DeiT III: Revenge of the ViT

Hugo Touvron, Matthieu Cord, Hervé Jégou

▶ **To cite this version:**

Hugo Touvron, Matthieu Cord, Hervé Jégou. DeiT III: Revenge of the ViT. 17th European Conference on Computer Vision (ECCV 2022), Oct 2022, Tel Aviv, Israel. 10.1007/978-3-031-20053-3_30 . hal-03945731

HAL Id: hal-03945731
https://hal.science/hal-03945731

Submitted on 18 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DeiT III: Revenge of the ViT

Hugo Touvron[⋆,†] Matthieu Cord[†] Hervé Jégou[⋆]

[⋆]Meta AI [†]Sorbonne University

Abstract

A Vision Transformer (ViT) is a simple neural architecture amenable to serve several computer vision tasks. It has limited built-in architectural priors, in contrast to more recent architectures that incorporate priors either about the input data or of specific tasks. Recent works show that ViTs benefit from self-supervised pre-training, in particular BerT-like pre-training like BeiT.

In this paper, we revisit the supervised training of ViTs. Our procedure builds upon and simplifies a recipe introduced for training ResNet-50. It includes a new simple data-augmentation procedure with only 3 augmentations, closer to the practice in self-supervised learning. Our evaluations on Image classification (ImageNet-1k with and without pre-training on ImageNet-21k), transfer learning and semantic segmentation show that our procedure outperforms by a large margin previous fully supervised training recipes for ViT. It also reveals that the performance of our ViT trained with supervision is comparable to that of more recent architectures. Our results could serve as better baselines for recent self-supervised approaches demonstrated on ViT.

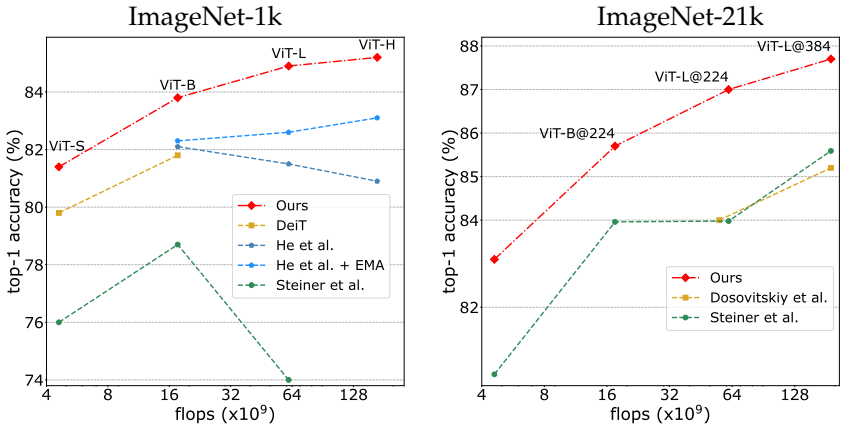

Figure 1: Comparison of training recipes for (*left*) vanilla vision transformers trained on ImageNet-1k and evaluated at resolution 224×224, and (*right*) pre-trained on ImageNet-21k at 224×224 and fine-tuned on ImageNet-1k at resolution 224×224 or 384×384.

1 Introduction

After their vast success in NLP, transformers models [55] and their derivatives are increasingly popular in computer vision. They are increasingly used in image classification [13], detection & segmentation [3], video analysis, etc. In particular, the vision transformers (ViT) of Dosovistky et al. [13] are a reasonable alternative to convolutional architectures. This supports the adoption of transformers as a general architecture able to learn convolutions as well as longer range operations through the attention process [5, 8]. In contrast, convolutional networks [20, 27, 29, 41] implicitly offer built-in translation invariance. As a result their training does not have to learn this prior. It is therefore not surprising that hybrid architectures that include convolution converge faster than vanilla transformers [18].

