Title: DeiT: Training ViTs Using Distillation through Attention

Angela Xiao, Liesel Wong, Yunoo Kim
github.com/kimyunoo/4782_final

1 Introduction

Because of their ability to capture more long-range dependencies and contexts, for some vision tasks, ViTs may be preferable over CNNs in some cases. However, ViTs also require a great deal more training data. To address this issue, Touvron and Cord come up with a new "distillation" mechanism where a pre-trained CNN is used as a teacher model in the paper "Training data-efficient image transformers & distillation through attention". This was incorporated into the model via the use of a "distillation token" and a "class token". Using this method, they were able to achieve results comparable to the ViT introduced in the original ViT paper with far less training data. We aim to reproduce the paper's results with a smaller dataset.

2 Chosen Result

The result we aimed to reproduce was the paper's comparison table of validation set accuracies for different distillation methods in Table 1. This table consists of results from several different models. The first is no distillation at all; this would be the baseline model. Then there is soft distillation, where the teacher CNN reports back a probability distribution for classification, and hard distillation, where the teacher CNN reports back only its final decision. Class embedding and distil embedding refer to architectures that only use a class token and a distillation respectively (note that distillation can be done without token architecture, as in usual/soft and hard), while class+distillation refers to the proposed architecture that uses both tokens.

	Supervision		ImageNet top-1 (%)			
method ↓	label	teacher	Ti 224	S 224	B 224	B↑384
DeiT– no distillation	✓	✗	72.2	79.8	81.8	83.1
DeiT– usual distillation	✗	soft	72.2	79.8	81.8	83.2
DeiT– hard distillation	✗	hard	74.3	80.9	83.0	84.0
DeiT⚗: class embedding	✓	hard	73.9	80.9	83.0	84.2
DeiT⚗: distil. embedding	✓	hard	74.6	81.1	83.1	84.4
DeiT⚗: class+distillation	✓	hard	74.5	81.2	83.4	84.5

Table 2.1: Original Results from the Paper, trained for 300 epochs

Model	ImageNet	CIFAR-10	CIFAR-100	Flowers	Cars	iNat-18	iNat-19	im/sec
Grafit ResNet-50 [49]	79.6	_	_	98.2	92.5	69.8	75.9	1226.1
Grafit RegNetY-8GF [49]	_	_	_	99.0	94.0	76.8	80.0	591.6
ResNet-152 [10]	_	_	_	_	_	69.1	_	526.3
EfficientNet-B7 [48]	84.3	98.9	91.7	98.8	94.7	_	_	55.1
ViT-B/32 [15]	73.4	97.8	86.3	85.4	_	_	_	394.5
ViT-B/16 [15]	77.9	98.1	87.1	89.5	_	_	_	85.9
ViT-L/32 [15]	71.2	97.9	87.1	86.4	_	_	_	124.1
ViT-L/16 [15]	76.5	97.9	86.4	89.7	_	_	_	27.3
DeiT-B	81.8	99.1	90.8	98.4	92.1	73.2	77.7	292.3
DeiT-B↑384	83.1	99.1	90.8	98.5	93.3	79.5	81.4	85.9
DeiT-B⚗	83.4	99.1	91.3	98.8	92.9	73.7	78.4	290.9
DeiT-B⚗ ↑384	84.4	99.2	91.4	98.9	93.9	80.1	83.0	85.9

Table 2.2: Table comparing DeiT, ViT, and CNN performance when trained on different datasets; our statistic of interest is the CIFAR-10 column.

These results are relevant because they ultimately show the difference in performance that applying this distillation method has over the original ViT training method, as well as showing which distillation method has the best performance. We aim to recreate the results in Table 1 for the small dataset CIFAR-10 used in the original paper's generalizability experimentation.

3 Methodology

We chose to use CIFAR-10, a dataset consisting of 64,000 32 by 32 images across 10 different classes. This is smaller than ImageNet, which allows us to test performance of these methods on even tighter data constraints. Even so, due to our limited compute and time resources and our need to train multiple versions to compare performance, we decided to train each model for 20 epochs, as opposed to the proposed 7200 epochs for CIFAR-10 the original paper uses for generalization testing. The original paper also rescaled images to 224 x 224 in order to ensure data augmentation consistency over different datasets for the sake of comparison. We are not doing any comparisons, so we decided to omit this step as well. For data augmentation, we decided to use RandAugment out of the methods presented in the original paper, as the other data augmentation methods caused Cuda OOM errors.

