A Review of *"Training data-efficient image transformers & distillation through attention"* [1]

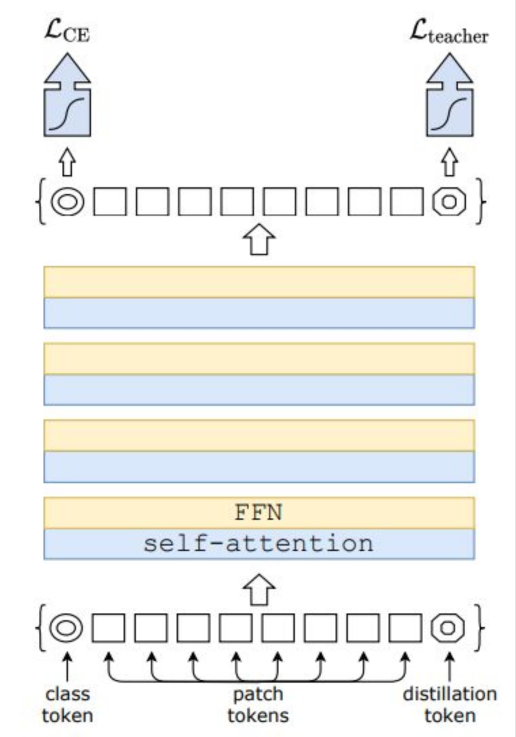
Moazam Soomro, Fatemah Najafali, Alec Kerrigan, and Connor Malley

Overview

- Introduction
- Related work
- Method
- Experiments
- Conclusion

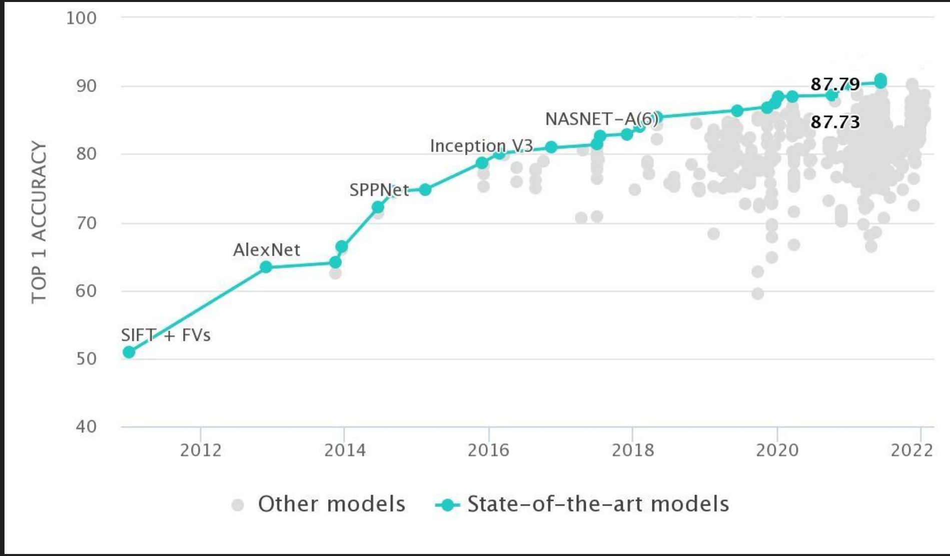
Data Efficient Image Transformers (DeiT)

- Reduces power consumption to produce high performance
- Different training strategies and added distillation token to the model
- Trained on a single node with 4 GPUs in 3 days



