Online Ensemble Model Compression using Knowledge Distillation

Devesh Walawalkar, Zhiqiang Shen, and Marios Savvides

Carnegie Mellon University, Pittsburgh PA 15213, USA
devwalkar64@gmail.com {zhiqians,marioss}@andrew.cmu.edu

Abstract. This paper presents a novel knowledge distillation based model compression framework consisting of a student ensemble. It enables distillation of simultaneously learnt ensemble knowledge onto each of the compressed student models. Each model learns unique representations from the data distribution due to its distinct architecture. This helps the ensemble generalize better by combining every model's knowledge. The distilled students and ensemble teacher are trained simultaneously without requiring any pretrained weights. Moreover, our proposed method can deliver multi-compressed students with single training, which is efficient and flexible for different scenarios. We provide comprehensive experiments using state-of-the-art classification models to validate our framework's effectiveness. Notably, using our framework a 97% compressed ResNet110 student model managed to produce a 10.64% relative accuracy gain over its individual baseline training on CIFAR100 dataset. Similarly a 95% compressed DenseNet-BC (k=12) model managed a 8.17% relative accuracy gain.

Keywords: Deep Model Compression, Image Classification, Knowledge Distillation, Ensemble Deep Model Training

1 Introduction

Deep Learning based neural networks have provided tremendous improvements over the past decade in various domains of Computer Vision. These include Image Classification [24,17,20,40], Object Detection [31,3,26,30,12], Semantic Segmentation [16,4,44,5,45] among others. The drawbacks of these methods however include the fact that a large amount of computational resources are required to achieve state-of-the-art accuracy. A trend started setting in where constructing deeper and wider models provided better accuracy at the cost of considerable resource utilization [35,38,27]. The difference in resource utilization is considerable compared to traditional computer vision techniques. To alleviate this gap, model compression techniques started being developed to reduce these large computational requirements. These techniques can broadly be classified into four types [7] i.e. Parameter Pruning [15,37,6,14], Low Rank Factorization [8,29,36], Transferred Convolutional Filters [9,33,25,11] and Knowledge Distillation methods

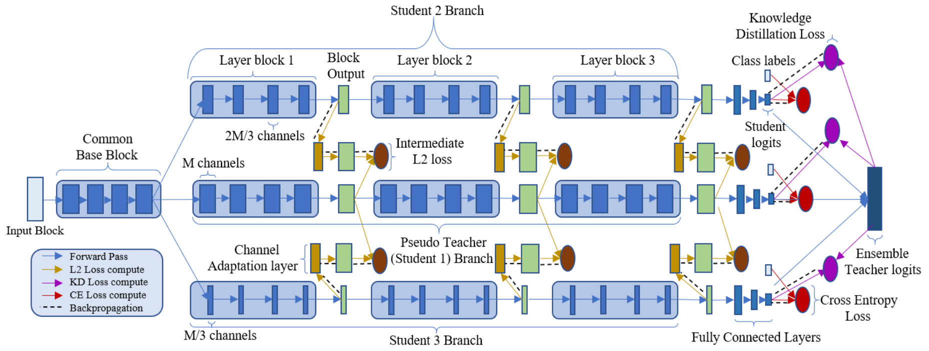


Fig. 1: Overview of our *model compression framework* for a *3 student ensemble*. Each student is composed of the base block and one of the network branches on top of it. The original model is the first student in the ensemble, termed as *pseudo teacher* due to its simultaneous knowledge transfer capability and training from scratch properties. The *ensemble teacher* is a weighted combination of all student's output logits. Each student is divided into four blocks such that each block incorporates approximately the same number of layers. The layer channels for the compressed student branches are reduced by a specific ratio with respect to the original model. For a 3 student ensemble, the layer channels assigned are M, 2M/3 and M/3, where M is the original layer channel count. *Channel adaptation layers* help map the compressed student's block output channels to the pseudo teacher's output channels in order to compute an intermediate feature representation loss

[18,21,1,13,22,46]. Each of these types was able to provide impressive computational reductions while simultaneously managing to keep the accuracy degradation to a minimum.

Knowledge Distillation (KD) in particular has provided great model compression capabilities using a novel teacher-student model concept [18]. Here a teacher, the original model trained for a specific task is used to teach a compressed or replicated version of itself referred to as student. The student is encouraged to mimic the teacher output distribution, which helps the student generalize much better and in certain cases leads to the student performing better than the teacher itself. Drawbacks of these methods include the fact that a pre-trained model is required to distill knowledge onto a student model. To solve this, simultaneous distillation methods [2,46] were developed wherein multiple students were being trained with an ensemble teacher learning on the fly with the students itself in an ensemble training scheme. These methods however primarily focused on replicating the teacher multiple times, causing the ensemble to gain sub-optimal generalized knowledge due to model architecture redundancy. Also in parallel, methods [22,34,1] were developed which focused on distilling not only the output knowledge but also the intermediate representations of teacher onto the student to have more effective knowledge transfer. These techniques are efficient, however still suffering from the drawback of requiring teacher model pre-training in addition to the student training.