For our baseline CNN model, we used ResNet-18. For our teacher CNN, we chose to use an ImageNet-pre-trained regnet_x_400mf (5M params) from torchvision, instead of RegNetY-16GF (84M parameters). We wanted to use a CIFAR-10 pre-trained teacher using checkpoints (we also tried creating a checkpoint with our trained baseline model), but encountered Cuda OOM errors. For the no distillation model, we used cross entropy loss. For the soft distillation and hard distillation models we used soft and hard distillation loss (as described in the original paper) respectively. For the class token/embedding-only model we used cross entropy loss between class token output and ground truth labels. For the distillation token-only model we used hard distillation loss. For the model using both a class and distillation token we attempted both soft and hard distillation loss.

Our model architecture was largely inspired by the ViT (patches, positional embedding, etc.) we implemented in Assignment 3. The architecture also includes dropout, transformer blocks, and a normalization layer as described in the paper. The architecture is flexible; according to the

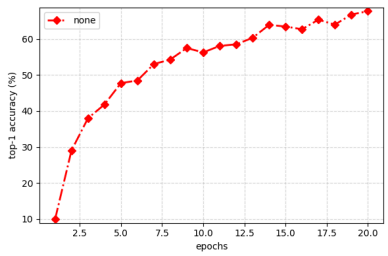
distillation mode, we append distillation tokens and/or class tokens and use the corresponding output heads. We use the parameters described for DeiT-Ti, the smallest version of the model described in the paper.
We used Top-1 accuracy on the CIFAR-10 test set as our evaluation metric.

Results and Analysis

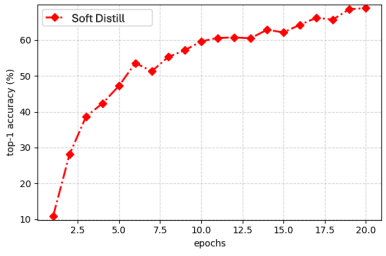
Method 🔻	Supervision		CIFAR-10 Accuracy
	Label	Teacher	
DeiT-Ti - no distillation	✔️	✖️	67.69%
DeiT-Ti - usual distillation	✖️	soft	68.85%
DeiT-Ti - hard distillation	✖️	hard	40.19%
DeiT-Ti ⚗️ - distil embedding	✔️	hard	8.07%
DeiT-Ti ⚗️ - class embedding	✔️	hard	61.69%
DeiT-Ti ⚗️ - class+distillation	✔️	soft	69.19%
DeiT-Ti ⚗️ - class+distillation	✔️	hard	46.11%
ResNet-18 (Pretrained)	n/a	n/a	88.46%

