) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} P_t^c(x_i) \log P_s^c(x_i) \tag{7}$$

As a result, the student loss function contained in Figure 2 is minimized during the training process based on (6) and (7) as:

$$L_{G_s} = L_s(\mathrm{G_s}, \mathrm{D}, \mathrm{Y}) + \lambda L_{kld}(P_t, P_s) \tag{8}$$

where λ is a balancing value between the two losses. The main purpose of the teacher–student framework is to force the student output probability to imitate or match the pre-trained teacher network's probability output.

2.2. Full Deep Distillation Mutual Learning

Inspired by the concepts of DML [11] and KD [8], we developed a new approach that combines the two methods into a formula to improve the performance of DML. If the concept used by DML is to pair two or more networks in the form of a cohort that aims to conduct training simultaneously by utilizing KL divergence loss to guide another network to increase the posterior entropy of each student. As a result, the process can converge to the minima more reliably, and a cohort can be the same small network and can also be various network pairs or even peer between several large and small networks. Note that this does not require a pre-trained teacher to improve the performance of a single student as was performed in previous studies using KD [8]. Therefore, we propose the full deep distillation mutual learning (FDDML) method as shown in Figure 3, which still uses the teacher–student framework.

Our proposed method adopts more than two KL divergence to improve the network performance. In the first stage, the teacher knows how to reduce the cross-entropy loss with the hyperparameters that we have determined (more details are given in the next Section) to produce a pre-training model, just like how the knowledge distillation method works in [8]. Then a cohort of untrained student networks using the DML concept in [11] work with a simultaneous training process where each student network deals with three losses.

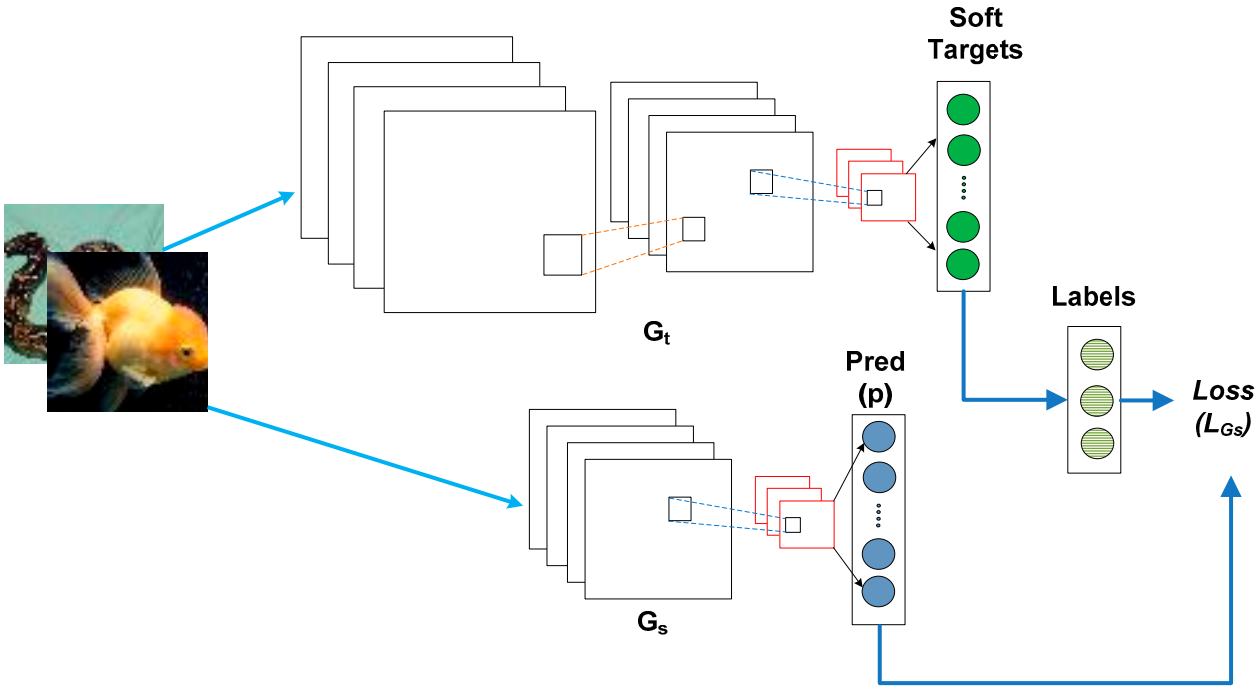


Figure 2. Training teacher–student framework using knowledge distillation (KD).

