Knowledge Distillation using Teaching Assistant

Ahmer Jamil

Abstract

Deep neural networks are powerful models and achieve appealing results on many tasks. However, they are too large to be deployed on edge devices like smartphones.

To solve this, there have been many efforts to compress DNNs using many methods. One such method is Knowledge Distillation where a larger network (Teacher) is used to train a smaller student network.

Moreover, variations of methods of learning have also been proposed such as learning the dense feature representation.

Research has also been conducted on the effect of gap between the Teacher model and the student model and intermediary models called Teacher Assistants have been proposed to improve learning.

In this paper we will discuss previous methodologies proposed and see the effect of using Teaching Assistants on dense feature learning.

We use CIFAR-10 for our experiments and analysis.

Introduction

A lot of work and research has been done to compress neural networks and many techniques have been used with multiple variations in each technique.

Some popular neural network compression techniques include pruning, which includes unstructured pruning and structured pruning, quantization, knowledge distillation etc.

Pruning works by removing connections between neurons or entire neurons, channels, or filters from a trained network, which is done by zeroing out values in its weights matrix or removing groups of weights entirely (Peterson).

In contrast, knowledge distillation works by training a smaller network from a larger teacher network.

Variations of the original proposed methods by (Hinton et al.) and others include techniques of using Teaching Assistants proposed by (Mirzadeh et al.) and the teacher-class network consisting of a single teacher and multiple students proposed by, (Malik et al.).

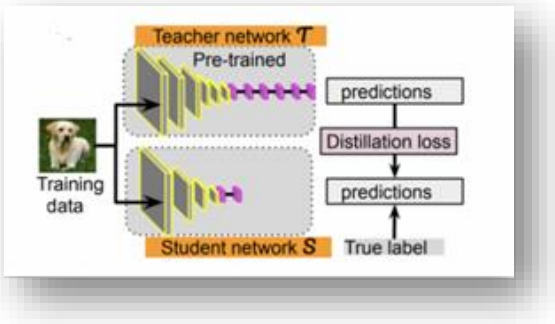
In this paper we aim to use multiple hierarchal student networks that learn the dense feature representation of the students in the layer above them. Hence combining both strategies proposed by (Mirzadeh et al.) and (Malik et al.) to see the effect of introducing Teaching Assistants In (Malik et al.)'s approach.

Related works

In 2006, Cornell researchers proposed the idea of transferring the knowledge from a large, trained model (or ensemble of models) to a smaller model for deployment by training it to mimic the larger model's output (Peterson).

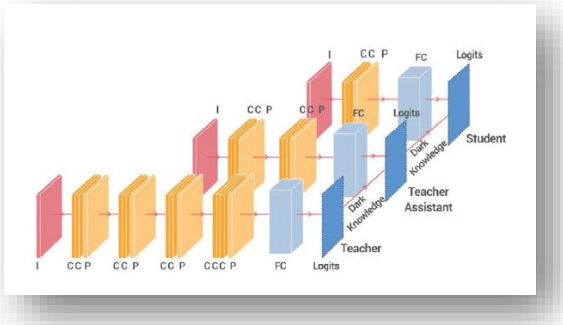
In 2015, Hinton (Hinton et al.) proposed a model compression technique to distill the knowledge in a cumbersome teacher model into a lightweight student model, hoping that it will enhance the student generalization ability. In this paper the class predictions between teacher and student models were aligned, with a newly introduced hyper-parameter—temperature in the softmax

activation to control the softness of predicted distributions (Chen et al.)



Hence, in Hinton's model, (Hinton et al.) the small "student" network learns to mimic the large "teacher" network by minimizing a loss function in which the target is based on the distribution of class probabilities outputted by the teacher's softmax function (Peterson).

In 2019, (Mirzadeh et al.) proposed that knowledge distillation is not always effective, especially when the gap (in size) between teacher and student is large. They proposed a new distillation framework called Teacher Assistant Knowledge Distillation (TAKD), which introduces intermediate models as teacher assistants (TAs) between the teacher and the student to fill in their gap. (Mirzadeh et al.)



In this model the Teacher Assistants are distilled from the teacher, and the student is then only distilled from the Teacher Assistants.