Because they incorporate as priors only the co-localisation of pixels in patches, transformers have to learn about the structure of images while optimizing the model such that it processes the input with the objective of solving a given task. This can be either reproducing labels in the supervised case, or other proxy tasks in the case of self-supervised approaches. Nevertheless, despite their huge success, there has been only few works in computer vision studying how to efficiently train vision transformers, and in particular on a midsize dataset like ImageNet-1k. Since the work of Dosovistky et al. [13], the training procedures are mostly variants from the proposal of Touvron et al. [48] and Steiner et al. [42]. In contrast, multiple works have proposed alternative architectures by introducing pooling, more efficient attention, or hybrid architectures re-incorporating convolutions and a pyramid structure. These new designs, while being particularly effective for some tasks, are less general. One difficult question to address is whether the improved performance is due to a specific architectural design, or because it facilitates the optimization as suggested it is the case for convolutions with ViTs [60].

Recently, self-supervised approaches inspired by the popular BerT pre-training have raised hopes for a BerT moment in computer vision. There are some analogies between the fields of NLP and computer vision, starting with the transformer architecture itself. However these fields are not identical in every way: The modalities processed are of different nature (continuous versus discrete). Computer vision offer large annotated databases like ImageNet [40], and fully supervised pre-training on ImageNet is effective for handling different downstream tasks such as transfer learning [37] or semantic segmentation.

Without further work on fully supervised approaches on ImageNet it is difficult to conclude if the intriguing performance of self-supervised approaches like BeiT [2] is due to the training, e.g. data augmentation, regularization, optimization, or to an underlying mechanism that is capable of learning more general implicit representations. In this paper, we do not pretend to answer this difficult question, but we want to feed this debate by renewing the training procedure for vanilla ViT architectures. We hope to contribute to a better understanding on how to fully exploit the potential of transformers and of the importance of BerT-like pre-training. Our work builds upon the recent state of the art on fully supervised and self-supervised approaches, with new insights regarding data-augmentation. We propose new training recipes for vision transformers on ImageNet-1k and ImageNet-21k. The main ingredients are as follows:

- We build upon the work of Wightman et al. [57] introduced for ResNet50. In

particular we adopt a binary cross entropy loss for Imagenet1k only training. We adapt this method by including ingredients that significantly improve the training of large ViT [51], namely stochastic depth [24] and LayerScale [51].

- **3-Augment**: is a simple data augmentation inspired by that employed for self-supervised learning. Surprisingly, with ViT we observe that it works better than the usual automatic/learned data-augmentation employed to train vision transformers like RandAugment [6].

- **Simple Random Cropping** is more effective than Random Resize Cropping when pre-training on a larger set like ImageNet-21k.

- **A lower resolution** at training time. This choice reduces the train-test discrepancy [53] but has not been much exploited with ViT. We observe that it also has a regularizing effect for the largest models by preventing overfitting. For instance, for a target resolution of 224×224, a ViT-H pre-trained at resolution 126×126 (81 tokens) achieves a better performance on ImageNet-1k than when pre-training at resolution 224×224 (256 tokens). This is also less demanding at pre-training time, as there are 70% fewer tokens. From this perspective it offers similar scaling properties as mask-autoencoders [19].

Our "new" training strategies do not saturate with the largest models, making another step beyond the Data-Efficient Image Transformer (DeiT) by Touvron et al. [48]. As a result, we obtain a competitive performance in image classification and segmentation, even when compared to recent popular architectures such as SwinTransformers [31] or modern convnet architectures like ConvNext [32]. Below we point out a few interesting outcomes.

- We leverage models with more capacity even on midsize datasets. For instance we reach 85.2% in top-1 accuracy when training a ViT-H on ImageNet1k only, which is an improvement of $+5.1\%$ over the best ViT-H with supervised training procedure reported in the literature at resolution 224×224.

- Our training procedure for ImageNet-1k allow us to train a **billion-parameter ViT-H** (52 layers) without any hyper-parameter adaptation, just using the same stochastic depth drop-rate as for the ViT-H. It attains 84.9% at 224×224, i.e., +0.2% higher than the corresponding ViT-H trained in the same setting.

