Experiments with Knowledge Distillation for building lightweight Deep Learning Models

Vaibhav Thakkar
170778
vaithak@iitk.ac.in

Arpit Kumar Jhunjhunwala
170149
arpitjjw@iitk.ac.in

Disclaimer
The work carried out in this project has not been re-used from any another course project at IITK or elsewhere, or any other project done elsewhere (e.g., an internship).
Further we have acknowledged all the sources used and cited them in the reference section.

1 Introduction and Motivation

Deep Learning has been successful in a variety of areas and applications, some of the classic ones being Robots, Chatbots and Face Recognition; some examples of surprising applications being Google's DeepMind AlphaFold2 for Protein Folding [1], Lip Reading, Agriculture [8] and many more. The great success of deep learning is mainly due to its capability to encode large-scale data and to maneuver billions of model parameters. However, it is a challenge to deploy these cumbersome deep models on devices with limited resources, e.g., mobile phones and embedded devices, not only because of the high computational complexity but also the large storage requirements.

The main aim of knowledge distillation is to give the power to deploy lightweight models on such devices without sacrificing much on accuracy and performance. Student models taught using this technique performed better than the student models trained using only the training data.

We will explore more insights into this technique in general using empirical analysis on several datasets and models combination.

2 Problem Description

A standard classification model is trained using the hard labelelled data, whereas a student model trained using knowledge distillation follows the given procedure as proposed by Hinton et. al [7]:

During training along with the labels of data, the student also uses the probabilities outputted by the teacher model, the benefit of using these probabilities is that it contains more information than just hard labels, as it also tells the similarity and differences between classes (this is called as the dark knowledge).

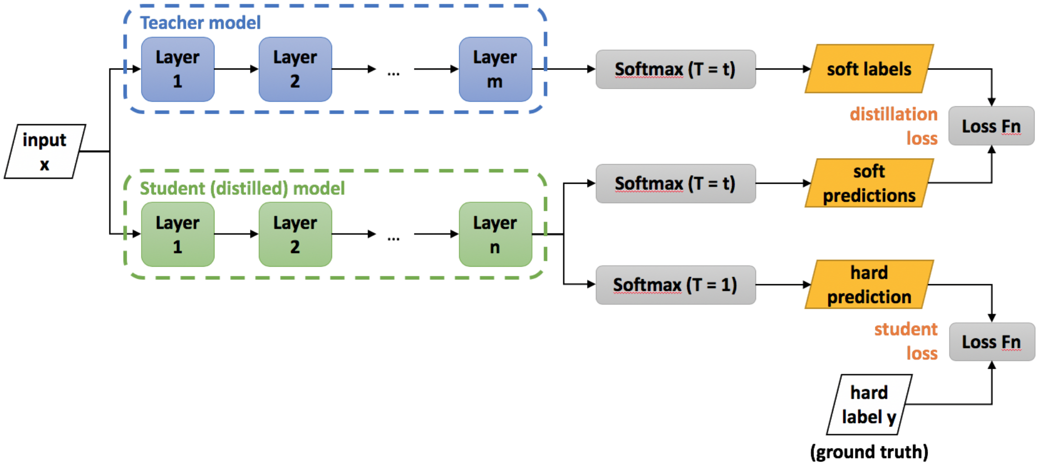
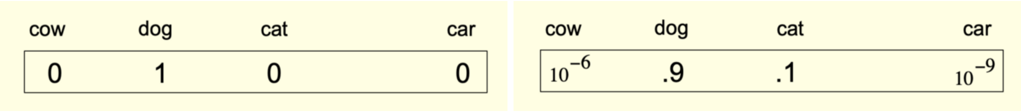


Figure 1: Overview of logit based knowledge distillation. Source: **medium.com**

For transferring the dark knowledge from teacher to student, the logits from the pre-softmax layer is used to output the soft probabilities at a temperature T (≥ 1), using the following equation:

$$\text{softmax}\left(\frac{a}{T}\right) = \left[\frac{\exp(a_i/T)}{\sum_j \exp(a_j/T)}\right]_i \tag{1}$$