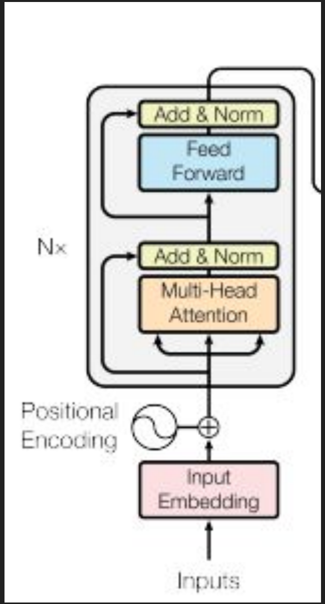
Related work: Image classification

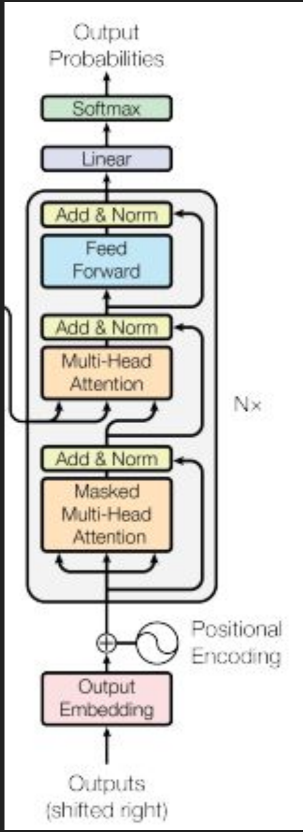
- Evolution of convnets since 2012
- Vision Transformers (ViT) used state of the art ImageNet without using any convolution



Related work: Transformer Architecture

- Convnets for image classification inspired by transformers





What is distillation?

- Distillation is the process in which the knowledge from a larger cumbersome model is transferred to a smaller model, typically for ease of deployment.[2]

- Use a weighted average of two objective functions; the larger models' predictions and the correct labels (can set weight for correct labels to 0)

- Allows for the larger model to be compressed since all neurons may not be utilized

Soft vs. Hard Distillation

- Predictions are made in the form of probabilistic distributions using softmax (hard labels will have a 1 for correct class and 0 for all others)

- Previous papers minimize Kullback-Leibler divergence between student and teacher models (in the “soft” case)[2,11]

- This paper minimizes cross entropy between student logits and the hard teacher prediction (after argmax)