The paper highlighted a few important points:

- Size (capacity) gap between teacher and student is important.
- Proposed model improves the accuracy of student network in the case of extreme compression.
- The framework can be extended to include a chain of multiple Tas from teacher to student to further improve the knowledge

Further work by (Son et al.) revealed the problem of error avalanche which states that if an error occurs during a specific TA model learning, this error will be passed forward to all students who learn from the TA. To solve this problem (Son et al.) proposed the densely guided knowledge distillation (DGKD) model which uses multiple TAs for efficient learning.

The paper argued that "it is necessary that the knowledge connection from the teachers is irregularly altered during the training" (Son et al.) and to cater to this the paper proposes to "randomly select the combination of distilled knowledge from the complex teacher and teacher assistant models. This process acts as a regularization function, and it alleviates the overfitting problem" (Son et al.)

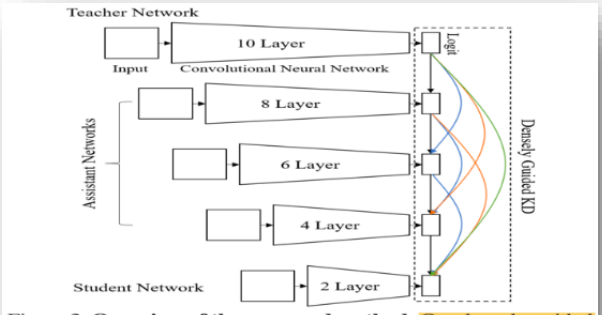


Figure 2. **Overview of the proposed method.** Our densely guided knowledge distillation using multiple teacher assistant networks is able to train a small-sized student network from a large-sized teacher network efficiently through the multiple teacher assistant networks.

In 2021, (Malik et al.) proposed a method called a teacher-class network consisting of a single teacher and multiple student networks where instead of transferring knowledge to one student only, it transfers a chunk of knowledge about the entire solution to each student.

The students are not trained for problem-specific logits, rather they are trained to mimic knowledge (dense representation) learned by the teacher network which enables them to solve other problems as well (Malik et al.).

This approach outperforms the state-of-the-art single student approach in terms of accuracy as well as computational cost and it achieves an accuracy equivalent to the teacher network while having 10 to 30 times fewer parameters (Malik et al.).

Two key differences between the proposed method and Teacher Student method used previously are that instead of just one student, the proposed architecture employs multiple students to learn mutually exclusive chunks of the knowledge and instead of training student on the soft labels (probabilities produced by the SoftMax) of the teacher, the architecture tries to learn dense feature representations, thus making the solution problem independent (Malik et al.).

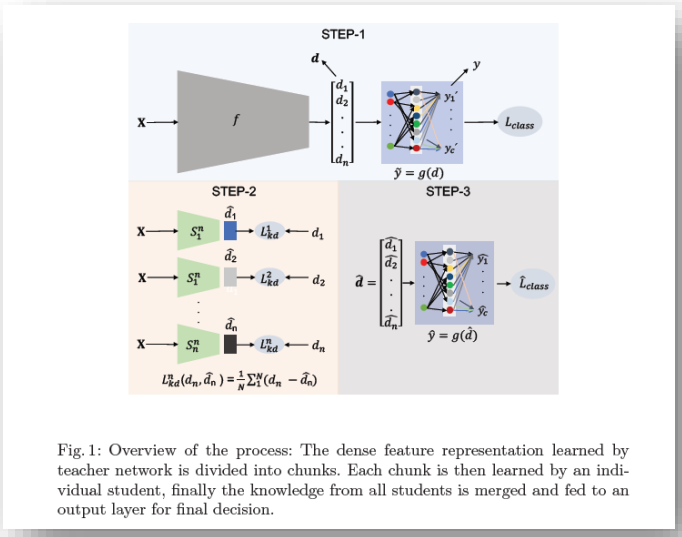


Fig. 1: Overview of the process: The dense feature representation learned by teacher network is divided into chunks. Each chunk is then learned by an individual student, finally the knowledge from all students is merged and fed to an output layer for final decision.

In this paper we use multiple hierarchal student networks that learn the dense feature representation of the students in the layer above them. Hence combining both strategies proposed by (Mirzadeh et al.) and (Malik et al.) to see the effect of introducing Teaching Assistants In (Malik et al.)'s approach where each Teaching Assistant is an ensemble of students that learns the Dense Feature representation of the Ensemble above it in hierarchy, up until the original teacher.