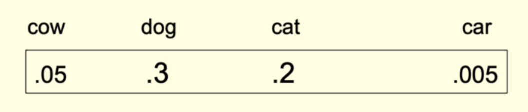
An example of why soft predictions at raised temperature are useful:



cow	dog	cat	car
0	1	0	0

cow	dog	cat	car
10^{-6}	.9	.1	10^{-9}

Figure 2: a) Hard Labels b) Soft Predictions at T = 1



cow	dog	cat	car
.05	.3	.2	.005

Figure 3: Soft Prediction at raised temperature T

Therefore the final loss function is:

$$\mathcal{L}_{student} = (1-\alpha)\mathcal{L}_{sl} + \alpha\mathcal{L}_{KD} \tag{2}$$

Where a_s, a_t are logits from layer of student and teacher model respectively, and:

$$\begin{aligned} \mathcal{L}_{sl} &= \mathcal{H}\left(\text{softmax}\left(a_s\right), y\right) \\ \mathcal{L}_{KD} &= T^2 \cdot KL\left(y_s, y_t\right) \\ y_s &= \text{softmax}\left(\frac{a_s}{T}\right) \\ y_t &= \text{softmax}\left(\frac{a_t}{T}\right) \end{aligned} \tag{3}$$

So, in a way the student is also trying to mimic the output of the teacher.

3 Literature review

- The idea of Knowledge distillation was first proposed by Caruana et al. [3] in 2006 for the purpose of model compression. In 2015, one of the most influential KD paper was published by Hinton et al. [7] which inspired many other variants. The class probabilities produced by the more complex Teacher model are used as ‘soft targets’ in addition to the available class labels to train the Student model. Our problem formulation in the above section is mainly referenced from this paper.

- A lot of recent works use the knowledge from intermediate layers of the teacher to improve the performance of the student model. In the Fitnets [12] architecture, the student model is thinner but deeper than the teacher. The paper proposes a ’Hint-based training’ method in which a layer (’guided’) of the student learns the output of an intermediate layer (’hint’) of the teacher using a fully-connected or convolutional regressor in a 2-stage training process. In this method the student’s guided layer tries to directly learn the activations of the hint layer using the MSE loss.
- Some recent works have proposed indirect ways of learning the intermediate feature maps. These methods perform distillation on derived maps. Zagoruyko and Komodakis [15] proposed the use of pair of attention maps derived from the intermediate layers of the student and teacher models. These pair of maps are computed across several layers and a transfer loss is minimized in addition to the standard cross-entropy loss. Yim et al. [13] proposed the use of Flow of Solution Procedure (FSP) matrix calculated from the feature maps to transfer knowledge between the intermediate layers.
- While most of the KD methods uses a Teacher model, Yuan et al. [14] proposed a Teacher-free framework which uses a pre-trained student model as the Teacher. A second proposed method uses a manually designed virtual teacher with 100% accurate soft predictions on training data. Yuan et al. provided a theoretical justification as to why these two methods work by examining KD from the perspective of Label Smoothing Regularization (LSR). The overall loss function when applying LSR to the student model is given by:

  $$\mathcal{L}_{Student} = (1-\alpha)\mathcal{H}_{Student} + \alpha\mathcal{L}_{KL}(u, p_{student}) \tag{4}$$

  where $\mathcal{H}_{Student}$ is the standard cross entropy loss of the student and $\mathcal{L}_{KL}(u, p)$ is the KL divergence loss between a uniform distribution u and the predicted class probability distribution p.
  Eq.(2) and Eq.(4) have similar form which which suggests that Knowledge-Distillation is a learned Label Smoothing Regularization.
- Mirzadeh et al [11] introduced the usage of Teacher assistants (TA) for bridging the complexity gap between student and teacher model. The idea was that the Teacher model first teaches the TA model, and then the TA model teaches the student model.
- Knowledge Distillation can also be used to build compact and efficient object detection networks. One such framework was proposed by Guobin Chen et al. [5] in 2017.