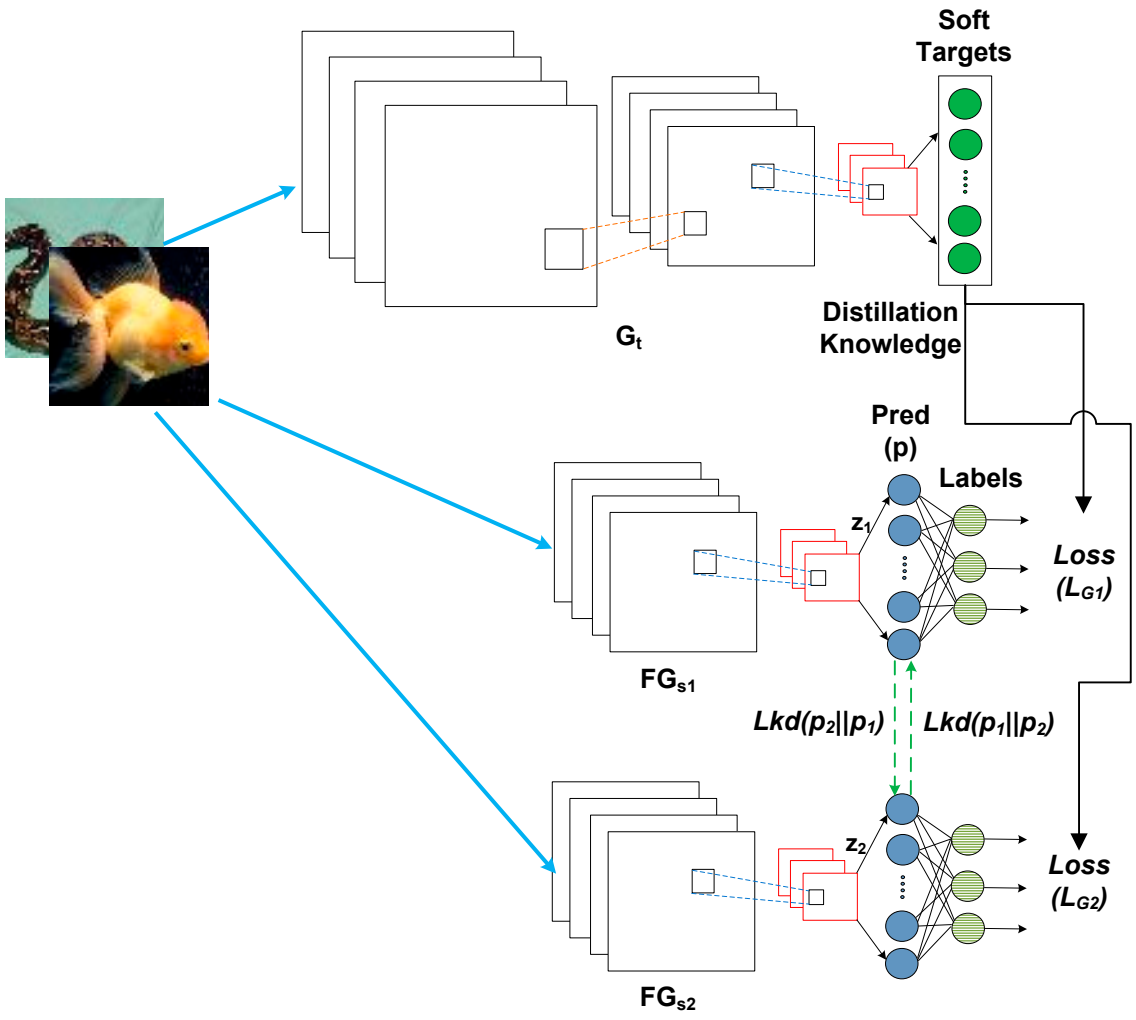


Figure 3. Full deep distillation mutual learning(FDDML) schematic.

The first one is a cross-entropy loss, used as a classification loss. The second one is a KL divergence, as a mimicry loss, to adjust each student's posterior class to suit the needs of another probability class student. The third one is a KL divergence, as knowledge distillation loss, to transfer knowledge from a pretrained large teacher network to a cohort student network (i.e., a pool of small networks). As a result, FDDML is trained to minimize L_{FGs1} for the first student network's loss.

$$L_{FG_{s1}} = L_{s1}(\mathrm{FG}_{s1}, \mathrm{D}, \mathrm{Y}) + \lambda L_{kld}(P_t, P_{s1}) + \beta L_{kd}(p_2 \parallel p_1) \tag{9}$$

Similarly, for their peer student network L_{FGs2} loss:

$$L_{FG_{s2}} = L_{s2}(\mathrm{FG}_{s2}, \mathrm{D}, \mathrm{Y}) + \lambda L_{kld}(P_t, P_{s2}) + \beta L_{kd}(p_1 \parallel p_2) \tag{10}$$

where λ and β are weight balancing factors these three loss terms that we set to become 1. While the temperature (T) that we use for each student network is 4.

2.3. Half Deep Distillation Mutual Learning

Furthermore, the second proposed method is half deep distillation mutual learning, as shown in Figure 4. In this method, we use the knowledge distillation method only on the H_{Gs2} student network, while we leave the H_{Gs1} student network as usual by only interacting with the H_{Gs2} network. As a result, the loss for H_{Gs1} can be calculated as:

$$L_{HGs_1} = L_{s1} + \lambda L_{kd}(p_2 \parallel p_1) \tag{11}$$

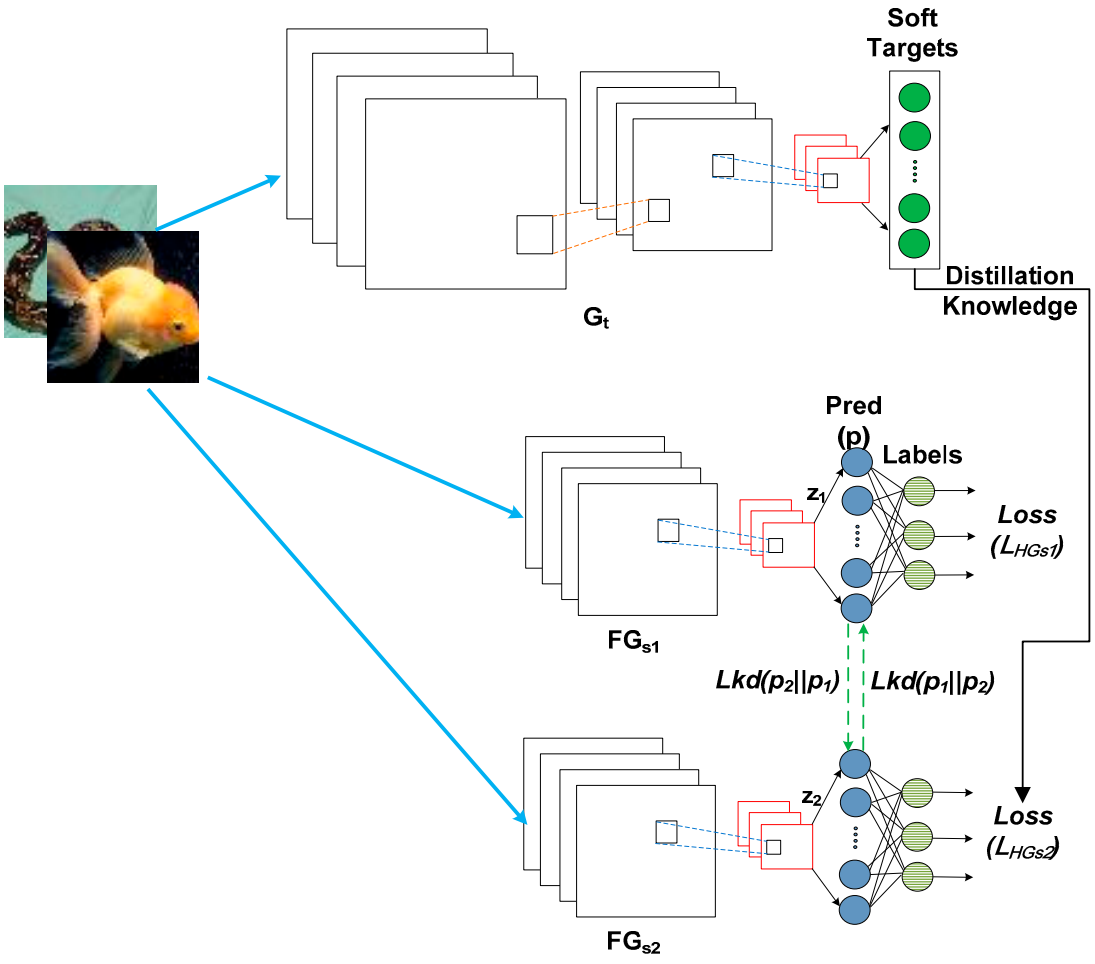


Figure 4. Half deep distillation mutual learning (HDDML) schematic.

Plus, a loss for G_{s2} can be defined as:

$$L_{HG_{s2}} = L_{s2}(\mathrm{HG}_{s2}, \mathrm{D}, \mathrm{Y}) + \lambda L_{kld}(P_t, P_{s2}) + \beta L_{kd}(p_1 \parallel p_2) \quad (12)$$

3. Results and Discussion

In this section, we describe the results of the evaluation of several comparisons with existing methods, including KD [8], DML [11], attention transfer (or AT for short) [19], similarity preserving (or SP for short) [13], correlation congruence (or CC for short) [14], and, most recently, contrastive representation distillation (or CRD for short) and CRD+KD [10]. Next, we show the effectiveness of our proposed method by comparing it with DML, especially in terms of increasing the number of networks. We only limit the number of networks to be four in the cohort. Then we use the batch size and temperature (T) variations to demonstrate the reliability of our approach in transferring knowledge between teacher models with the originally trained size to several untrained students with downsampled sizes on the TinyImageNet and only batch size variations on cinic-10 dataset.

3.1. Dataset

The dataset that we use in this experiment is the CIFAR-100 [20], which consists of 100 classes with a size of 32 × 32 color images drawn, which are divided into 10,000 testing images and 50,000 training images. Plus, the second dataset that we chose is TinyImageNet [21], which consists of 120,000 images divided into 200 classes with a size of 64 × 64 color images drawn, with a total of 500 training images and 50 testing images per class. This second dataset is larger both in number and dimensions, so it can show the reliability of the proposed method. Then, we use Cinic-10 dataset [22] as a third dataset consists of 270,000 images as an extension of CIFAR-10 [20] by combined with images chosen from ImageNet [23] and converted to 32 × 32 pixel images.

3.2. Network Architectures

3.2.1. On CIFAR-100 Training and Testing

For MobileNetV2 [24], we use a multiplier of 0.5. For VGG [25], we use its original ImageNet. For wide residual network [12], we use the width factor w and depth d. For Resnet [7], we use Resnet8 × 4 and Resnet32 × 4, indicating a 4 times wider network (64, 128, 256 channels for each of the block) and Resnet-d to represent cifar-style Resnet with three groups of basic blocks. For ShuffleNetV2 [26], we adapt them to input 32 × 32.

3.2.2. On TinyImageNet 64 × 64 Image Size Training and Testing

For Resnet [7], which represents ImageNetstyle Resnet using bottleneck blocks and more channels. For VGG [25], we use their original ImageNet-style. In this study, we use vgg16, vgg13, and vgg8.