In this paper, we present a novel model compression and ensemble training framework which improves on all aforementioned drawbacks and provides a new perspective for ensemble model compression techniques. We present a framework which enables multiple student model training using knowledge transfer from an *ensemble teacher*. Each of these student models represents a version of the original model compressed to different degrees. Knowledge is distilled onto each of the compressed student through the ensemble teacher and also through intermediate representations from a *pseudo teacher*. The original model is the first student in our ensemble, termed as pseudo teacher due to its simultaneous knowledge transfer capability and training from scratch property. Moreover this framework simultaneously provides multiple compressed models having different computational budget with each having benefited from every other model's training. Our framework facilitates the choice of selecting a student that fits the resource budget for a particular use case with the knowledge that every student provides decent comparable performance to the original model. Also, the ensemble training significantly reduces the training time of all student combined, compared to when they are trained individually. We specifically focus on image classification task while providing extensive experimentation on popular image classification models and bench-marked datasets.
Our contributions from this paper can be thus summarized as follows:

1. We present a novel ensemble model compression framework based on knowledge distillation that can generate multiple compressed networks simultaneously with a single training run and is completely model agnostic.
2. We provide extensive experiments on popular classification models on standard datasets with corresponding baseline comparisons of each individual student training to provide evidence of our framework's effectiveness.
3. We provide hyper parameter ablation studies which help provide insights into the effective ensemble knowledge distillation provided by our framework.

2 Related Works

2.1 Model Compression using Knowledge Distillation

Hinton et al. [18] introduced the concept of distilling knowledge from a larger teacher model onto a smaller compressed student model. Mathematically this meant training the student on softened teacher output distribution in addition to the traditional cross entropy with dataset labels. The paper argued that the teacher distribution provided much richer information about an image compared to just one hot labels. For e.g. consider a classification task of differentiating between various breeds of dogs. The output distribution of a higher capability teacher provides the student with the information of how alike one breed of dogs looks to the other. This helps the student learn more generalized features of each dog breed compared to providing just one hot labels which fails to provide any comparative knowledge. Also, in process of trying to mimic the distribution of a much deeper teacher model the student tries to find a compact series of transformation that tries to mimic the teacher's larger series of transformations. This inherently helps the student negate the accuracy loss due to compression. Impressively, in certain cases the student manages to outperform its teacher due to this superior generalization training. Further works extended this concept by using internal feature representations [1], adversarial learning [34] and inner product between feature maps [42].

2.2 Online Knowledge Distillation

The single practical drawback of Knowledge Distillation is the fact that a pretrained model is required to play the role of a teacher. This entails multiple sequential training runs on the same dataset. Anil et al. [2] came up with a method to train the student in parallel with the teacher termed as codistillation. Here the student is an exact replica of the teacher and roles of teacher and student were continuously interchanged between the models during training with one training the other iteratively. The primary distinguishing property between the models was their distinct parameter initialization. This enabled each model to learn unique features which were then distilled from one to the other as training progressed.

Zhang et al. [43] employed a unique KD loss, by using KL divergence loss between two model output distributions to penalize the differences between them. Each model's training involved a combination loss of KL divergence loss with other model's distribution and traditional cross entropy loss. Both models acting as teacher and student simultaneously were trained jointly in an online fashion.

Lan et al. [46] extended this online KD concept by having multiple replicas of a given model in a multi branch architecture fashion. Multi branch architecture designs became popular with the image classification models like Resnet [17], Inception [38,39] and ResNext [41]. In this paper, the multiple replicated models have a common base of block of layers with each model represented as a branch on top of this base with subsequent layer blocks from the original model architecture right until the final fully connected layers. The teacher in this concept was the combined output of all the student models in the ensemble. Each student model learnt from the ensemble joint knowledge represented by the teacher outputs. Our paper builds on this core concept however is fundamentally different as we incorporate compressed student branches, more efficient training procedure, incorporate intermediate representation distillation in addition to the final output distillation among others.

2.3 Intermediate Representation Knowledge Distillation

A separate branch of Knowledge Distillation focuses on training the student to mimic the intermediate representations obtained in form of feature maps from certain intermediate layer blocks within the teacher. This provides a more stricter learning regime for the student who has to focus not only on the final teacher output distribution but also on its intermediate layer feature maps. Romero et al. [1] provide one of the preliminary works in this direction by distilling a single intermediate layer's knowledge onto the student which they term as providing hint to the student. Mathematically, hint training involves minimizing a combination of L2 loss between the features maps at an immediate layer of the two models and the regular KD (KL divergence loss) between the output distributions.

Koratana et al. [22] extend this concept by comparing feature maps at not just one but multiple intermediate locations within the model which can be related to as multiple hint training. This method however again requires a pre-trained teacher model which might be time consuming and compute expensive for certain real world scenarios. Our method incorporates multiple hint training for all student models with respect to pseudo teacher, which is also the first student in the ensemble. A network's depth measures its function modeling capacity in general. Koratana et al. [22] compress the model by removing blocks of layers from the network which severely affects the network depth. Our work incorporates a more robust compression logic where the number of channels in every student layer are simply reduced by a certain percent, thus preserving the model depth.

3 Methodology

An overview of our ensemble compression framework is presented in Figure 1, which can be split up into three major sections for the ease of understanding. We would be going into their details in the following sections.

3.1 Ensemble Student Model Compression

First, the entire architecture of a given neural network is broken down into a series of layer blocks, ideally into four blocks. The first block is designated as a common base block and the rest of the blocks are replicated in parallel to create branches as shown in Figure 1. A single student model can be viewed as a series of base block and one of the branches on top of it. As previously mentioned, the original network is designated as the first student model, also termed as pseudo teacher. For every successive student branch the number of channels in each layer of its blocks is reduced by a certain ratio with respect to the pseudo teacher. This ratio becomes higher for every new student branch created. For example, for a four student ensemble and C being number of channels in pseudo teacher, the channels in other three students are assigned to be $0.75C$, $0.5C$ and $0.25C$. The students are compressed versions of the original model to varying degrees, which still manage to maintain the original network depth. The channels in the common base block are kept the same as original whose main purpose is to provide constant low level features to all the student branches.