Soft vs. Hard Distillation

$\mathcal{L}_{CE}$ = *Cross Entropy Loss*

KL = *Kullback-Leibler Divergence*

ψ = *Softmax function*

τ = *KL temperature*

λ = *balancing coefficient*

Z_s = *student logits*

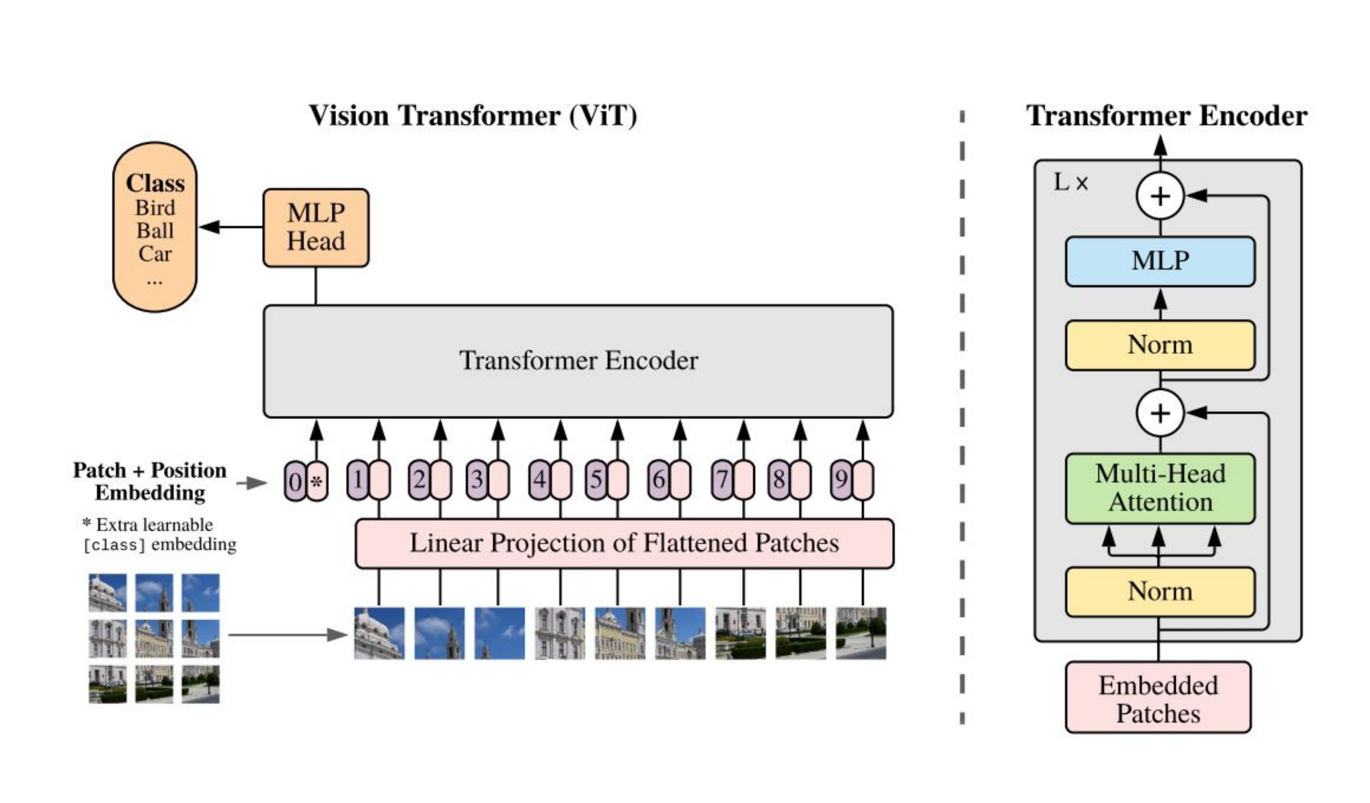
Z_t = *teacher logits*

$y_t = argmax_c(Z_t(c))$

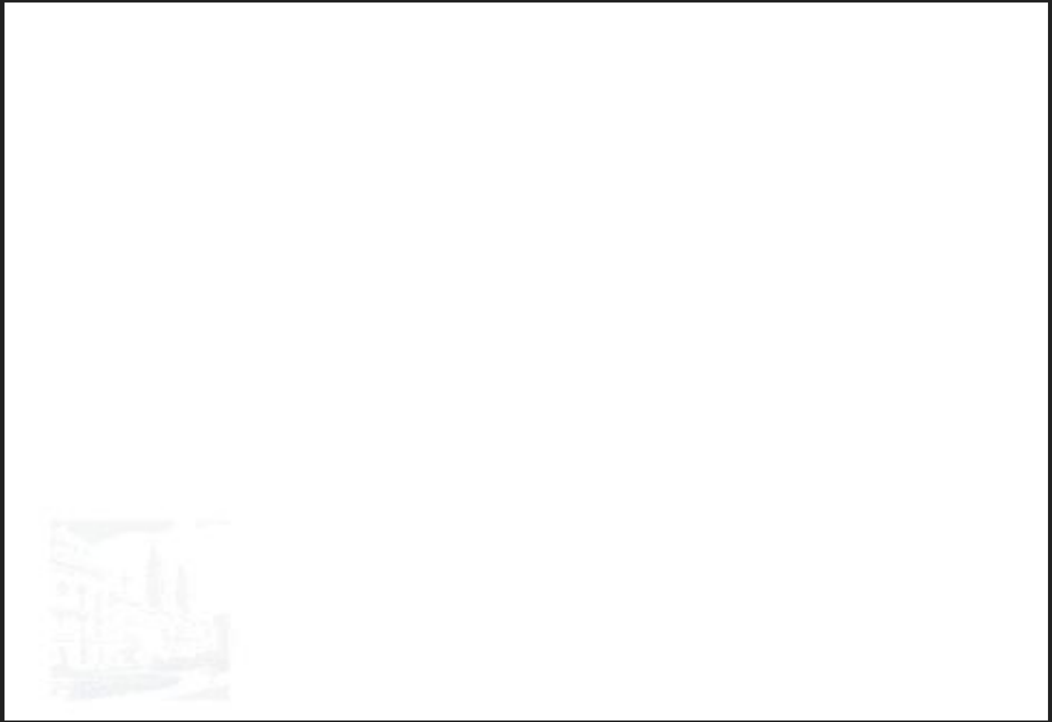
$$\mathcal{L}_{soft} = (1-\lambda)\mathcal{L}_{CE}(\psi(Z_s), y) + \lambda\tau^2 KL(\psi(Z_s/\tau), \psi(Z_t/\tau))$$

$$\mathcal{L}_{hard} = \frac{1}{2}\mathcal{L}_{CE}(\psi(Z_s), y) + \frac{1}{2}\mathcal{L}_{CE}(\psi(Z_s), y_t)$$