3.2.3. On TinyImageNet 32 × 32 DownsampledImage Size Training and Testing

For Resnet [7], we use Resnet-d with three groups of basic blocks and Resnet 8 × 4 with a four times wider network to input TinyImageNet with downsampled size input of 32 × 32. For wide residual network [12], we use downsampled size input of 32 × 32.

3.2.4. On Cinic-10 32 × 32 image Size Training and Testing

We use wide residual network [12], with wide Resnet with width factor w and depth d. Resnet [7], and ShuffleNetV2 [26] that the input dimension as 32 × 32.

3.3. Implementation Details

In all of these experiments, we used the NVIDIA Geforce GTX 1080 GPU and used the Ubuntu 16.04 operating system, while the training and testing procedures were running using PyTorch 1.0 version [27]. To maintain fairness during comparisons, we follow the settings used in [10]. We train all models with 240 epochs for both CIFAR-100 and

TinyImageNet. For CIFAR-100, we used the optimization with SGD, initial learning rate of 0.1, momentum of 0.9, weight decay of 5×10^{-4}, and the batch size of 64. For TinyImageNet, we used the initial learning rate of 0.01, batch size of 40, momentum, and weight decay to be the same as in CIFAR-100, as well as using the SGD optimizer. Both types of experiments adopt data augmentation, including random crops and horizontal flips. Then, we use Cinic-10 datasets has the same hyperparameter as the two previous datasets except two different batch size include 64 and 128.

3.4. Experiment on CIFAR-100

We use three experimental scenarios to demonstrate experimental results on the CIFAR-100 dataset. The first scenario uses the same type of architecture network for the student network, while for a teacher network, we use a slightly more complex type of architecture in terms of the number of parameters, as shown in Table 2, the accuracy results of our approach outperform the existing methods, except the case where Resnet 32×4 as a teacher and Resnet8 $\times$ 4 as a student shows better accuracy in CRD+KD for being 75.53%. Meanwhile, the comparison between the two methods that we propose shows that FDDML dominates over HDDML. On the CIFAR-100 dataset, learning between two students together improves accuracy with the support of directions from knowledge transfer by a teacher to increase the accuracy of these students. When compared with KD and DML, the accuracy of our method far exceeds them, and even the accuracy of the students can exceed the accuracy of the teacher.

Table 2. Comparing resultstop-1 accuracy on the CIFAR-100 datasets with twin student network.

	Network Architecture							
Teacher	Resnet32	Resnet56	Resnet110	Resnet110	WRN-40-2	Resnet32x4	WRN-40-2	Resnet56
Student 1	Resnet20	Resnet20	Resnet56	Resnet20	MobileNetv2	Resnet8x4	WRN-16-2	Resnet44
Student 2	Resnet20	Resnet20	Resnet56	Resnet20	MobileNetv2	Resnet8x4	WRN-16-2	Resnet44
	Test Accuracy (%)							
Teacher	70.12	72.04	73.79	73.79	75.3	76.95	75.3	72.04
Student 1	69.05	69.05	72.04	69.05	64.55	71.14	73.15	71.54
Student 2	69.05	69.05	72.04	69.05	64.55	71.14	73.15	71.54
KD [8]	70.5	70.33	74.41	70.88	68.51	75.24	74.76	73.87
AT [19]	69.42	70.91	74.54	69.95	59.27	72.46	73.02	71.46
SP [13]	70.16	69.71	73.83	69.65	64.82	73.12	73.74	73.54
CC [14]	68.78	68.45	71.89	69.06	65.91	72.4	72.97	72.06
CRD [10]	70.95	70.63	75.1	70.86	69.62	75.14	75.42	74.36
CRD + KD [10]	71.16	71	74.99	70.92	69.78	**75.53**	75.54	74.43
DML: [22]								
Student 1	69.98	69.98	73.39	69.98	67.31	71.58	73.74	72.14
Student 2	69.71	69.71	73.26	69.71	66.88	71.72	73.96	72.8
FDDML:								
Student 1	70.6	70.9	75	**71.04**	**70.63**	75.27	75.21	74.31
Student 2	**71.17**	**71.16**	74.9	70.85	69.88	74.99	**75.75**	**74.49**
HDDML:								
Student 1	70.71	70.87	73.7	69.98	68.75	73.81	74.56	74.31
Student 2	70.18	68.46	**75.11**	70.79	70.14	74.52	75.43	72.88

Next, in the second scenario, we use a method similar to the one in the original DML test, namely using different student network architectures whose accuracy results are shown in Table 3, it indicates a significant increase in accuracy, such as in the test using WRN-40-2 as a teacher and MobileNetv2 and Resnet32 as the first and second students, with an accuracy of 73.85% for students. Note that the accuracy of the second student

outperforms the accuracy of FDDML and DML, where only the maximum accuracy for the second student is shown, to be 73.0% and 71.84%, respectively. This test also shows better accuracy when compared with the same network architecture of students in Table 2, where WRN-40-2 as a teacher and MobileNetv2 as the first and second students can only obtain the highest accuracy on the first and second FDDML students, namely 70.63% and 69.88%. Next, in the last scenario, we take the same test steps as [22], namely with a larger student cohort, to observe the training process of multiple student networks by increasing the number of networks to demonstrate the advantages of our approach, as shown in Figure 5.