The output logits from all the student models are averaged together to create the ensemble teacher logits. This ensemble teacher output distribution represents the joint knowledge of the ensemble. During inference stage, any of the individual

Table 1: Model size and test set accuracy comparison of every student in a five student ensemble, with their percent relative size in respect to the original model and their CIFAR10 test accuracies achieved using our ensemble framework.

Classification Model	**Student Model size and accuracy (%)**									
	First		Second		Third		Fourth		Fifth	
	Size	Accuracy	Size	Accuracy	Size	Accuracy	Size	Accuracy	Size	Accuracy
ResNet20 [17]	100.0	92.13	62.95	92.18	35.61	91.78	15.50	91.45	3.95	91.03
ResNet33 [17]	100.0	92.25	62.87	92.76	35.43	92.45	15.33	92.11	3.8	91.78
ResNet44 [17]	100.0	93.45	62.84	93.29	35.36	93.11	15.25	92.89	3.74	92.56
ResNet110 [17]	100.0	94.24	62.79	94.18	35.26	93.98	15.15	93.57	3.64	93.28
DenseNet (k=12) [20]	100.0	94.76	58.01	94.51	36.34	94.29	13.31	94.08	4.29	93.57
ResNext50 ($32 \times 4d$) [41]	100.0	96.03	62.10	95.95	35.60	95.84	16.09	95.69	4.76	95.47
EfficientNet-B0 [40]	100.0	98.20	64.11	98.13	35.94	98.01	18.88	97.84	5.43	97.57
EfficientNet-B2 [40]	100.0	98.41	65.16	98.35	37.87	98.23	18.13	98.02	4.69	97.88
EfficientNet-B4 [40]	100.0	98.70	64.43	98.59	36.27	98.47	16.40	98.23	4.91	98.14

student models can be selected from the ensemble depending on the computational hardware constraints. In case of lenient constraints, the entire ensemble can be used with the ensemble teacher providing inference based on the learnt ensemble knowledge. From our studies we find that having 5 students (inclusive of pseudo teacher) provides an optimal trade off between training time and effective model compression. Table 1 provides an overview of compressed student model sizes and their CIFAR10 [23] trained accuracies for a five student ensemble based on various classification model architectures.

3.2 Intermediate Knowledge Distillation

The intermediate block knowledge (feature map representation) is additionally distilled onto every compressed student from the pseudo teacher. This provides a more stricter training and distillation regime for all the compressed students. The loss in every student's representational capacity due to compression is countered, by making each student block try and learn the intermediate feature map of its corresponding pseudo teacher block. The feature map pairs are compared using traditional Mean Squared Error loss, on which the network is trained to reduce any differences between them. Since the number of feature map channels varies across every corresponding student and pseudo teacher block, an adaptation layer consisting of pointwise convolution (1×1 kernel) is used to map compressed student block channels to its pseudo teacher counterpart. Figure 2 (a) presents this idea in detail for an EfficientNet-B0 [40] based ensemble. The intermediate knowledge is transferred at three locations corresponding to the three blocks in every student branch as shown in Figure 1.

3.3 Knowledge Distillation Based Training

The overall ensemble is trained using a combination of three separate losses which are described in detail as follows:

Cross-Entropy Loss Each student model is trained individually on classical cross entropy loss [Equation 1, 2] with one hot vector of labels and student output logits. This loss helps each student directly train on a given dataset. This loss procedure makes the pseudo teacher learn alongside the compressed students as the framework doesn't use pretrained weights of any sort. It also helps the ensemble teacher gain richer knowledge of the dataset as it incorporates combination of every student's learnt knowledge. It additionally enables the framework to avoid training the ensemble teacher separately and is trained implicitly through the students. The output softmax distribution and combined normal loss can be expressed as follows,

$$X_{ijk} = \frac{exp(x_{ijk})}{\sum_{k=1}^{C} exp(x_{ijk})} \tag{1}$$

$$L^{Normal} = \sum_{i=1}^{S} \sum_{j=1}^{N} \sum_{k=1}^{C} -\mathbb{1}_{jk} \log(X_{ijk}) \tag{2}$$

where i, j, k represents student, batch sample and class number indices respectively. $\mathbb{1}_{jk}$ is an one hot label indicator function for j^{th} sample and k^{th} class. Similarly, x_{ijk} is a single output logit from i^{th} student model for j^{th} batch sample and k^{th} class and X_{ijk} is its corresponding softmax output.

Intermediate Loss For every pseudo teacher and compressed student pair, the output feature maps from every pseudo teacher block are compared to the ones at its corresponding compressed student block. In order to facilitate an effective knowledge distillation between these respective map pairs, the compressed student maps are first passed through an adaptation layer which as mentioned earlier is a simple 1 × 1 convolution, mapping the student map channels to the pseudo teacher map channels. A Mean Squared Error loss is used to compare each single element of a given pseudo teacher-student feature map pair. This loss is averaged across the batch. The loss for a single block across all students can be expressed as follows,

$$l_{block}^{intermediate} = \sum_{l=2}^{S} \left(\sum_{m=1}^{N} (|x_m^{PT} - x_m^{l}|)^2 \right) \tag{3}$$

where x_m^l is a feature map of size $H \times W \times C$ corresponding to m^{th} batch sample of the l^{th} student model. $|.|^2$ represents element wise squared L2 norm. $l = 1$ represents the pseudo teacher, also designated as PT in x_m^{PT} which is the corresponding pseudo teacher feature map. The overall intermediate loss can be expressed as:

$$L^{intermediate} = \sum_{b=1}^{B} l_b^{intermediate} \tag{4}$$

This loss is used to update only the compressed student model parameters in order to have the compressed student learn from the pseudo teacher and not the other way round. In our experiments we observed that the mean of adaptation layer weights is on average lower for larger student models. This in turn propagates a smaller model response term in the intermediate loss equation, thus increasing their losses slightly compared to thinner students. This helps balance this loss term across all students.