Vision Transformer (ViT)[3]



Vision Transformer (ViT)[3]



https://ai.googleblog.com/2020/12/transformers-for-image-recognition-at.html

Distillation Token

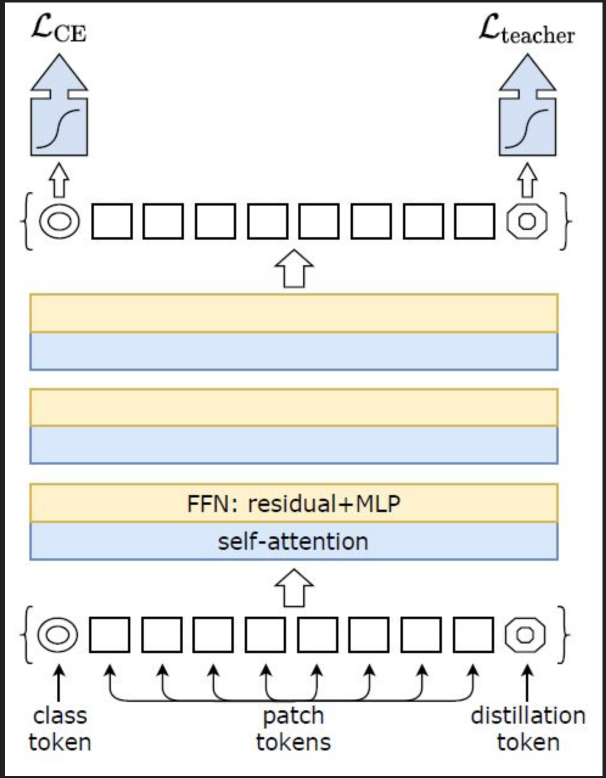
- Similar to CLS token used for final image classification

- Loss calculated using pseudo-label (prediction) from teacher model

- Goal of distillation embedding is to mimic teacher predictions

DeIT⚗️ Architecture[1]

True Label



RegNet Y-16GF Prediction (84M Params)[4]

Size 16 x 16 patches

Vision Transformer

Class and distillation tokens are learned by backprop.

Datasets

- **ImageNet-1k**
  - **1.2 Million images**
  - **1,000 classes**
  - **Used to pretrain/finetune both student and teacher models**

- **JFT-300M**
  - **300 Million images**
  - **18,291 classes**
  - **Used to pretrain other models**

Pretraining and Finetuning Method

- Pretrain on ImageNet-1k at low resolution (224x224)
  Finetune on ImageNet-1k at higher resolution (384x384)

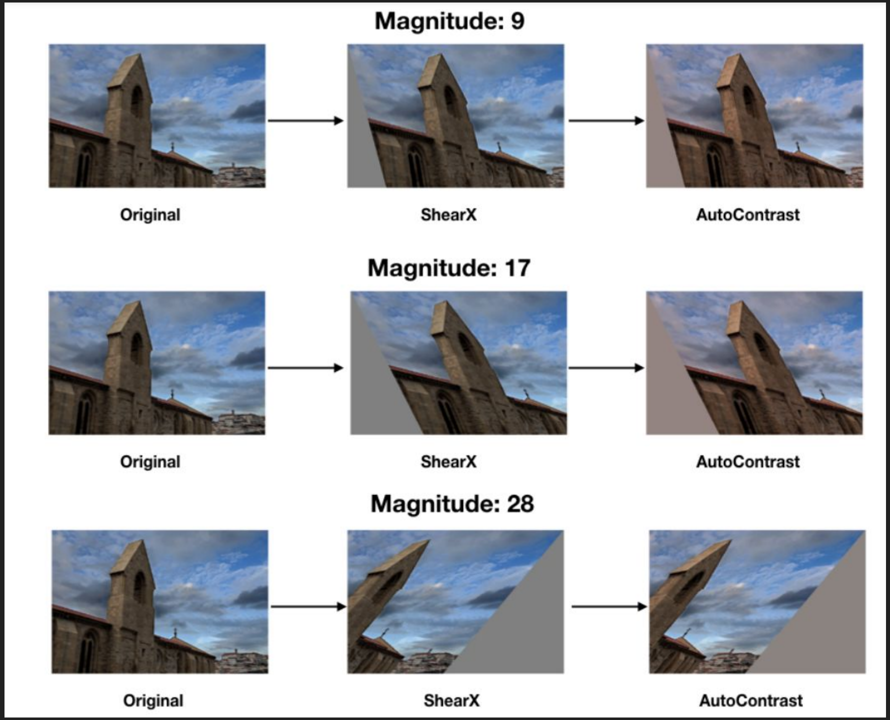
- Authors use extensive data augmentation to give the illusion of a large dataset while processing less images (Rand-Augment)