Table 3. Comparing resultstop-1 accuracy on the CIFAR-100 datasets with different student network.

Network Architecture							
Teacher Student 1 Student 2	WRN-40-2 ShuffleV2 Resnet32	WRN-40-2 WRN-28-2 Resnet32	WRN-40-2 MobileNetv2 Resnet32	WRN-40-2 Resnet32x4 MobileNetv2	Resnet110 Resnet56 WRN-16-2	Resnet56 Vgg8 Resnet32	Vgg13 ShuffleV2 MobileNetv2
			Test Accuracy (%)				
Teacher	75.3	75.3	75.3	75.3	73.79	72.04	74.94
Student 1	71.88	73.89	64.55	71.14	72.04	69.83	71.88
Student 2	70.12	70.12	70.12	64.55	73.15	70.12	64.55
DML: [11]							
Student 1	76.71	75.58	69.13	77.8	74.03	72.73	73.81
Student 2	72.34	72.72	71.84	68.2	74.14	72.37	67.72
FDDML:							
Student 1	**77.31**	**77.04**	69.93	**78.68**	74.55	**73.61**	**75.71**
Student 2	73.23	73.52	73	70.5	74.91	73.04	69.82
HDDML:							
Student 1	76.29	75.83	69.55	77.6	74.03	72.3	74.24
Student 2	74.23	73.32	**73.85**	70	**74.95**	73.58	69.54

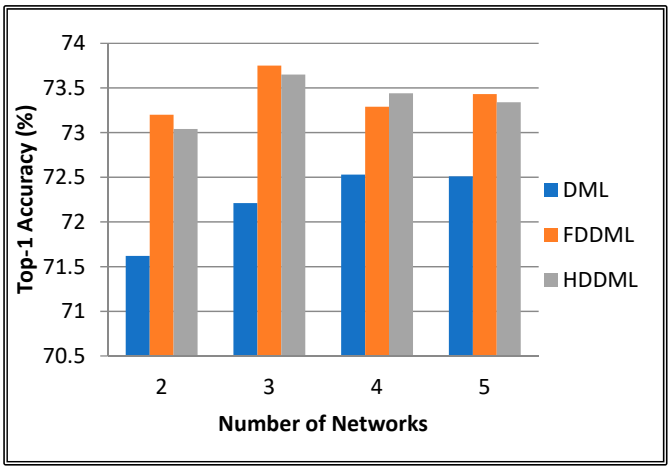


Figure 5. Accuracy performance (top-1) on CIFAR100 with teacher WRN-40-2 to different numbers of Resnet32 student networks in a cohort.

By simultaneously training multiple students, the accuracy results increase when the number of student networks is added to the cohort. This test scenario uses WRN-40-2 as the teacher network and Resnet32 as the student network for our two proposed methods, while DML is the baseline as in the original form without using the soft probabilities of the teacher network.

In all experiments, the variation in the number of additions to the student network shows the superiority of our method, especially when the total number of student networks being three, where the highest accuracy was achieved by FDDML with 73.75%, followed

by HDDML with 73.65%, but this trend of accuracy gradually decreases as the number of students increased in a cohort.

On the other hand, for DML itself, the trend of accuracy is increasing but is not better than our proposed method. This shows that the minimization of mimicry loss for the transfer of teacher knowledge to a pool of student networks trained simultaneously can work better.

3.5. Experiment on TinyImageNet

Then, to see if our method is robust or not, we conducted a test with a larger dataset using TinyImageNet with the original size of 64 × 64 images. In this test, we use two training scenarios. In the first scenario, we use the dataset of the original size, as shown in the test results as follows.

In contrast to the testing using the CIFAR-100 dataset, in the second scenario, we show the accuracy of our approach in Table 4, where the highest accuracy is generally by HDDML, and one CRD+KD [10] process is dominant over other tests, that is, when VGG16 is the teacher and VGG8 is a student, which is marked in bold, followed by the second-highest accuracy marked with underline. It can be seen that the performance of our proposed methods can exceed the accuracy produced by the teacher network, also shown in the existing method [10], which is able to beat the accuracy of the teacher network, but it is not better than HDDML, one of the two methods we propose. Then, to ensure the performance of our proposed method, we tested it by downsampling the TinyImageNet datasets to become 32 × 32.

Table 4. Comparing results top-1 accuracy on the TinyImageNet datasets with 64-Image size.

	Network Architecture					
Teacher	Resnet50	Resnet50	Resnet50	Vgg13	Vgg16	Vgg16
Student 1	Resnet34	Resnet18	Vgg16	Vgg8	Vgg13	Vgg8
Student 2	Resnet34	Resnet18	Vgg16	Vgg8	Vgg13	Vgg8
	Test Accuracy (%)					
Teacher	55.34	55.34	55.34	48.1	49.2	49.2
Student 1	37.94	36.04	49.2	41.68	48.1	41.68
Student 2	37.94	36.04	49.2	41.68	48.1	41.68
KD [8]	55.26	55.72	53.92	44.42	52.4	44.68
AT [19]	50.13	50.22	45.68	44.84	48.92	43.18
SP [13]	55.38	55.68	54.28	45.06	51.88	44.66
CC [14]	50.9	49.82	48.72	42.1	48.08	41.68
CRD [10]	57.48	55.92	55.82	45.12	51.1	45.38
CRD + KD [10]	57.14	<u>56.94</u>	55.74	<u>46.08</u>	51.98	**46.88**
DML: [11]						
Student 1	55.1	54.86	53.88	44.94	52.84	44.94
Student 2	55.13	54.92	53.26	45.16	<u>53.76</u>	45.16
FDDML:						
Student 1	57.33	56.58	56.12	45.02	53.22	<u>46.3</u>
Student 2	<u>57.98</u>	55.88	<u>56.54</u>	44.52	53.1	44.36
HDDML:						
Student 1	57.05	55.1	53,66	46.06	52.32	46.26
Student 2	**58.27**	**57.48**	**56.68**	**46.26**	**54.24**	45.74