4 Our Experiments

4.1 Datasets

- **MNIST Dataset of Handwritten Digits**: [10] This dataset consists of a training set of 60,000 examples, and a test set of 10,000 examples. All the images are in grayscale and are of size 28x28 with a label indicating the digit from 0-9. Also, the digits have been size-normalized and centered.
- **CIFAR 10 Dataset**: [9] This dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class.

4.2 Experiment 1: Basic Validation

Our first experiment was basically to validate the transfer of the dark knowledge.

- **Dataset:** MNIST Dataset
- **Teacher Model:** 1 Convolutional Layer + 4 FC layers

- **Student Model:** 1 Convolutional Layer + 2 FC layers

Table 1: Comparison of models for MNIST Dataset

Model	Size	No. of Parameters
Teacher Model	4.01 MB	1,422,378
Student Model	343 KB	693,962

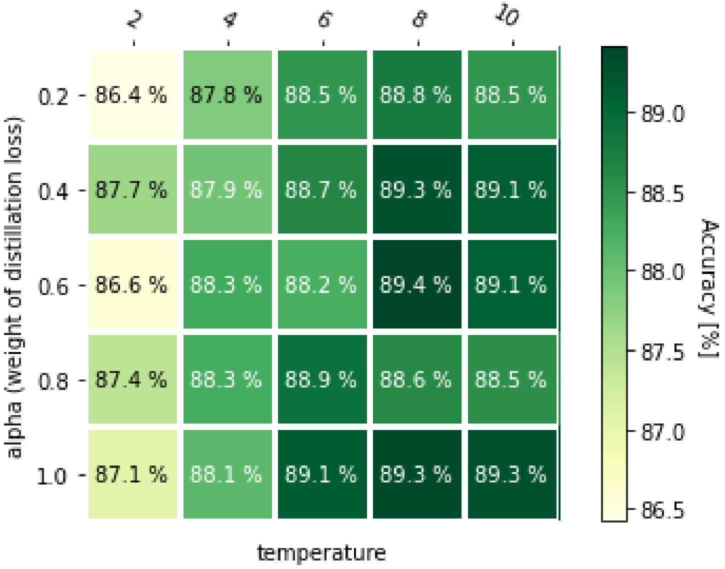
Using the distilled knowledge from teacher, the student was able to achieve a mean accuracy of 98.87%, which is a good improvement over the normal student model (98.6%) and closer to the accuracy attained by the teacher model (98.96%).

4.3 Experiment 2: Effect of Hyperparameters α, T

We then tried a grid search for analysing the effect of hyperparameters α and T on the learning of student model.

- **Dataset:** CIFAR-10 Dataset
- **Teacher Model:** ResNet 18
- **Student Model:** 6 Convolutional layers (with Batch Norm) + 2 FC layers.

For the ResNet18 model, we had to make some modifications in the original architecture to make it possible to run on our dataset, as the original architecture works for larger image sizes and thus contains some down-sampling layers in the start, so we had to remove them. Similarly, we had to use data augmentation and time varying learning rate with SGD to prevent overfitting. Finally, we achieved an accuracy of **93.5 %** with the teacher model, and the student model achieved an accuracy of **87.25 %** without KD.
The heatmap below is the result of the grid search:



Some key observations from the above heatmap:

- Best improvement is of about **2.1%**.
- The given student-teacher pair works best for high temperature due to high confidence of teacher model.
- When $\alpha = 1$, this means no use of labels but still we get **2%** improvement.