- Idea is to use the authors distillation method to beat state-of-the-art while utilizing much less data (compared to JFT-300M)

RandAugment[5]

- identity
- rotate
- posterize
- sharpness
- translate-x
- autoContrast
- solarize
- contrast
- shear-x
- translate-y
- equalize
- color
- brightness
- shear-y

1. Choose magnitude of augmentation
2. Choose ***n** out of 14 augmentations*
3. Chain them together



Other Data Augmentations

CutMix[6]

MixUp[7]

Regularization

- Stochastic Depth[8]
  - Drop a subset of layers and bypass with identity function
- Repeated Augment[9]
  - Replicate instances in same batch with different augmentations
- Label Smoothing[10]
  - Set true label to 0.9 and split remaining 0.1 across all other labels

Experiments - Transformer Models

- DeiT-B: reference model (same as ViT-B)
- DeiT-B↑384: fine-tune DeiT at a larger resolution
- DeiT⚗: DeiT with distillation (using distillation tokens)
- Deit-S(small), DeiT-Ti(Tiny): smaller models of DeiT

Model	embedding dimension	#heads	#layers	#params	training resolution	throughput (im/sec)
DeiT-Ti	192	3	12	5M	224	2536
DeiT-S	384	6	12	22M	224	940
DeiT-B	768	12	12	86M	224	292

Experiment - Distillation

Teacher		Student: DeiT-B	
Models	acc.	pretrain	↑384
DeiT-B	81.8	81.9	83.1
RegNetY-4GF	80.0	82.7	83.6
RegNetY-8GF	81.7	82.7	83.8
RegNetY-12GF	82.4	83.0	83.9
RegNetY-16GF	82.9	83.0	84.0

Experiment - Distillation

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

Experiment - Disagreement Analysis

	no distillation		DeiT⚗ student		
	convnet	DeiT	class	distil.	DeiT⚗
groundtruth	0.171	0.182	0.170	0.169	0.166
convnet (RegNetY)	0.000	0.133	0.112	0.100	0.102
DeiT	0.133	0.000	0.109	0.110	0.107
DeiT⚗– class only	0.112	0.109	0.000	0.050	0.033
DeiT⚗– distil. only	0.100	0.110	0.050	0.000	0.019
DeiT⚗– class+distil.	0.102	0.107	0.033	0.019	0.000