The downsampled images carried out in this experiment are intended to demonstrate the reliability of our proposed method in adapting to different conditions and to illustrate the results of comparisons with similar methods [11] as a baseline.

Although it appears that the student network accuracy performance has decreased due to the changes in image size and image resolution compared with the cases when trained with the original image size, it can adapt well when compared with the baseline.

Table 5 shows that the method we propose is still superior to the accuracy of teachers in general. The last scenario in the experiment is that we adapt our proposed method to the transfer of knowledge of student network trained using a downsampled image from the original image for being an example of a previously trained exemplary teacher. This test involves variations in temperatures and variations in batch sizes.

Table 5. Comparing results top-1 accuracy on The TinyImageNet datasets resize with 32-image size.

	Network Architecture					
Teacher	WRN-40-2	Resnet50	Resnet34	WRN-40-2	Resnet50	Resnet50
Student 1	Resnet34	Resnet18	Resnet18	WRN-16-2	Resnet34	Resnet8x4
Student 2	Resnet34	Resnet18	Resnet18	WRN-16-2	Resnet34	Resnet8x4
	Test Accuracy (%)					
Teacher	31.06	33.26	29.96	31.06	33.26	33.26
Student 1	29.96	29.68	29.68	28.32	29.96	29.42
Student 2	29.96	29.68	29.68	28.32	29.96	29.42
DML: [11]						
Student 1	30.92	32.06	32.06	28.8	30.92	30.12
Student 2	32.86	30.94	30.94	29.22	32.86	29.46
FDDML:						
Student 1	**33.56**	**33.77**	32.44	29.54	**33.18**	30.06
Student 2	32.62	33.66	31.94	30.06	32.76	29.9
HDDML:						
Student 1	32.3	31.52	31.78	**30.8**	31.52	29.7
Student 2	33.02	33.5	**32.64**	29.54	32.68	**31.18**

In Table 6, the highest accuracy is marked in bold for each method. Here, we use two types of networks, namely, Resnet50 as a teacher and WRN-16-2 as a student, with an accuracy of 55.34% and 28.32%, respectively. The results are the accuracy of the teacher model using the original image size of 64 × 64 and the accuracy of the independent student using the downsampled image size of 32 × 32 image. It can be seen from the comparisons that this method tries to approximate the accuracy of a pre-trained teacher, and we use variations in temperatures and batch sizes to see how well the existing and proposed methods of knowledge transfer are able to adapt to the downsampled dataset.

Although almost all of the involved methods are not able to approach or even exceed the individual student accuracy, except for HDDML, which is one of our proposed methods with an accuracy of 28.36% (student 1) with the hyperparameter batch size of 128 at temperature 5, and batch size of 64 at temperatures 5 and 6, the accuracy only reached 28.20% and 28.12% for student 1, respectively.

It turns out, the results of this experiment show that all existing methods and our proposed method suffer from using downsampled size data because there is a lot of information loss due to the changes in image resolution so that it is not optimal for conducting a training to classify test sample images, even though it is assisted by a pre-trained teacher with soft probabilities.

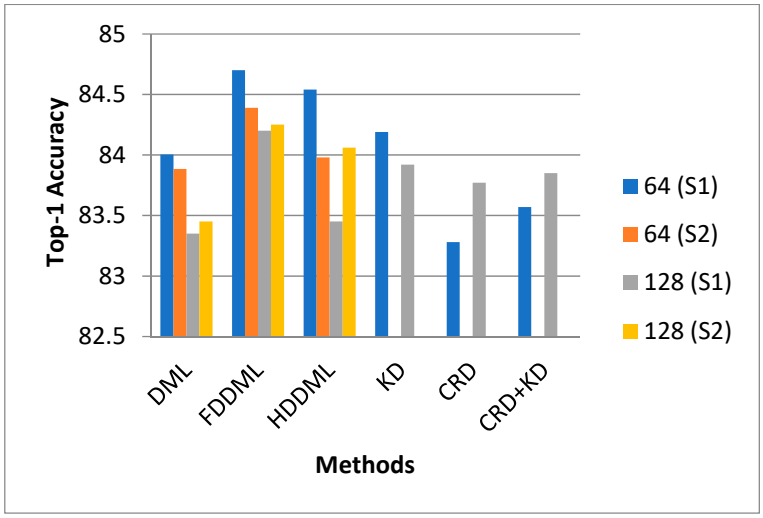
3.6. Experiment on Cinic-10

In addition to verifying our proposed method, we use the dataset, namely Cinic-10. The dataset section that we use consists of two parts, namely the training set and the testing set. We use the Resnet20 network, Shufflev2 as a different student and the WRN-40-2 network as the teacher, each of which has an accuracy of Resnet20: 81.42%, Shufflev2: 74.03%, and WRN-40-2: 85.07%.