Table 2: Comparison of models for CIFAR-10 Dataset

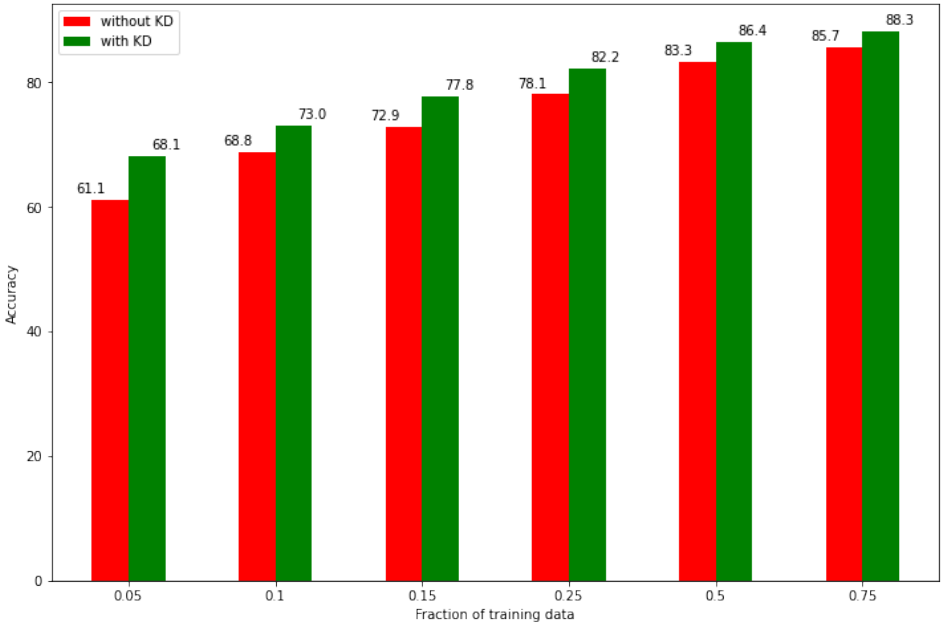
Model	Size	No. of Parameters
Teacher Model	86 MB	11,183,562
Student Model	9 MB	816,938

4.4 Experiment 3: Variation with Amount of Training Data

The motivation of this experiment was that the dark knowledge contains more information than hard label, so the student model should be able to learn from less data.

- **Dataset:** CIFAR-10 Dataset
- **Teacher Model:** ResNet 18
- **Student Model:** 6 Convolutional layers (with Batch Norm) + 2 FC layers.

We obtain the following results:



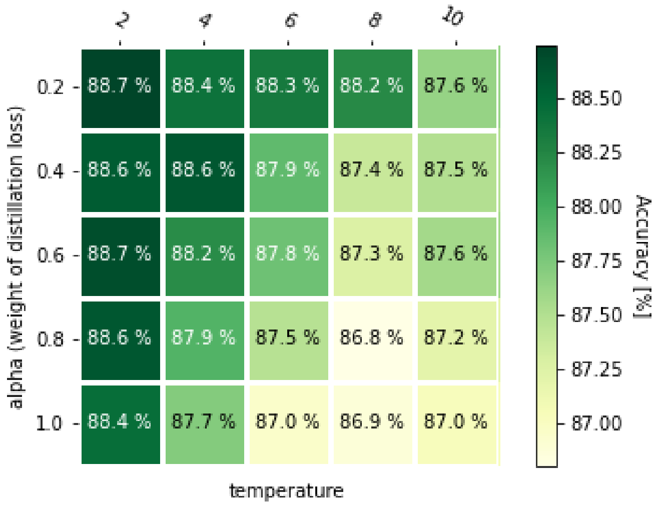
Some key observations from the above graph:

- Even with only **25%** data, the accuracy achieved is comparable to the student model trained from scratch.
- The improvement due to Knowledge Distillation is more prominent for less data.

4.5 Experiment 4: Teacher Free Knowledge Distillation

- **Dataset:** CIFAR-10 Dataset
- **Student Model**: 6 Convolutional layers + 2 FC layers

In this experiment we use the pre-trained student as a teacher (proposed by Yuan et al. [14]) and find the accuracy for different values of α and T.



Using pre-trained student model as teacher has a regularizing effect and we achieve a maximum accuracy of **88.7%** (+1.45%). When we use high values of T, the distribution of teacher's soft predictions becomes similar to uniform distribution and has the same effect as Label Smoothing Regularization.
The student model is smaller and predicts with lower confidence (softer probabilities) than a more complex model like ResNet-18, therefore it is able to obtain good accuracy at lower values of T.