- Without sacrificing performance, we **divide by more than 2** the number of GPUs required and the training time for ViT-H, making it effectively possible to train such models without a reduced amount of resources. This is thanks to our pre-training at lower resolution, which reduces the peak memory.

- For ViT-B and Vit-L models, our supervised training approach is on par with BerT-like self-supervised approaches [2, 19] with their default setting and when using the same level of annotations and less epochs, both for the tasks of image classification and of semantic segmentation.

- With this improved training procedure, a vanilla ViT closes the gap with recent state-of-the art architectures, often offering better compute/performance trade-offs. Our models are also comparatively better on the additional test set

ImageNet-V2 [39], which indicates that our trained models generalize better to another validation set than most prior works.

- An ablation on the effect of the crop ratio employed in transfer learning classification tasks. We observe that it has a noticeable impact on the performance but that the best value depends a lot on the target dataset/task.

2 Related work

Vision Transformers were introduced by Dosovitskiy et al. [13]. This architecture, which derives from the transformer by Vaswani et al. [55], is now used as an alternative to convnets in many tasks: image classification [13, 48], detection [3, 31], semantic segmentation [2, 31] video analysis [17, 35], to name only a few. This greater flexibility typically comes with the downside that they need larger datasets, or the training must be adapted when the data is scarcer [14, 48]. Many variants have been introduced to reduce the cost of attention by introducing for example more efficient attention [16, 17, 31] or pooling layers [21, 31, 56]. Some papers re-introduce spatial biases specific to convolutions within hybrid architectures [18, 58, 60]. These models are less general than vanilla transformers but generally perform well in certain computer vision tasks, because their architectural priors reduce the need to learn from scratch the task biases. This is especially important for smaller models, where specialized models do not have to devote some capacity to reproduce known priors such as translation invariance. The models are formally less flexible but they do not require sophisticated training procedures.

Training procedures: The first procedure proposed in the ViT paper [13] was mostly effective for larger models trained on large datasets. In particular the ViT were not competitive with convnets when trained from scratch on ImageNet. Touvron et al. [48] showed that by adapting the training procedure, it is possible to achieve a performance comparable to that of convnets with Imagenet training only. After this Data Efficient Image Transformer procedure (DeiT), only few adaptations have been proposed to improve the training vision transformers. Steiner et al. [42] published a complete study on how to train vision transformers on different datasets by doing a complete ablation of the different training components. Their results on ImageNet [40] are slightly inferior to those of DeiT but they report improvements on ImageNet-21k compared to Dosovitskiy et al. [13]. The self-supervised approach referred to as masked auto-encoder (MAE) [19] proposes an improved supervised baseline for the larger ViT models.

BerT pre-training: In the absence of a strong fully supervised training procedure, BerT [10]-like approaches that train ViT with a self-supervised proxy objective, followed by full finetuning on the target dataset, seem to be the best paradigm to fully exploit the potential of vision transformers. Indeed, BeiT [2] or MAE [19] significantly outperform the fully-supervised approach, especially for the largest models. Nevertheless, to date these approaches have mostly shown their interest in the context of mid-size datasets. For example MAE [19] report its most impressive results when pre-training on ImageNet-1k with a full finetuning on ImageNet-1k. When

pre-training on ImageNet-21k and finetuning on ImageNet-1k, BeiT [2] requires a full 90-epochs finetuning on ImageNet-21k followed by another full finetuning on ImageNet-1k to reach its best performance, suggesting that a large labeled dataset is needed so that BeiT realizes its best potential. A recent work suggests that such auto-encoders are mostly interesting in a data starving context [15], but this questions their advantage in the case where more labelled data is actually available.