We compare the results of a hierarchal network with that of learning directly from the Teacher versus one single teaching assistant.

Methodology

In the first step we Train a Teacher, a Single Student, 2 Students, 4 Students, and 8 Students all in a hierarchal manner.

The network and number of parameters for each model is listed below and the number of layers and parameters are kept constant throughout.

Teacher - 14,728,266 Parameters

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 64, 32, 32]	1,792
BatchNorm2d-2	[-1, 64, 32, 32]	128
ReLU-3	[-1, 64, 32, 32]	0
Conv2d-4	[-1, 64, 32, 32]	36,928
BatchNorm2d-5	[-1, 64, 32, 32]	128
ReLU-6	[-1, 64, 32, 32]	0
MaxPool2d-7	[-1, 64, 16, 16]	0
Conv2d-8	[-1, 128, 16, 16]	73,856
BatchNorm2d-9	[-1, 128, 16, 16]	256
ReLU-10	[-1, 128, 16, 16]	0
Conv2d-11	[-1, 128, 16, 16]	147,584
BatchNorm2d-12	[-1, 128, 16, 16]	256
ReLU-13	[-1, 128, 16, 16]	0
MaxPool2d-14	[-1, 128, 8, 8]	0
Conv2d-15	[-1, 256, 8, 8]	295,168
BatchNorm2d-16	[-1, 256, 8, 8]	512
ReLU-17	[-1, 256, 8, 8]	0
Conv2d-18	[-1, 256, 8, 8]	590,080
BatchNorm2d-19	[-1, 256, 8, 8]	512
ReLU-20	[-1, 256, 8, 8]	0
Conv2d-21	[-1, 256, 8, 8]	590,080
BatchNorm2d-22	[-1, 256, 8, 8]	512
ReLU-23	[-1, 256, 8, 8]	0
MaxPool2d-24	[-1, 256, 4, 4]	0
Conv2d-25	[-1, 512, 4, 4]	1,180,160
BatchNorm2d-26	[-1, 512, 4, 4]	1,024
ReLU-27	[-1, 512, 4, 4]	0
Conv2d-28	[-1, 512, 4, 4]	2,359,808
BatchNorm2d-29	[-1, 512, 4, 4]	1,024
ReLU-30	[-1, 512, 4, 4]	0
Conv2d-31	[-1, 512, 4, 4]	2,359,808
BatchNorm2d-32	[-1, 512, 4, 4]	1,024
ReLU-33	[-1, 512, 4, 4]	0
MaxPool2d-34	[-1, 512, 2, 2]	0
Conv2d-35	[-1, 512, 2, 2]	2,359,808
BatchNorm2d-36	[-1, 512, 2, 2]	1,024
ReLU-37	[-1, 512, 2, 2]	0
Conv2d-38	[-1, 512, 2, 2]	2,359,808
BatchNorm2d-39	[-1, 512, 2, 2]	1,024
ReLU-40	[-1, 512, 2, 2]	0
Conv2d-41	[-1, 512, 2, 2]	2,359,808
BatchNorm2d-42	[-1, 512, 2, 2]	1,024
ReLU-43	[-1, 512, 2, 2]	0
MaxPool2d-44	[-1, 512, 1, 1]	0
AvgPool2d-45	[-1, 512, 1, 1]	0
Linear-46	[-1, 10]	5,130

Student-(S1) - 2,618,016 parameters

Student learns the entire dense feature of the teacher.