Knowledge Distillation Loss To facilitate global knowledge transfer from the ensemble teacher to each of the students, a KD loss in form of Kullback-Leibler Divergence Loss is incorporated between the ensemble teacher and student outputs. The outputs of the ensemble teacher and each respective student are softened using a certain temperature T to help students learn efficiently from highly confident ensemble teacher predictions where the wrong class outputs are almost zero. The softened softmax and overall KD loss can be expressed as follows,

$$X_{ijk} = \frac{exp(\frac{x_{ijk}}{T})}{\sum_{k=1}^{C} exp(\frac{x_{ijk}}{T})} \tag{5}$$

$$L^{KD} = \sum_{i=1}^{S}\sum_{j=1}^{N}\sum_{k=1}^{C} X_{jk}^{T} \log\left(\frac{X_{jk}^{T}}{X_{ijk}}\right) \tag{6}$$

where X_{ijk} is the softened softmax output of the i^{th} student for j^{th} batch sample and k^{th} class. Similarly X_{jk}^{T} represents the ensemble teacher softened softmax output for j^{th} batch sample and k^{th} class.

Combined Loss The above presented three losses are combined using a weighted combination, on which the entire framework is trained to reduce this overall loss. This can be mathematically expressed as,

$$L = \alpha L^{Normal} + \beta L^{intermediate} + \gamma L^{KD} \tag{7}$$

The optimal weight value combination which was found out to be $\alpha = 0.7$, $\beta = 0.15$, $\gamma = 0.15$ is discussed in detail in an ablation study presented in later sections.

4 Experiments

Datasets. We incorporate four major academic datasets: (1) CIFAR10 dataset [23] which contains 50,000/10,000 training/test samples drawn from 10 classes. Each class has 6,000 images included in both training and test set sized at 32 × 32 pixels. (2) CIFAR100 dataset [23] which contains 50,000/10,000 training/test samples drawn from 100 classes. Each class has 600 images included in both training and test set sized at 32 × 32 pixels. (3) SVHN dataset [28] which contains 73,257/26,032 training/test samples drawn from 10 classes. Each class represents a digit from 0 to 9. Each image is sized at 32 × 32 pixels. (4) ImageNet dataset [10] is a comprehensive database containing around 1.2 million images, specifically 1,281,184/50,000 training/testing images drawn from 1000 classes.

Experimental Hypothesis. Experiments are conducted in order to: (1) compare every compressed student's test set performance trained using our ensemble framework versus simply training each one of them individually without any

Table 2: Individual Test Set performance comparison for five compressed students trained using our ensemble and using baseline training on CIFAR10 dataset. Reported results are averaged over five individual experimental runs.

Classification Model	**Student Test Accuracy (%)**									
	First		Second		Third		Fourth		Fifth	
	Baseline	Ensemble	Baseline	Ensemble	Baseline	Ensemble	Baseline	Ensemble	Baseline	Ensemble
Resnet20 [17]	91.34	**92.13**	91.12	**92.18**	90.89	**91.78**	90.16	**91.45**	89.67	**91.03**
Resnet32 [17]	92.12	**92.95**	91.94	**92.76**	91.56	**92.45**	91.07	**92.11**	90.47	**91.78**
Resnet44 [17]	92.94	**93.45**	92.67	**93.29**	92.24	**93.11**	91.97	**92.89**	91.23	**92.56**
Resnet110 [17]	93.51	**94.24**	93.25	**94.18**	93.11	**93.98**	92.86	**93.57**	92.27	**93.28**
Densenet-BC (k=12)[20]	94.02	**94.76**	93.78	**94.51**	93.52	**94.29**	93.24	**94.08**	92.85	**93.57**
ResNext50 (32 × $4d$)[41]	95.78	**96.03**	95.56	**95.95**	95.27	**95.84**	95.09	**95.69**	94.97	**95.47**
EfficientNet-B0[40]	97.82	**98.20**	97.58	**98.13**	97.28	**98.01**	97.04	**97.84**	96.73	**97.57**
EfficientNet-B2[40]	98.21	**98.41**	98.13	**98.35**	97.99	**98.23**	97.77	**98.02**	97.41	**97.88**
EfficientNet-B4 [40]	98.56	**98.70**	98.36	**98.59**	98.21	**98.47**	98.04	**98.23**	97.92	**98.14**

Table 3: Individual Test Set performance comparison for five compressed students trained using our ensemble and using baseline training on CIFAR100. Reported results are averaged over five individual experimental runs.

Classification Model	**Student Test Accuracy (%)**									
	First		Second		Third		Fourth		Fifth	
	Baseline	Ensemble	Baseline	Ensemble	Baseline	Ensemble	Baseline	Ensemble	Baseline	Ensemble
Resnet32 [17]	70.21	**70.97**	67.87	**68.24**	64.17	**65.67**	**61.85**	61.17	39.12	**42.17**
Resnet44 [17]	71.12	**71.76**	68.42	**69.12**	65.69	**67.04**	62.31	**62.87**	40.82	**43.11**
Resnet56 [17]	71.59	**72.16**	**68.45**	68.39	65.37	**66.21**	**62.42**	62.21	41.19	**43.27**
Resnet110 [17]	72.64	**72.81**	69.53	**70.14**	67.12	**67.73**	64.58	**65.08**	42.26	**46.76**
Densenet-BC (k=12)[20]	75.79	**75.96**	71.97	**72.39**	**70.23**	70.09	67.13	**68.14**	45.41	**49.12**
ResNeXt50 (32 × $4d$)[41]	72.37	**72.59**	70.19	**70.32**	67.02	**67.81**	65.19	**65.72**	42.82	**45.29**
EfficientNet-B0[40]	87.17	**88.12**	85.78	**86.94**	83.25	**85.14**	80.24	**83.21**	76.35	**78.45**
EfficientNet-B2[40]	89.05	**89.31**	87.34	**88.78**	85.23	**87.58**	82.14	**84.13**	79.34	**81.12**
EfficientNet-B4 [40]	90.26	**90.81**	88.59	**89.78**	86.34	**88.04**	84.32	**86.78**	81.34	**84.10**