Data-augmentation: For supervised training, the community commonly employs data-augmentations offered by automatic design procedures such as RandAugment [6] or Auto-Augment [7]. These data-augmentations seem to be essential for training vision transformers [48]. Nevertheless, papers like TrivialAugment [34] and Uniform Augment [30] have shown that it is possible to reach interesting performance levels when simplifying the approaches. However these approaches were initially optimized for convnets. In our work we propose to go further in this direction and drastically limit and simplify data-augmentation: we introduce a data-augmentation policy that employs only 3 different transformations randomly drawn with uniform probability. That's it.

3 Revisit training & pre-training for Vision Transformers

In this section, we present our training procedure for vision transformers and compare it with existing approaches. We detail the different ingredients in Table 1. Building upon Wightman et al. [57] and Touvron et al. [48], we introduce several changes that have a significant impact on the final model accuracy.

3.1 Regularization & loss

Stochastic depth is a regularization method that is especially useful for training deep networks. We use a uniform drop rate across all layers and adapt it according to the model size [51]. Table 13 (A) gives the stochastic depth drop-rate per model.

LayerScale. We use LayerScale [51]. This method was introduced to facilitate the convergence of deep transformers. With our training procedure, we do not have convergence problems, however we observe that LayerScale allows our models to attain a higher accuracy for the largest models. In the original paper [51], the initialization of LayerScale is adapted according to the depth. In order to simplify the method we use the same initialization (10^{-4}) for all our models.

Binary Cross entropy. Wigthman et al. [57] adopt a binary cross-entropy (BCE) loss instead of the more common cross-entropy (CE) to train ResNet-50. They conclude that the gains are limited compared to the CE loss but that this choice is more convenient when employed with Mixup [64] and CutMix [63]. For larger ViTs and with our training procedure on ImageNet-1k, the BCE loss provides us a significant improvement in performance, see an ablation in Table 4. We did not achieve compelling results during our exploration phase on Imagenet21k, and therefore keep CE when pre-training with this dataset as well as for the subsequent fine-tuning.

Table 1: Summary of our training procedures with ImageNet-1k and ImageNet-21k. We also provide DeiT [48], Wightman et al [57] and Steiner et al. [42] baselines for reference. Adapt. means the hparams is adapted to the size of the model. For finetuning to higher resolution with model pre-trained on ImageNet-1k only we use the finetuning procedure from DeiT see section A for more details.

	Previous approaches				Ours		
Procedure →	ViT	Steiner	DeiT	Wightman	ImNet-1k	ImNet-21k	
Reference	[13]	et al. [42]	[48]	et al. [57]		Pretrain.	Finetune.
Batch size	4096	4096	1024	2048	2048	2048	2048
Optimizer	AdamW	AdamW	AdamW	LAMB	LAMB	LAMB	LAMB
LR	3.10^{-3}	3.10^{-3}	1.10^{-3}	5.10^{-3}	3.10^{-3}	3.10^{-3}	3.10^{-4}
LR decay	cosine	cosine	cosine	cosine	cosine	cosine	cosine
Weight decay	0.1	0.3	0.05	0.02	0.02	0.02	0.02
Warmup epochs	3.4	3.4	5	5	5	5	5
Label smoothing ε	0.1	0.1	0.1	✗	✗	0.1	0.1
Dropout	✓	✓	✗	✗	✗	✗	✗
Stoch. Depth	✗	✓	✓	✓	✓	✓	✓
Repeated Aug	✗	✗	✓	✓	✓	✗	✗
Gradient Clip.	1.0	1.0	✗	1.0	1.0	1.0	1.0
H. flip	✓	✓	✓	✓	✓	✓	✓
RRC	✓	✓	✓	✓	✓	✗	✗
Rand Augment	✗	Adapt.	9/0.5	7/0.5	✗	✗	✗
3 Augment (ours)	✗	✗	✗	✗	✓	✓	✓
LayerScale	✗	✗	✗	✗	✓	✓	✓
Mixup alpha	✗	Adapt.	0.8	0.2	0.8	✗	✗
Cutmix alpha	✗	✗	1.0	1.0	1.0	1.0	1.0
Erasing prob.	✗	✗	0.25	✗	✗	✗	✗
ColorJitter	✗	✗	✗	✗	0.3	0.3	0.3
Test crop ratio	0.875	0.875	0.875	0.95	1.0	1.0	1.0
Loss	CE	CE	CE	BCE	BCE	CE	CE

The optimizer is LAMB [61], a derivative of AdamW [33]. It includes gradient clipping by default in `Apex`'s [1] implementation.