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 32, 32]	896
BatchNorm2d-2	[-1, 32, 32, 32]	64
ReLU-3	[-1, 32, 32, 32]	0
Conv2d-4	[-1, 32, 32, 32]	9,248
BatchNorm2d-5	[-1, 32, 32, 32]	64
ReLU-6	[-1, 32, 32, 32]	0
MaxPool2d-7	[-1, 32, 16, 16]	0
Conv2d-8	[-1, 64, 16, 16]	18,496
BatchNorm2d-9	[-1, 64, 16, 16]	128
ReLU-10	[-1, 64, 16, 16]	0
Conv2d-11	[-1, 64, 16, 16]	36,928
BatchNorm2d-12	[-1, 64, 16, 16]	128
ReLU-13	[-1, 64, 16, 16]	0
MaxPool2d-14	[-1, 64, 8, 8]	0
Conv2d-15	[-1, 128, 8, 8]	73,856
BatchNorm2d-16	[-1, 128, 8, 8]	256
ReLU-17	[-1, 128, 8, 8]	0
Conv2d-18	[-1, 128, 8, 8]	147,584
BatchNorm2d-19	[-1, 128, 8, 8]	256
ReLU-20	[-1, 128, 8, 8]	0
MaxPool2d-21	[-1, 128, 4, 4]	0
Conv2d-22	[-1, 256, 4, 4]	295,168
BatchNorm2d-23	[-1, 256, 4, 4]	512
ReLU-24	[-1, 256, 4, 4]	0
Conv2d-25	[-1, 256, 4, 4]	590,080
BatchNorm2d-26	[-1, 256, 4, 4]	512
ReLU-27	[-1, 256, 4, 4]	0
MaxPool2d-28	[-1, 256, 2, 2]	0
Conv2d-29	[-1, 512, 2, 2]	1,180,160
BatchNorm2d-30	[-1, 512, 2, 2]	1,024
ReLU-31	[-1, 512, 2, 2]	0
MaxPool2d-32	[-1, 512, 1, 1]	0
AvgPool2d-33	[-1, 512, 1, 1]	0
Linear-34	[-1, 512]	262,656

2 Students- (S01-S2) - 1,239,968 parameters each. Each student learns half of the dense features of S1.

Ensemble of both students –

2,485,066 parameters

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 32, 32]	896
BatchNorm2d-2	[-1, 32, 32, 32]	64
ReLU-3	[-1, 32, 32, 32]	0
Conv2d-4	[-1, 32, 32, 32]	9,248
BatchNorm2d-5	[-1, 32, 32, 32]	64
ReLU-6	[-1, 32, 32, 32]	0
MaxPool2d-7	[-1, 32, 16, 16]	0
Conv2d-8	[-1, 64, 16, 16]	18,496
BatchNorm2d-9	[-1, 64, 16, 16]	128
ReLU-10	[-1, 64, 16, 16]	0
Conv2d-11	[-1, 64, 16, 16]	36,928
BatchNorm2d-12	[-1, 64, 16, 16]	128
ReLU-13	[-1, 64, 16, 16]	0
MaxPool2d-14	[-1, 64, 8, 8]	0
Conv2d-15	[-1, 128, 8, 8]	73,856
BatchNorm2d-16	[-1, 128, 8, 8]	256
ReLU-17	[-1, 128, 8, 8]	0
Conv2d-18	[-1, 128, 8, 8]	147,584
BatchNorm2d-19	[-1, 128, 8, 8]	256
ReLU-20	[-1, 128, 8, 8]	0
MaxPool2d-21	[-1, 128, 4, 4]	0
Conv2d-22	[-1, 256, 4, 4]	295,168
BatchNorm2d-23	[-1, 256, 4, 4]	512
ReLU-24	[-1, 256, 4, 4]	0
Conv2d-25	[-1, 256, 4, 4]	590,080
BatchNorm2d-26	[-1, 256, 4, 4]	512
ReLU-27	[-1, 256, 4, 4]	0
MaxPool2d-28	[-1, 256, 2, 2]	0
MaxPool2d-29	[-1, 256, 1, 1]	0
AvgPool2d-30	[-1, 256, 1, 1]	0
Linear-31	[-1, 256]	65,792

4 Students – (S11, S22, S33, S44)

600,096 parameters each.

S11 and S22 learn from S01 (each learns half of S01's dense features) whereas S33 and S44 learn from S2.

Ensembled network - 2,405,514 parameters.