knowledge distillation component. These experiments help validate the advantages of using an ensemble teacher and intermediate knowledge transfer for every compressed student compared to using only the traditional individual cross entropy loss based training. (2) Compare the test set accuracy of our ensemble teacher to other notable ensemble knowledge distillation based techniques in literature on all four mentioned datasets to prove our framework's overall superiority and effectiveness, which are presented in Table 4. (3) Compare the time taken for training our five student based ensemble versus the combined time taken for training each of those students individually. This comparison helps substantiate the training time benefits of our hybrid multi-student architecture compared to training each student alone either sequentially or in parallel. These are presented in Figure 2 (b).

Performance Metrics. We compare the test set accuracy (Top-1) of each of our student models within the ensemble, trained using our framework and as an individual baseline model with only the traditional cross entropy loss. For each of our ensemble students, this test set accuracy is computed as an average of the best student test accuracies achieved during each of five conducted runs.

Table 4: Comparison of notable knowledge distillation and ensemble based techniques with our ensemble teacher reported test accuracy performance (Error rate %). The best performing model accuracy is chosen for DML.

Ensemble Technique	**Dataset**							
	CIFAR10		CIFAR100		SVHN		ImageNet	
	ResNet-32	ResNet-110	ResNet-32	ResNet-110	ResNet-32	ResNet-110	Resnet-18	ResNeXt-50
KD-ONE [46]	5.99	5.17	26.61	21.62	**1.83**	1.76	29.45	21.85
DML [43]	–	–	29.03	24.10	–	–	–	–
Snopshot Ensemble [19]	–	5.32	27.12	24.19	–	1.63	–	–
Ours	**5.73**	**4.85**	**26.09**	**21.14**	1.97	**1.61**	**29.34**	**21.17**

Table 5: Individual Test Set performance comparison for five compressed students trained using our ensemble and using baseline training on SVHN. Reported results are averaged over five individual experimental runs.

Classification Model	**Student Test Accuracy (%)**									
	First		Second		Third		Fourth		Fifth	
	Baseline	Ensemble	Baseline	Ensemble	Baseline	Ensemble	Baseline	Ensemble	Baseline	Ensemble
Resnet20 [17]	96.64	**97.10**	95.03	**96.92**	94.45	**95.53**	**92.12**	92.03	89.58	**92.67**
Resnet32 [17]	96.78	**96.92**	95.67	**96.31**	**94.85**	94.61	92.78	**95.03**	90.75	**92.89**
Resnet44 [17]	97.23	**97.46**	**96.38**	96.26	95.35	**96.32**	93.26	**95.76**	91.24	**93.47**
Resnet110 [17]	97.64	**97.87**	96.61	**97.84**	95.83	**96.81**	93.73	**95.90**	91.77	**93.78**
Densenet-BC (k=12)[20]	97.92	**98.03**	97.31	**98.02**	96.12	**97.59**	**94.58**	94.25	92.15	**94.17**
ResNext50 (32 × $4d$)[41]	97.65	**97.88**	**96.84**	96.69	95.72	**96.64**	**94.79**	94.23	91.73	**93.80**
EfficientNet-B0[40]	97.53	**97.72**	97.07	**97.79**	95.52	**96.71**	**94.44**	94.26	91.12	**93.34**
EfficientNet-B2[40]	97.75	**97.92**	**97.76**	97.63	95.87	**96.92**	93.37	**96.24**	**91.42**	91.29
EfficientNet-B4 [40]	98.16	**98.56**	97.79	**98.03**	**96.71**	96.48	93.64	**96.83**	91.75	**94.17**

Experimental Setup. For fair comparison, we keep the training schedule the same for both our ensemble framework and baseline training. Specifically, for ResNet, DenseNet and ResNeXt models SGD is used with Nesterov momentum set to 0.9, following a standard learning rate schedule that drops from 0.1 to 0.01 at 50% training and to 0.001 at 75%. For EfficientNet models RMSProp optimizer is implemented with decay 0.9 and momentum 0.9 and initial learning rate of 0.256 that decays by 0.97 every 3 epochs. The models are trained for 350/450/50/100 epochs each for the CIFAR10/CIFAR100/SVHN/ImageNet datasets respectively.

4.1 Evaluation of our online model compression framework

Results on CIFAR10 and CIFAR100. Tables 2,3 present our experimental results for CIFAR10 and CIFAR100 dataset respectively. Each compressed student's test set performance is on an average 1% better using our ensemble framework as compared to the simple baseline training for both the datasets. Our ensemble teacher also provides the best Test set accuracy when compared to the teacher accuracies of three other ensemble knowledge distillation techniques for ResNet32 and ResNet110 models as presented in Table 4. Our framework provides substantial training time benefits for all models tested with CIFAR10 and CIFAR100 datasets as presented in Fig 2 (b). For fair comparison the ensemble

Table 6: Individual Test Set performance comparison for five compressed students trained using our ensemble and using baseline training on ImageNet (Top-1 accuracy). Reported results are averaged over five individual experimental runs.