3.2 Data-augmentation

Since the advent of AlexNet, there has been significant modifications to the data-augmentation procedures employed to train neural networks. Interestingly, the same data augmentation, like RandAugment [6], is widely employed for ViT while their policy was initially learned for convnets. Given that the architectural priors and biases are quite different in these architectures, the augmentation policy may not be adapted, and possibly overfitted considering the large amount of choices involved in their selection. We therefore revisit this prior choice.

3-Augment: We propose a simple data augmentation inspired by what is used in self-supervised learning (SSL). We consider the following transformations:

- Grayscale: This favors color invariance and give more focus on shapes.
- Solarization: This adds strong noise on the colour to be more robust to the variation of colour intensity and so focus more on shape.

Data-Augmentation				ImageNet-1k		
ColorJitter	Grayscale	Gaussian Blur	Solarization	Val	Real	V2
0.3	✗	✗	✗	81.4	86.1	70.3
0.3	✓	✗	✗	81.0	86.0	69.7
0.3	✓	✓	✗	82.7	87.6	**72.7**
0.3	✓	✓	✓	**83.1**	**87.7**	72.6
0.0	✓	✓	✓	**83.1**	**87.7**	72.0

Table 2: Ablation of the components of our data-augmentation strategy with ViT-B on ImageNet-1k.

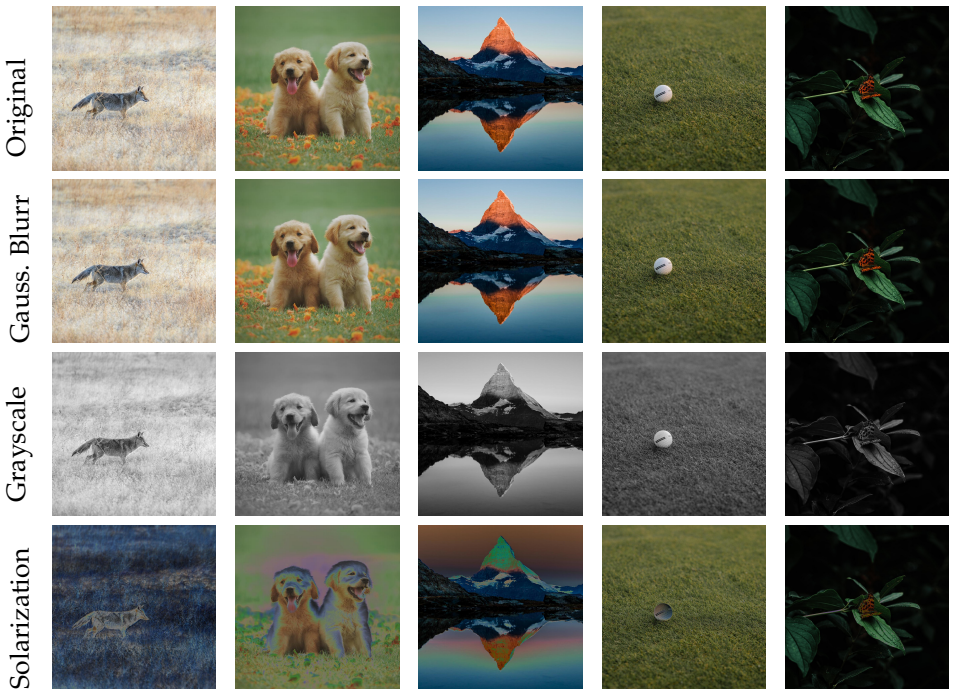

Figure 2: Illustration of the 3 types of data-augmentations used in 3-Augment.