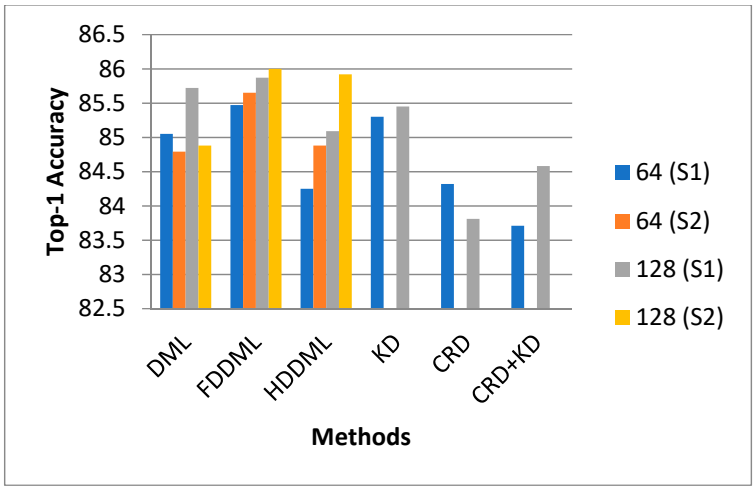
Table 6. Knowledge transfer from original images sample 64 × 64 model teacher pre-trained to train student with images downsampled to 32 × 32 using TinyImageNet dataset.

Temperature (T)	Network Architecture	KD	CRD	CRD+KD	FDDML	HDDML	Teacher	Student
				Batch size 64				
4	Resnet50 (T) WRN-16-2 (S) WRN-16-2 (S)	14.58	27.78	25.06	24.06 24.92	27.04 24.94		
5	Resnet50 (T) WRN-16-2 (S) WRN-16-2 (S)	14.52	27.16	25.56	24.08 24.96	28.2 25.82	55.34	28.32
6	Resnet50 (T) WRN-16-2 (S) WRN-16-2 (S)	15.22	27.74	25.52	25.34 24.84	28.12 26.62		
				Batch size 128				
4	Resnet50 (T) WRN-16-2 (S) WRN-16-2 (S)	14.18	26.92	25.52	23.6 24.28	23.22 23.94		
5	Resnet50 (T) WRN-16-2 (S) WRN-16-2 (S)	14.6	27.48	25.34	24.04 25.14	28.36 25.16	55.34	28.32
6	Resnet50 (T) WRN-16-2 (S) WRN-16-2 (S)	14.56	26.32	24.42	23.22 24.62	26.76 25		
				Batch size 180				
4	Resnet50 (T) WRN-16-2 (S) WRN-16-2 (S)	13.82	27.2	24.52	22.58 24.04	26.46 24.16		
5	Resnet50 (T) WRN-16-2 (S) WRN-16-2 (S)	13.9	27.22	25.52	24.34 23.96	27.46 23.86	55.34	28.32
6	Resnet50 (T) WRN-16-2 (S) WRN-16-2 (S)	13.78	26.08	24.68	23.94 24.58	27.48 25.62		

The experiment results in the form of top-1 accuracy are shown in Figure 6. Compared with the results from students for 5 different methods, namely DML, FDDML(ours), HDDML(ours), KD, CRD, and CRD+KD who were trained with two different batch sizes, namely 64 and 128. In Figure 6a, in batch size 64 FDDML achieved the highest accuracy of 84.70% followed by HDDML with an accuracy of 84.54% all obtained by student 1 (S1) while student 2 (S2) FDDML still outperformed the other four methods besides HDDML. While, when using batch size 128 all methods experienced a decrease in accuracy such as FDDML for (S1) was reduced by 0.5%, (S2) was reduced by at least 0.14% as well as DML (S1) decreased by 0.65%, (S2) 0.43%, and KD at least 0.27% while CRD and CRD+KD tend to increase in accuracy but not better than other methods. Then in Figure 6b, when using the ShuffleV2 network as a student, the accuracy of all methods tends to increase when using batch size 128 except for CRD, there is a decrease in accuracy of 0.51%. This section shows that the highest accuracy occurs in FDDML (S2) as much as 86%, then followed by HDDML (S2) with 85.92%.



(**a**)



(**b**)

Figure 6. Comparing top-1 accuracy on Cinic-10 datasets. (**a**) WRN-40-2 as a teacher and Resnet20 as a student when the batch size 64 and 128, and (**b**) WRN-40-2 as a teacher and ShuffleV2 as a student when the batch size 64 and 128.

4. Conclusions

In this study, the new approaches with knowledge transfer that we propose, namely FDDML and HDDML, that show the main key of teacher's knowledge are able to significantly improve the accuracy performance of a convolutional neural network with a teacher–student framework and can outperform several existing methods as well. We conduct experiments with three public benchmark datasets on images classification task. On the CIFAR-100 dataset, TinyImageNet and Cinic-10 show that the influence of a pre-trained teacher or teacher's knowledge is very large in the knowledge transfer student's network. On CIFAR-100 with twin student networks, FDDML obtained the highest accuracy of 75.75% for students and teachers WRN-16-2 and WRN-40-2, respectively. While for different students, the paired student of ShuffleV2 and MobileNetv2 with VGG13 as the teacher obtained 75.71% accuracy. Good results are also shown by HDDML twin students with vgg13 and vgg16 as a teacher with 54.24% accuracy on TinyImageNet. Likewise, when the dataset was resized to 32 × 32, FDDML with Resnet18 as a pair of students and Resnet50 as a teacher with the highest accuracy reached 33.77%. In the cinic-10 experiment, FDDML has the highest accuracy compared to other methods.For further research, we will explore

our proposed method for implementing general applications related to object recognition, incremental learning, and object tracking.