Classification Model	**Student Test Accuracy (Top-1 accuracy %)**									
	First		Second		Third		Fourth		Fifth	
	Baseline	Ensemble	Baseline	Ensemble	Baseline	Ensemble	Baseline	Ensemble	Baseline	Ensemble
Resnet18 [17]	69.73	**70.47**	67.27	**67.61**	62.98	**64.88**	59.47	**61.17**	55.23	**58.52**
Resnet34 [17]	73.22	**74.13**	71.95	**73.64**	67.62	**69.32**	63.07	**64.19**	60.76	**61.29**
Resnet50 [17]	76.18	**76.52**	**75.43**	75.32	70.16	**71.93**	66.89	**69.46**	62.24	**66.78**
Resnet101 [17]	77.31	**77.97**	76.27	**76.71**	73.49	**74.04**	69.47	**71.10**	65.79	**68.57**
Densenet-121[20]	74.96	**75.82**	73.94	**74.17**	**68.53**	68.44	66.64	**67.83**	63.42	**66.09**
ResNext50 ($32 \times 4d$)[41]	77.58	**78.19**	76.62	**77.85**	**73.45**	73.37	69.73	**70.89**	65.82	**68.48**

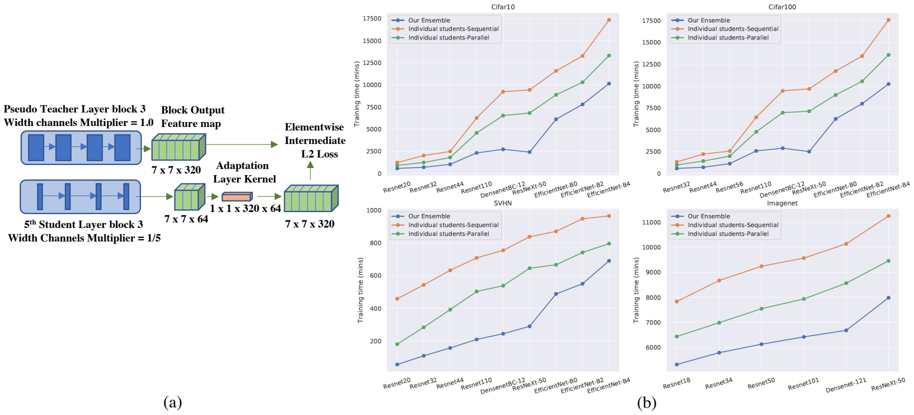


Fig. 2: (a) *Channel Adaptation Logic* for mapping output channels of an *EfficientNet-B0* model based ensemble, depicting Block 3 outputs of the pseudo Teacher and 5th student in the ensemble. (b) Comparison of our ensemble framework training time (Blue) to the combined training time of individual baseline students performed sequentially (Orange) and in parallel (Green). Timings recorded for training carried out on a GPU cluster of Nvidia GTX 1080Ti.

and each of the baseline students are trained for the same number epochs on both datasets. Notably, training a five student ensemble of an EfficientNet-B4 architecture is roughly 7.5K GPU minutes quicker as compared to their combined individual baseline training.

Results on SVHN and ImageNet. Tables 5,6 present our experimental results for SVHN and ImageNet datasets respectively. Again, each compressed student test set performance is on an average 1% better using our ensemble framework as compared to the simple baseline training for both the datasets. Notably for both datasets, the heavily compressed fourth and fifth students perform around 3% on average better than their baseline counterparts. This provide an excellent evidence of our framework's efficient knowledge transfer capabilities for the heavily compressed student cases. Similar to the aforementioned datasets,

Table 7: Ablation Study for α contribution ratio grid search conducted with ResNet110 [17] model on CIFAR100 [23] dataset. Weighted average technique used for calculating effective student model accuracy with higher weight given to more compressed students.

α	**Student Test Accuracy (%)**					**Ensemble Teacher**	**Weighted**
	1	**2**	**3**	**4**	**5**	**Test Accuracy (%)**	**Average**
0.3	69.58	64.47	63.21	55.64	39.57	69.08	161.71
0.4	68.27	65.36	62.35	57.54	40.14	68.81	163.38
0.5	71.32	69.58	67.92	62.69	45.16	72.15	178.16
0.6	72.11	67.82	65.55	60.93	43.19	71.64	172.81
0.7	**71.25**	**69.32**	**67.29**	**62.16**	**47.23**	**72.51**	**179.31**
0.8	71.05	70.21	68.01	62.61	45.10	71.85	178.29
0.9	69.21	66.56	63.12	58.85	45.76	71.86	171.18

our ensemble teacher provides the best Test set accuracy when compared to the ensemble teacher accuracy of three other ensemble knowledge distillation based techniques for ResNet32 and ResNet110 models as presented in Table 4.

5 Ablation Studies

Loss Contribution Ratios. A selective grid search was conducted for the optimal values of loss contribution ratios, specifically α, β, γ referenced in Equation 7. The grid search was conducted with the constraint that the ratio should sum to one which would represent the overall loss factor. The study was carried out using ResNet110 [17] model on the CIFAR100 [23] dataset. Firstly, a grid search was conducted for α which is the normal loss contribution ratio. The other two contribution ratio namely β, γ were kept equal to half the fraction left from subtracting the grid search α from 1. This study is presented in Table 7. All accuracies are averaged over five runs to reduce any weight initialization effect. The value of 0.7 provided the best weighted average accuracy across all the students. Weights of $\frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1$ were assigned to the students, with higher importance given to accuracy achieved by more compressed student.

With the value of α set as 0.7, a grid search was then conducted for the value of β and γ. This study is presented in Table 8. Here also the same weighted average technique was used to find the effective student test accuracy. $\beta = 0.15$ and $\gamma = 0.15$ gave the optimal performance and were thus selected as the final contribution ratios for our framework. These final ratios seem to indicate the major importance of cross entropy loss with $\alpha = 0.7$ in individual student's training and equal importance of intermediate and output distribution knowledge transfer with $\beta = \gamma = 0.15$ for the knowledge distillation process.