4.6 Experiment 5: Distilled Student as Teacher

- **Dataset:** CIFAR-10 Dataset
- **Student Model**: 6 Convolutional layers + 2 FC layers

In this experiment we use the pre-trained **distilled** student model ($\alpha = 0.6$ and $T = 2$) from the previous experiment as teacher. We try different values of α and T and observe that for $\alpha = 0.4$ and $T = 4$ the student model attains an accuracy of $\mathbf{89}\%$ which is close to the maximum accuracy ($\mathbf{89.4}\%$) obtained by using ResNet-18 as teacher. Distilled student gives soft predictions with better confidence than a normal student when used as a teacher which explains the maximum accuracy obtained in this experiment is at $T = 4$ instead of $T = 2$.

4.7 Experiment 6: Using Teaching Assistants

Problem: Gap between student and teacher model affects the knowledge transfer.
We conducted experiments with 5 different models which are differentiable in terms of number of convolutional layers: 2, 4, 6, 8, 10.
Note that all models are regularized properly and trained for 100 epochs using SGD

Abbreviations used: T: Teacher Model size, S: Student Model Size

Below in figure 4 and 5 are the empirical results for evidence of the above problem:

- The left graph shows that even though the teacher model is improving in terms of accuracy (as shown in 4), but still the student model (S = 2) is not able to benefit from that using knowledge distillation.
- Similarly, the right graph shows that the variation of gain in accuracy with different student sizes.

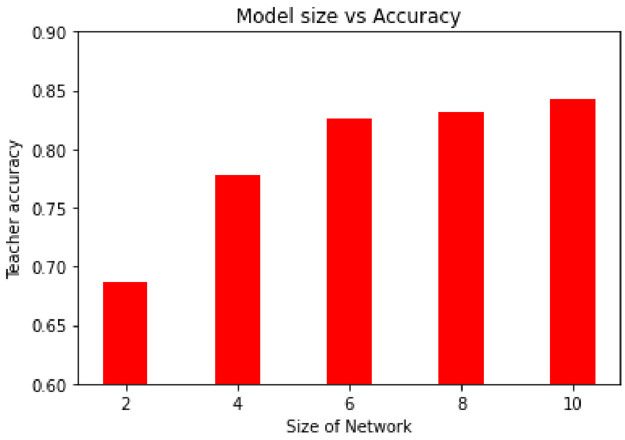


Figure 4: Model Size vs Accuracy

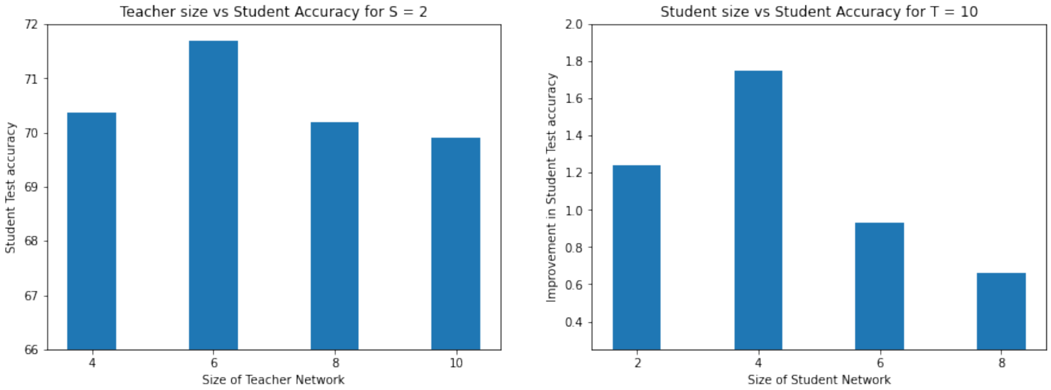


Figure 5: a) Teacher size vs Student Accuracy b) Student Size vs Gain in Student Accuracy

These are the reasons that are mentioned in the paper[11] and seemed very logical are as following:

- Teacher's performance increases, thus it provides better supervision for the student by being a better predictor.
- The teacher is becoming so complex that the student does not have the sufficient capacity or mechanics to mimic her behavior despite receiving hints.
- Teacher's certainty about data increases, thus making its logits (soft targets) less soft. This weakens the knowledge transfer which is done via matching the soft targets.