Author Contributions: Conceptualization, A.L. and C.-K.Y.; methodology, A.L.; software, A.L.; validation, A.L. and C.-K.Y.; formal analysis, A.L.; investigation, A.L.; resources, A.L.; data curation, A.L.; writing—original draft preparation, A.L.; writing—review and editing, C.-K.Y.; visualization, A.L.; supervision, C.-K.Y.; project administration, A.L. and C.-K.Y. and funding acquisition, C.-K.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Ministry of Science and Technology of Taiwan under the grants MOST 109-2221-E-011-133, MOST 109-2228-E-011-007, MOST 110-2221-E-011-099-MY3 and MOST 110-2221-E-011-095-MY3.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This research was trained and tested on open source CIFAR-100 [20], TinyImageNet [21], and Cinic-10 [22] datasets.

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

References

1. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
2. LeCun, Y.; Jackel, L.D.; Bottou, L.; Cortes, C.; Denker, J.S.; Drucker, H.; Guyon, I.; Muller, U.A.; Sackinger, E.; Simard, P.; et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural Netw. Stat. Mech. Perspect.* **1995**, *261*, 2.
3. Wu, M.; Chen, L. Image recognition based on deep learning. In Proceedings of the 2015 Chinese Automation Congress (CAC), Wuhan, China, 27–29 November 2015; pp. 542–546.
4. Kaur, P.; Harnal, S.; Tiwari, R.; Alharithi, F.S.; Almulihi, A.H.; Noya, I.D.; Goyal, N. A hybrid convolutional neural network model for diagnosis of COVID-19 using chest x-ray images. *Int. J. Environ. Res. Public Health* **2021**, *18*, 12191. [CrossRef] [PubMed]
5. Lilhore, U.K.; Imoize, A.L.; Lee, C.C.; Simaiya, S.; Pani, S.K.; Goyal, N.; Kumar, A.; Li, C.T. Enhanced Convolutional Neural Network Model for Cassava Leaf Disease Identification and Classification. *Mathematics* **2022**, *10*, 580. [CrossRef]
6. Singh, T.P.; Gupta, S.; Garg, M.; Gupta, D.; Alharbi, A.; Alyami, H.; Anand, D.; Ortega-Mansilla, A.; Goyal, N. Visualization of Customized Convolutional Neural Network for Natural Language Recognition. *Sensors* **2022**, *22*, 2881. [CrossRef] [PubMed]
7. 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.
8. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:150302531.
9. Ba, L.J.; Caruana, R. Do deep nets really need to be deep? *arXiv* **2013**, arXiv:13126184.
10. Tian, Y.; Krishnan, D.; Isola, P. Contrastive representation distillation. *arXiv* **2019**, arXiv:191010699.
11. 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.
12. Zagoruyko, S.; Komodakis, N. Wide residual networks. *arXiv* **2016**, arXiv:160507146.
13. Tung, F.; Mori, G. Similarity-preserving knowledge distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 1365–1374.
14. Peng, B.; Jin, X.; Liu, J.; Li, D.; Wu, Y.; Liu, Y.; Zhou, S.; Zhang, Z. Correlation congruence for knowledge distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 5007–5016.
15. Gao, Y.; Parcollet, T.; Lane, N.D. Distilling Knowledge from Ensembles of Acoustic Models for Joint CTC-Attention End-to-End Speech Recognition. In Proceedings of the 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Cartagena, Colombia, 13–17 December 2021; pp. 138–145.
16. Yao, A.; Sun, D. Knowledge transfer via dense cross-layer mutual-distillation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 294–311.
17. Park, W.; Kim, D.; Lu, Y.; Cho, M. Relational knowledge distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3967–3976.
18. Zhao, H.; Yang, G.; Wang, D.; Lu, H. Deep mutual learning for visual object tracking. *Pattern Recognit.* **2021**, *112*, 107796. [CrossRef]
19. Zagoruyko, S.; Komodakis, N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv* **2016**, arXiv:161203928.

20. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*. Tech Report. 2009. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.222.9220&rep=rep1&type=pdf (accessed on 23 July 2022).
21. Le, Y.; Yang, X. Tiny imagenet visual recognition challenge. *CS 231N* **2015**, *7*, 3.
22. Darlow, L.N.; Crowley, E.J.; Antoniou, A.; Storkey, A.J. Cinic-10 is not imagenet or cifar-10. *arXiv* **2018**, arXiv:181003505.
23. 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 IEEEConference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June2009; pp. 248–255.
24. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
25. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:14091556.
26. Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; Le, Q.V. Mnasnet: Platform-aware neural architecture search for mobile. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2820–2828.
27. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in pytorch. In Proceedings of the NIPS Workshop 2017, Long Beach, CA, USA, 9 December 2017. Available online: https://openreview.net/forum?id=BJJsrmfCZ (accessed on 23 July 2022).