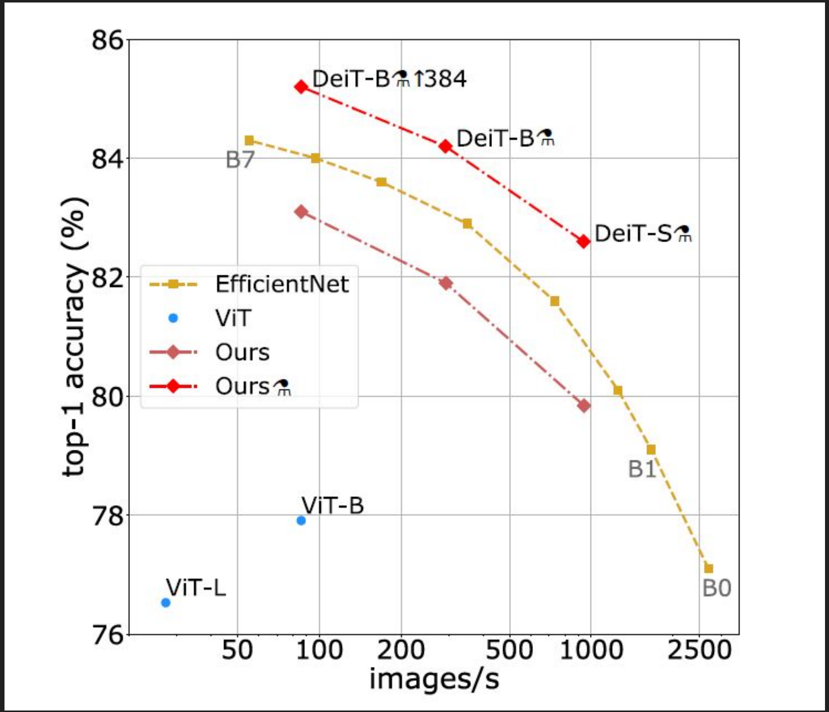
Analysis of the Tokens

- The distillation and class tokens converge to different vectors.

- As the training progresses class and distillation embeddings become similar through the network.

- Authors verified by initializing both randomly and independently yet they converge to same vector.

Efficiency of ViT and EfficientNet vs. DeIT



- Most of the gains in performance from ViT are from the training method
- With no distillation, accuracy is slightly below EfficientNet
- With hard distillation, accuracy is better than EfficientNet

Experiment - Throughput vs Accuracy

- One of DeiT's main advantages is efficiency
- DeiT is faster than its teacher at a similar accuracy
- While small DeiTs are slower than equivilantly accurate CNNs, it scales better

Network	#param.	image size	throughput (image/s)	ImNet top-1	Real top-1	V2 top-1
Convnets						
ResNet-18 [21]	12M	224^2	4458.4	69.8	77.3	57.1
ResNet-50 [21]	25M	224^2	1226.1	76.2	82.5	63.3
ResNet-101 [21]	45M	224^2	753.6	77.4	83.7	65.7
ResNet-152 [21]	60M	224^2	526.4	78.3	84.1	67.0
RegNetY-4GF [40]⋆	21M	224^2	1156.7	80.0	86.4	69.4
RegNetY-8GF [40]⋆	39M	224^2	591.6	81.7	87.4	70.8
RegNetY-16GF [40]⋆	84M	224^2	334.7	82.9	88.1	72.4
EfficientNet-B0 [48]	5M	224^2	2694.3	77.1	83.5	64.3
EfficientNet-B1 [48]	8M	240^2	1662.5	79.1	84.9	66.9
EfficientNet-B2 [48]	9M	260^2	1255.7	80.1	85.9	68.8
EfficientNet-B3 [48]	12M	300^2	732.1	81.6	86.8	70.6
EfficientNet-B4 [48]	19M	380^2	349.4	82.9	88.0	72.3
EfficientNet-B5 [48]	30M	456^2	169.1	83.6	88.3	73.6
EfficientNet-B6 [48]	43M	528^2	96.9	84.0	88.8	73.9
EfficientNet-B7 [48]	66M	600^2	55.1	84.3	_	_
EfficientNet-B5 RA [12]	30M	456^2	96.9	83.7	_	_
EfficientNet-B7 RA [12]	66M	600^2	55.1	84.7	_	_
KDforAA-B8	87M	800^2	25.2	85.8	_	_

Transformers						
ViT-B/16 [15]	86M	384^2	85.9	77.9	83.6	_
ViT-L/16 [15]	307M	384^2	27.3	76.5	82.2	_
DeiT-Ti	5M	224^2	2536.5	72.2	80.1	60.4
DeiT-S	22M	224^2	940.4	79.8	85.7	68.5
DeiT-B	86M	224^2	292.3	81.8	86.7	71.5
DeiT-B↑384	86M	384^2	85.9	83.1	87.7	72.4
DeiT-Ti⚗	6M	224^2	2529.5	74.5	82.1	62.9
DeiT-S⚗	22M	224^2	936.2	81.2	86.8	70.0
DeiT-B⚗	87M	224^2	290.9	83.4	88.3	73.2
DeiT-Ti⚗ / 1000 epochs	6M	224^2	2529.5	76.6	83.9	65.4
DeiT-S⚗ / 1000 epochs	22M	224^2	936.2	82.6	87.8	71.7
DeiT-B⚗ / 1000 epochs	87M	224^2	290.9	84.2	88.7	73.9
DeiT-B⚗ ↑384	87M	384^2	85.8	84.5	89.0	74.8
DeiT-B⚗ ↑384 / 1000 epochs	87M	384^2	85.8	85.2	89.3	75.2