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 32, 32]	896
BatchNorm2d-2	[-1, 32, 32, 32]	64
ReLU-3	[-1, 32, 32, 32]	0
MaxPool2d-4	[-1, 32, 16, 16]	0
Conv2d-5	[-1, 32, 16, 16]	9,248
BatchNorm2d-6	[-1, 32, 16, 16]	64
ReLU-7	[-1, 32, 16, 16]	0
MaxPool2d-8	[-1, 32, 8, 8]	0
Conv2d-9	[-1, 64, 8, 8]	18,496
BatchNorm2d-10	[-1, 64, 8, 8]	128
ReLU-11	[-1, 64, 8, 8]	0
Conv2d-12	[-1, 64, 8, 8]	36,928
BatchNorm2d-13	[-1, 64, 8, 8]	128
ReLU-14	[-1, 64, 8, 8]	0
MaxPool2d-15	[-1, 64, 4, 4]	0
Conv2d-16	[-1, 128, 4, 4]	73,856
BatchNorm2d-17	[-1, 128, 4, 4]	256
ReLU-18	[-1, 128, 4, 4]	0
Conv2d-19	[-1, 128, 4, 4]	147,584
BatchNorm2d-20	[-1, 128, 4, 4]	256
ReLU-21	[-1, 128, 4, 4]	0
MaxPool2d-22	[-1, 128, 2, 2]	0
Conv2d-23	[-1, 128, 2, 2]	147,584
BatchNorm2d-24	[-1, 128, 2, 2]	256
ReLU-25	[-1, 128, 2, 2]	0
Conv2d-26	[-1, 128, 2, 2]	147,584
BatchNorm2d-27	[-1, 128, 2, 2]	256
ReLU-28	[-1, 128, 2, 2]	0
MaxPool2d-29	[-1, 128, 1, 1]	0
AvgPool2d-30	[-1, 128, 1, 1]	0
Linear-31	[-1, 128]	16,512

8 Students-(SS11, SS22, SS33, SS44, SS55, SS66, SS77, SS88)

242,112 parameters each.

Ensemble - 1,942,026 parameters.

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 32, 32]	896
BatchNorm2d-2	[-1, 32, 32, 32]	64
ReLU-3	[-1, 32, 32, 32]	0
Conv2d-4	[-1, 32, 32, 32]	9,248
BatchNorm2d-5	[-1, 32, 32, 32]	64
ReLU-6	[-1, 32, 32, 32]	0
MaxPool2d-7	[-1, 32, 16, 16]	0
Conv2d-8	[-1, 32, 16, 16]	9,248
BatchNorm2d-9	[-1, 32, 16, 16]	64
ReLU-10	[-1, 32, 16, 16]	0
Conv2d-11	[-1, 32, 16, 16]	9,248
BatchNorm2d-12	[-1, 32, 16, 16]	64
ReLU-13	[-1, 32, 16, 16]	0
MaxPool2d-14	[-1, 32, 8, 8]	0
Conv2d-15	[-1, 32, 8, 8]	9,248
BatchNorm2d-16	[-1, 32, 8, 8]	64
ReLU-17	[-1, 32, 8, 8]	0
Conv2d-18	[-1, 64, 8, 8]	18,496
BatchNorm2d-19	[-1, 64, 8, 8]	128
ReLU-20	[-1, 64, 8, 8]	0
MaxPool2d-21	[-1, 64, 4, 4]	0
Conv2d-22	[-1, 64, 4, 4]	36,928
BatchNorm2d-23	[-1, 64, 4, 4]	128
ReLU-24	[-1, 64, 4, 4]	0
Conv2d-25	[-1, 64, 4, 4]	36,928
BatchNorm2d-26	[-1, 64, 4, 4]	128
ReLU-27	[-1, 64, 4, 4]	0
MaxPool2d-28	[-1, 64, 2, 2]	0
Conv2d-29	[-1, 64, 2, 2]	36,928
BatchNorm2d-30	[-1, 64, 2, 2]	128
ReLU-31	[-1, 64, 2, 2]	0
Conv2d-32	[-1, 64, 2, 2]	36,928
BatchNorm2d-33	[-1, 64, 2, 2]	128
ReLU-34	[-1, 64, 2, 2]	0
Conv2d-35	[-1, 64, 2, 2]	36,928
BatchNorm2d-36	[-1, 64, 2, 2]	128
ReLU-37	[-1, 64, 2, 2]	0
MaxPool2d-38	[-1, 64, 1, 1]	0
AvgPool2d-39	[-1, 64, 1, 1]	0

Then we train the Single Student, 2 Students, 4 Students, and 8 Students all directly from the teacher.

In the third experiment we train Single student from the Teacher and all others from the student that acts as a single TA.

All models are trained on 60 epochs, fine tuning of ensembled networks is done at 10 epochs and Teacher is trained for 10 Epochs.

Teacher and single student is trained once and then frozen.