Reason number 1 is stating that the knowledge transfer is beneficial when the gap between student and teacher model is big, whereas other 2 reasons tells why that would not be so.

Using TA's to bridge the gap: achieved the following results:
Student Size: 2 conv layers, **Teacher Size:** 10 conv layers

Table 3: TA Size vs Student Accuracy

No KD	Baseline KD	TA size = 4	TA size = 6	TA size = 8
68.6%	71.1%	71.5%	72.5%	72.1%

5 Tools and Softwares used

All our models were implemented using Tensorflow [2] and Keras [6]. Tensorflow is an automatic differentiation framework, which has become really popular for Deep Learning models. It allows us to customize the entire training and testing process which was very essential for this project.
Keras is a high-level API that is built on top of TensorFlow. It is extremely user-friendly and comparatively easier than TensorFlow but offers less flexibility.
We trained our models on Google colab notebooks using GPU as hardware accelerator. All python notebooks and models can be obtained at https://github.com/vaithak/Knowledge_Distillation

6 Learnings from the Project

- One of the main learnings that we got from this project is realizing that deep learning is more than just implementing big neural networks and training them on a heavy machine. There are many insights and learning that one can get, if one thinks more about the inner workings and intricacies of machine learning.
- We also got more deep insights into the learning process of deep neural networks as we also had to read many papers and articles even before writing a single line of code.
- We became more familiar with implementing deep learning models, going from an idea to implementation quickly with the help of frameworks like Tensorflow. Also, for the project we had to implement custom learning rate scheduler, custom training step and custom loss function which gave us even more insight to the framework than just using the prebuilt functions.

7 Future Work

Although in our complete project, we did analysis of cases when knowledge is transferred from the logits of teacher model. As stated in the literature review section, there have been a good amount of work in which along with the output logits, the teacher also provides hints to the student using output of the inner feature maps. This can be explored more in the future.

All our empirical analysis was mainly for classification models, as mentioned in the literature review, there also has been some work on knowledge distillation for object detection models [4] . This can be analysed empirically in future.

References

[1] Google deepmind alphafold: using ai for scientific discovery 2020. https://deepmind.com/blog/article/AlphaFold-Using-AI-for-scientific-discovery, 2020.

[2] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2016.

[3] C. Buciluundefined, R. Caruana, and A. Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, page 535–541, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933395. doi: 10.1145/1150402.1150464. URL https://doi.org/10.1145/1150402.1150464.

[4] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker. Learning efficient object detection models with knowledge distillation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 742–751. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/e1e32e235eee1f970470a3a6658dfdd5-Paper.pdf.

[5] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker. Learning efficient object detection models with knowledge distillation. *Advances in neural information processing systems*, 30: 742–751, 2017.

[6] F. Chollet et al. Keras. `https://keras.io`, 2015.

[7] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network, 2015.

[8] A. Kamilaris and F. X. Prenafeta-Boldú. Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147:70–90, Apr 2018. ISSN 0168-1699. doi: 10.1016/j.compag.2018.02.016. URL `http://dx.doi.org/10.1016/j.compag.2018.02.016`.

[9] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[10] Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[11] S.-I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh. Improved knowledge distillation via teacher assistant, 2019.

[12] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets, 2015.

[13] J. Yim, D. Joo, J. Bae, and J. Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141, 2017.

[14] L. Yuan, F. E.H.Tay, G. Li, T. Wang, and J. Feng. Revisit knowledge distillation: a teacher-free framework, 2020.

[15] S. Zagoruyko and N. Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer, 2017.