Generalizing DeiT

- DeiT outperforms other methods on a wide variety of datasets when pretrained on ImageNet
- On a smaller dataset (CIFAR), when not pretrained, DeiT underperforms CNNs

Table 7: We compare Transformers based models on different transfer learning task with ImageNet pre-training. We also report results with convolutional architectures for reference.

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

Method	RegNetY-16GF	DeiT-B	DeiT-B⚗
Top-1	98.0	97.5	98.5

Strengths

- Accuracy beats state of the art convnets with similar throughput (EfficientNet).
- Even without distillation, their training method produces better results than the original ViT, with an identical architecture.
- Combination of data augmentation and regularization methods could prove to be useful on other tasks or in other architectures.
- Introduces a new type of ‘hybrid’ architecture, since the knowledge from the convnet is learned within the transformer.

Weaknesses

- Paper focuses entirely on softmax probabilities
  - No consideration of distilling logits or image encoding
- Only CLS token for distillation was considered
  - Transformers can pool information in a variety of ways
  - Mean pooling
  - 1D conv
- Are gains from the paper’s method, or from ViT just being better?
  - Again, would help to try other distillation methods
- Rather handwavey on why ConvNets are better teachers than other ViT nets
  - Defeats the purpose if we need an entirely different architecture for teacher

Conclusion

- Distillation is the process of using a pre-trained model to teach a different, typically smaller, model
- DeiT designed an efficient method for distillation
- Uses a single distillation token, rather than distilling on normal output
- Outperformed previous state of the art in both accuracy and images per second
- Generalized to perform strongly on a variety of datasets

References

1. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. arXiv [cs.CV]. Opgehaal van http://arxiv.org/abs/2012.12877
2. Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. arXiv [stat.ML]. Opgehaal van http://arxiv.org/abs/1503.02531
3. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., … Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. CoRR, abs/2010.11929. Opgehaal van https://arxiv.org/abs/2010.11929
4. Radosavovic, I., Kosaraju, R. P., Girshick, R. B., He, K., & Dollár, P. (2020). Designing Network Design Spaces. CoRR, abs/2003.13678. Opgehaal van https://arxiv.org/abs/2003.13678
5. Cubuk, E. D., Zoph, B., Shlens, J., & Le, Q. V. (2019). RandAugment: Practical data augmentation with no separate search. CoRR, abs/1909.13719. Opgehaal van http://arxiv.org/abs/1909.13719
6. Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. CoRR, abs/1905.04899. Opgehaal van http://arxiv.org/abs/1905.04899
7. Zhang, H., Cissé, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond Empirical Risk Minimization. CoRR, abs/1710.09412. Opgehaal van http://arxiv.org/abs/1710.09412

References

8. Huang, G., Sun, Y., Liu, Z., Sedra, D., & Weinberger, K. Q. (2016). Deep Networks with Stochastic Depth. CoRR, abs/1603.09382. Opgehaal van http://arxiv.org/abs/1603.09382

9. Hoffer, E., Ben-Nun, T., Hubara, I., Giladi, N., Hoefler, T., & Soudry, D. (2019). Augment your batch: better training with larger batches. CoRR, abs/1901.09335. Opgehaal van http://arxiv.org/abs/1901.09335

10. Müller, R., Kornblith, S., & Hinton, G. E. (2019). When Does Label Smoothing Help? CoRR, abs/1906.02629. Opgehaal van http://arxiv.org/abs/1906.02629

11. Wei, L., Xiao, A., Xie, L., Chen, X., Zhang, X., & Tian, Q. (2020). Circumventing Outliers of AutoAugment with Knowledge Distillation. CoRR, abs/2003.11342. Opgehaal van https://arxiv.org/abs/2003.11342