Results

1) Hierarchal training

	Student 1	2 Students		4 Students			
	S1	S01	S2	S11	S22	S33	S44
Training Loss	0.079894418	0.023744282	0.042217705	0.030895538	0.043216041	0.043713677	0.044545129

	8 Students							
	SS11	SS22	SS33	SS44	SS55	SS66	SS77	SS88
Training Loss	0.043700559	0.062312505	0.065712594	0.056858827	0.066922603	0.064318213	0.070024853	0.063700383

	Teacher	Student 1	2 Students Ensemble	4 Students Ensemble	8 Students Ensemble
Accuracy	82.889	85.60494	85.55555556	84.09876543	83.08641975

2) All models trained directly on Teacher.

	Student 1	2 Students		4 Students			
	S1	S01	S2	S11	S22	S33	S44
Training Loss	0.079894418	0.025378459	0.04553935	0.02932269	0.045931648	0.047523758	0.050684244

	8 Students							
	SS11	SS22	SS33	SS44	SS55	SS66	SS77	SS88
Training Loss	0.09173475	0.110174057	0.119646394	0.116199981	0.134839853	0.129923464	0.137830899	0.145678369

	Teacher	Student 1	2 Students Ensemble	4 Students Ensemble	8 Students Ensemble
Accuracy	82.8889	85.604938	86.56790123	85.12345679	84.92592593

3) All models trained on Student.

	Student 1	2 Students		4 Students			
	S1	S01	S2	S11	S22	S33	S44
Training Loss	0.07989	0.0686612	0.128356292	0.082879596	0.118804942	0.1174384	0.125142182

	8 Students							
	SS11	SS22	SS33	SS44	SS55	SS66	SS77	SS88
Training Loss	0.032402387	0.046275529	0.043223184	0.058200278	0.06028414	0.050087708	0.062227251	0.0523487

	Teacher	Student 1	2 Students Ensemble	4 Students Ensemble	8 Students Ensemble
Accuracy	82.88888889	85.60493827	85.32098765	83.71604938	83.9382716

Results

Comparing the results from our experiments, we can see some patterns in the accuracies.

In hierarchal training, the accuracies of the ensemble models decrease as we keep increasing the number of Teaching Assistants.

The best accuracies are obtained when all the ensembles are directly trained on the teacher.

Comparing the hierarchal training with training 2 student, 4 student and 8 student models on the single student (TA) resulted in decrease in accuracies for the 2 students as well as 4 student models whereas the 8-student model trained through TA resulted in better accuracy of 83.938 in comparison of hierarchal training accuracy of 83.08.

However, both single TA as well as hierarchal TAs resulted in lowered accuracies as compared to direct training, hence indicating better learning of students directly on the Teacher's Dense Vectors.

Future Work

The findings of this paper are limited to CIFAR-10 and can and should be further tested on other data sets.

Moreover, other approaches such as the DGKD approach proposed by (Son et al.) and the Residual error-based knowledge distillation approach proposed by (Gao et al.) can further be used to test the effect of using teaching assistants on Teacher Class network proposed by (Malik et al.).

References

Chen, Defang, et al. "Exploring the Connection between Knowledge ... - Arxiv.org." *Exploring the Connection between Knowledge Distillation and Logits Matching*, https://arxiv.org/pdf/2109.06458.

Gao, Mengya, et al. "Residual Knowledge Distillation." *ArXiv.org*, 21 Feb. 2020, https://arxiv.org/abs/2002.09168.

Hinton, Geoffrey, et al. "Distilling the Knowledge in a Neural Network." *ArXiv.org*, 9 Mar. 2015, https://arxiv.org/abs/1503.02531.

Malik, Shaiq Munir, et al. "Teacher-Class Network: A Neural Network Compression Mechanism." *ArXiv.org*, 29 Oct. 2021, https://arxiv.org/abs/2004.03281.

Mirzadeh, Seyed-Iman, et al. "Improved Knowledge Distillation via Teacher Assistant." *ArXiv.org*, 17 Dec. 2019, https://arxiv.org/abs/1902.03393.

Peterson, Hannah. "An Overview of Model Compression Techniques for Deep Learning in Space." *Medium*, GSI Technology, 10 Sept. 2020, https://medium.com/gsi-technology/an-overview-of-model-compression-techniques-for-deep-learning-in-space-3fd8d4ce84e5.

Son, Wonchul, et al. "Densely Guided Knowledge Distillation Using Multiple Teacher Assistants." *ArXiv.org*, 9 Aug. 2021, https://arxiv.org/abs/2009.08825.