Table 3.1: Our Results from our implementation, trained for 20 epochs

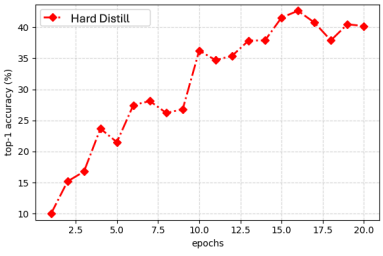
DeiT-Ti - no distillation



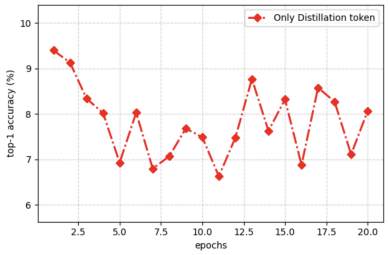
DeiT-Ti - usual distillation



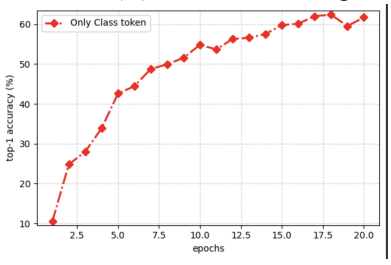
DeiT-Ti - hard distillation



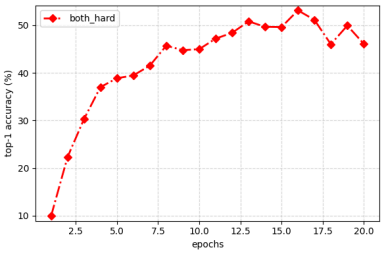
DeiT-Ti ⚗️ - distil embedding



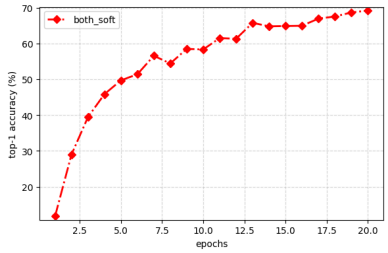
DeiT-Ti ⚗️ - class embedding



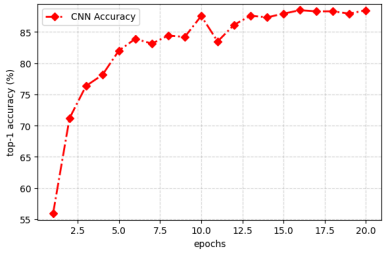
DeiT-Ti ⚗️ - class+distillation hard



DeiT-Ti - class+distillation soft



CNN Baseline



Comparisons across 20 Epochs

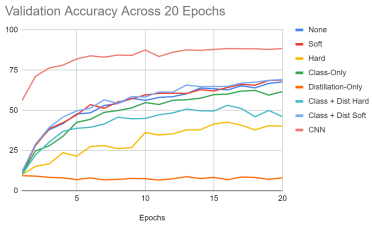


Table 3.2: Validation accuracies across 20 epochs for different methods

Interestingly, the model using both class and distillation tokens with hard distillation loss did not perform as well as the model with no distillation loss. We trained the same architecture with soft distillation loss instead, which achieved a better validation accuracy (69.19%, the highest out of all the distillation modes). We suspect that this may be because our teacher is trained on ImageNet rather than CIFAR-10, so the domain mismatch may cause differences with the paper's observations (CIFAR-10 has 10 classes, while ImageNet has 1000). Additionally, since the distillation token may have had largely inaccurate classification, the addition of it could hinder the model that includes both the class and distillation token, rather than increase its accuracy.

We achieved 88.46% validation accuracy when we trained a ResNet-18, a CNN, for 20 epochs on the same task. Thus, our DeiT model is unfortunately not comparable to CNN performance. This may be due to the teacher domain mismatch as well as having less data augmentation and compute than in the original paper.

Despite our distillation models not achieving accuracy comparable to a CNN or the models described in the paper, we can still conclude that the proposed DeiT architecture is capable of achieving improved performance over ViT (which is equivalent to the DeiT with no distillation loss) and models that do not use token-based distillation.

Overall our results vary significantly from those in the paper, possibly due to the smaller amount of training time. However we can observe that at 20 epochs the test accuracy for the model with class and distillation tokens and soft distillation loss still has a positive slope and does not seem to have plateaued yet**,** so we can assume if we trained the full number of epochs it would have achieved an even higher validation accuracy, perhaps comparable to that achieved in the paper.

4 Reflection

Overall our efforts to implement this paper taught us about distillation strategies and how they can help improve ViTs performance. We also realized the importance of computational resources and just how data-hungry and resource-hungry transformers are. If we wanted to improve upon our work in the future, we could obtain more resources and train for longer with ImageNet rather than CIFAR-10, along with more data augmentation. Additional areas of research could also understand what teacher model to train with, and continue comparisons from there.

5 References

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

Alex Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. Technical report, University of Toronto, 2009. Available at: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In Proceedings of the NIPS 2017 Workshop, 2017.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2020. Training data-efficient image transformers & distillation through attention. arXiv [cs.CV]. Retrieved from http://arxiv.org/abs/2012.12877

Ilija Radosavovic, Raj Prateek Kosaraju, Ross B. Girshick, Kaiming He, and Piotr Doll´ ar. Designing network design spaces. Conference on Computer Vision and Pattern Recognition, 2020

J. D. Hunter. *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering, 9(3):90–95, 2007. doi:10.1109/MCSE.2007.55.