Knowledge distillation Temperature. A temperature variable T is used to soften the student and ensemble teacher model logits before computing its respective softmax distribution as referenced in Equation 5. A grid search was conducted for its optimal value, which would facilitate optimum knowledge transfer from the ensemble teacher to each student model. Similar to the previous study,

Table 8: Ablation Study for β, γ contribution ratios grid search conducted with ResNet110 [17] model on CIFAR100 [23] dataset. Weighted average technique used for calculating combined student model test accuracy with higher weight given to more compressed students. α is set at optimal value of 0.7 referred from Table 7.

β	γ	**Student Test Accuracy (%)**					**Ensemble Teacher**	**Weighted**
		1	**2**	**3**	**4**	**5**	**Test Accuracy (%)**	**Average**
0.05	0.25	68.57	66.69	64.34	59.92	47.26	71.97	174.19
0.1	0.2	68.72	68.19	65.94	62.09	44.88	71.73	175.14
0.15	**0.15**	**69.37**	**68.39**	**66.44**	**62.17**	**46.82**	**72.28**	**177.65**
0.2	0.1	66.79	64.92	62.46	57.78	41.34	69.64	164.37
0.25	0.05	67.97	67.22	65.37	61.08	46.69	71.12	175.26

Table 9: Ablation Study for softmax temperature (T) grid search conducted with ResNet110 [17] model on CIFAR100 [23] dataset. Mean accuracy computed using only student test accuracies.

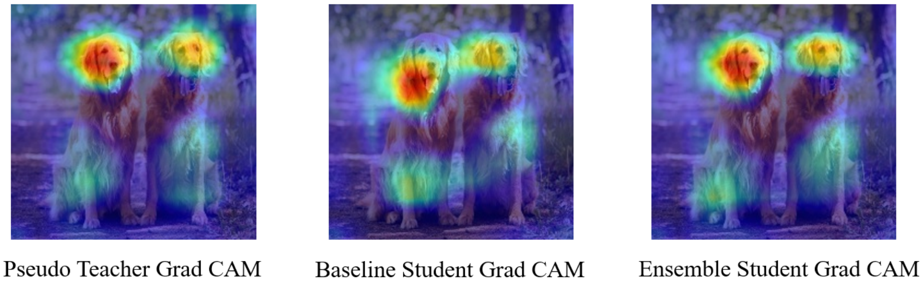
Temperature	**Student Test Accuracy (%)**					**Ensemble Teacher**	**Mean**
T	**1**	**2**	**3**	**4**	**5**	**Test Accuracy (%)**	**Accuracy (%)**
1	70.29	67.16	64.22	62.46	44.75	71.57	61.78
2	**71.89**	**69.56**	**68.43**	**59.48**	**47.26**	**71.02**	**63.32**
3	69.14	68.18	65.33	57.52	46.33	69.36	61.3
4	68.35	66.77	64.36	56.41	46.45	69.37	60.49
5	66.58	66.95	65.67	56.1	42.62	69.27	59.58
6	66.92	66.28	65.33	55.97	43.06	69.34	59.51

we incorporate a ResNet110 [17] model to train on CIFAR100 [23] dataset. This study is presented in Table 9. The resulting optimal value of 2 is used for all of our conducted experiments. The study results provide evidence to the fact that higher temperature values tend to over-soften the output logits leading to sub optimal knowledge transfer and test accuracy gains.

6 Discussion

The performed experiments provide a strong evidence of the efficient compression and generalization capabilities of our framework over individual baseline training for every compressed student model. In most of the experiments the ensemble teacher's test accuracy is much better than any of its ensemble students and their baseline counterparts. This additional test accuracy gain can be attributed to the joint ensemble knowledge learnt by the framework.

The intermediate knowledge transfer from the pseudo teacher onto each one of the compressed students helps guide every student compressed block to reproduce the same transformations its respective higher capacity pseudo teacher block is learning. Enabling the low capacity compressed block to try and imitate the higher capacity pseudo teacher block helps reduce any redundant (sub-optimal) transformations inside the student block that would generally be



Pseudo Teacher Grad CAM

Baseline Student Grad CAM

Ensemble Student Grad CAM

Fig. 3: Gradient Class Activation Mapping (Grad CAM) [32] comparison of a EfficientNet-B4 based ensemble pseudo teacher and one of its compressed students with that of its respective individually trained student. The ensemble student's CAM is more accurate compared to that of baseline student. Also the former follows the pseudo teacher more closely as compared to the latter, which provides evidence of the effective knowledge distillation taking place in our ensemble framework.

present during baseline training. This is substantiated by the fact that the test accuracy gains of heavily compressed students, specifically the fourth and fifth students in the ensemble are substantial over their baseline counterparts. Figure 3 presents a comparison of the gradient based class activation mapping (Grad-CAM) [32] of the last block of an EfficientNet-B4 framework pseudo teacher and one of its compressed students. These are compared to the Grad-CAM of the same compressed student with baseline training. The smaller differences between the Grad-CAMs of pseudo teacher and its ensemble student compared to those between the pseudo teacher and the baseline student provide evidence of how our efficient knowledge distillation helps the student imitate the pseudo teacher and learn better as compared to the baseline student.

7 Conclusion

We present a novel model compression technique using an ensemble knowledge distillation learning procedure without requiring the need of any pretrained weights. The framework manages to provide multiple compressed versions of a given base (pseudo teacher) model simultaneously, providing gains in each of the participating model's test performance and in overall framework's training time compared to each model's individual baseline training. Comprehensive experiments conducted using a variety of current state-of-the-art image classification based models and benchmarked academic datasets provide substantial evidence of the framework's effectiveness. It also provides an account of the highly modular nature of the framework which makes it easier to incorporate any existing classification model into the framework without any major modifications. It manages to provide multiple efficient versions of the same, compressed to varying degree without making any major manual architecture changes on the user's part.

References

1. Adriana, R., Nicolas, B., Ebrahimi, K.S., Antoine, C., Carlo, G., Yoshua, B.: Fitnets: Hints for thin deep nets. In: Proceedings of International Conference on Learning Representations (2015) 2, 4
2. Anil, R., Pereyra, G., Passos, A., Ormandi, R., Dahl, G.E., Hinton, G.E.: Large scale distributed neural network training through online distillation. In: International Conference on Learning Representations (2018) 2, 4
3. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6154–6162 (2018) 1
4. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017) 1
5. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 801–818 (2018) 1
6. Chen, W., Wilson, J., Tyree, S., Weinberger, K., Chen, Y.: Compressing neural networks with the hashing trick. In: International conference on machine learning. pp. 2285–2294 (2015) 1
7. Cheng, Y., Wang, D., Zhou, P., Zhang, T.: A survey of model compression and acceleration for deep neural networks. arXiv preprint arXiv:1710.09282 (2017) 1
8. Cheng, Y., Yu, F.X., Feris, R.S., Kumar, S., Choudhary, A., Chang, S.F.: An exploration of parameter redundancy in deep networks with circulant projections. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2857–2865 (2015) 1
9. Cohen, T., Welling, M.: Group equivariant convolutional networks. In: International conference on machine learning. pp. 2990–2999 (2016) 1
10. 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 IEEE conference on computer vision and pattern recognition. pp. 248–255 (2009) 8
11. Dieleman, S., De Fauw, J., Kavukcuoglu, K.: Exploiting cyclic symmetry in convolutional neural networks. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48. pp. 1889–1898 (2016) 1
12. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Keypoint triplets for object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6569–6578 (2019) 1
13. Furlanello, T., Lipton, Z.C., Tschannen, M., Itti, L., Anandkumar, A.: Born again neural networks. arXiv preprint arXiv:1805.04770 (2018) 2
14. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. In: Advances in neural information processing systems. pp. 1135–1143 (2015) 1
15. Hanson, S.J., Pratt, L.Y.: Comparing biases for minimal network construction with back-propagation. In: Advances in neural information processing systems. pp. 177–185 (1989) 1
16. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017) 1
17. 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. pp. 770–778 (2016) 1, 4, 6, 9, 10, 11, 12, 13

18. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015) 2, 3
19. Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J.E., Weinberger, K.Q.: Snapshot ensembles: Train 1, get m for free. arXiv preprint arXiv:1704.00109 (2017) 10
20. 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. pp. 4700–4708 (2017) 1, 6, 9, 10, 11
21. Kim, Y., Rush, A.M.: Sequence-level knowledge distillation. arXiv preprint arXiv:1606.07947 (2016) 2
22. Koratana, A., Kang, D., Bailis, P., Zaharia, M.: Lit: Learned intermediate representation training for model compression. In: International Conference on Machine Learning. pp. 3509–3518 (2019) 2, 5
23. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009), https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf 6, 8, 12, 13
24. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012) 1
25. Li, H., Ouyang, W., Wang, X.: Multi-bias non-linear activation in deep neural networks. In: International conference on machine learning. pp. 221–229 (2016) 1
26. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016) 1
27. Liu, Y., Wang, Y., Wang, S., Liang, T., Zhao, Q., Tang, Z., Ling, H.: Cbnet: A novel composite backbone network architecture for object detection. arXiv preprint arXiv:1909.03625 (2019) 1
28. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning (2011) 8
29. Rakhuba, M., Oseledets, I.V.: Fast multidimensional convolution in low-rank tensor formats via cross approximation. SIAM Journal on Scientific Computing **37**(2), A565–A582 (2015) 1
30. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016) 1
31. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015) 1
32. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017) 14
33. Shang, W., Sohn, K., Almeida, D., Lee, H.: Understanding and improving convolutional neural networks via concatenated rectified linear units. In: international conference on machine learning. pp. 2217–2225 (2016) 1
34. Shen, Z., He, Z., Xue, X.: Meal: Multi-model ensemble via adversarial learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 4886–4893 (2019) 2, 4
35. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) 1

36. Sindhwani, V., Sainath, T., Kumar, S.: Structured transforms for small-footprint deep learning. In: Advances in Neural Information Processing Systems. pp. 3088–3096 (2015) 1
37. Srinivas, S., Babu, R.V.: Data-free parameter pruning for deep neural networks. arXiv preprint arXiv:1507.06149 (2015) 1
38. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015) 1, 4
39. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016) 4
40. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946 (2019) 1, 6, 9, 10
41. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1492–1500 (2017) 4, 6, 9, 10, 11
42. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4133–4141 (2017) 4
43. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4320–4328 (2018) 4, 10
44. Zhang, Z., Zhang, X., Peng, C., Xue, X., Sun, J.: Exfuse: Enhancing feature fusion for semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 269–284 (2018) 1
45. Zhong, Z., Lin, Z.Q., Bidart, R., Hu, X., Daya, I.B., Li, J., Wong, A.: Squeeze-and-attention networks for semantic segmentation. arXiv preprint arXiv:1909.03402 (2019) 1
46. Zhu, X., Gong, S., et al.: Knowledge distillation by on-the-fly native ensemble. In: Advances in neural information processing systems. pp. 7517–7527 (2018) 2, 4, 10