- Gaussian Blur: In order to slightly alter details in the image.

For each image, we select only one of this data-augmentation with a uniform probability over 3 different ones. In addition to these 3 aumgnentations choices, we include the common color-jitter and horizontal flip. Figure 2 illustrates the different augmentations used in our 3-Augment approach. In Table 2 we provide an ablation on our different data-augmentation components.

3.3 Cropping

Random Resized Crop (RRC) was introduced in the GoogleNet [43] paper. It serves as a regularisation to limit model overfitting, while favoring that the decision done by the model is invariant to a certain class of transformations. This data augmentation was deemed important on Imagenet1k to prevent overfitting, which happens to occur rapidly with modern large models.

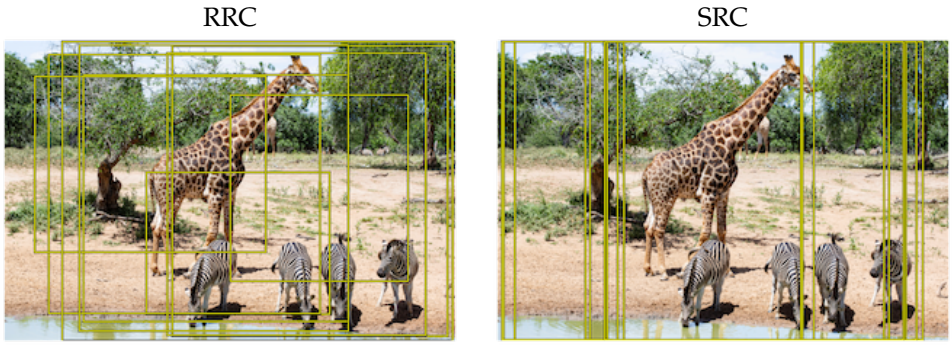

Figure 3: Example of crops selected by two strategies: Resized Crop and Simple Random Crop.

This cropping strategy however introduces some discrepancy between train and test images in terms of the aspect ratio and the apparent size of objects [53]. Since ImageNet-21k includes significantly more images, it is less prone to overfitting. Therefore we question whether the benefit of the strong RRC regularization compensates for its drawback when training on larger sets.

Simple Random Crop (SRC) is a much simpler way to extract crops. It is similar to the original cropping choice proposed in AlexNet [27]: We resize the image such that the smallest side matches the training resolution. Then we apply a reflect padding of 4 pixels on all sides, and finally we apply a square Crop of training size randomly selected along the x-axis of the image.

Figure 3 vizualizes cropping boxes sampled for RRC and SRC. RRC provides a lot of diversity and very different sizes for crops. In contrast SRC covers a much larger fraction of the image overall and preserve the aspect ratio, but offers less diversity: The crops overlaps significantly. As a result, when training on ImageNet-1k the performance is better with the commonly used RRC. For instance a ViT-S reduces its top-1 accuracy by -0.9% if we do not use RRC.

However, in the case of ImageNet-21k ($\times 10$ bigger than ImageNet-1k), there is less risk of overfitting and increasing the regularisation and diversity offered by RRC is less important. In this context, SRC offers the advantage of reducing the discrepancy in apparent size and aspect ratio. More importantly, it gives a higher chance that the actual label of the image matches that of the crop: RRC is relatively aggressive in terms of cropping and in many cases the labelled object is not even present in the crop, as shown in Figure 4 where some of the crops do not contain the labelled object. For instance, with RRC there is a crop no zebra in the left example, or no train in three of the crops from the middle example. This is more unlikely to happen with SRC, which covers a much larger fraction of the image pixels. In Table 5 we provide an ablation of random resized crop on ImageNet-21k, where we see that these observations translate as a significant gain in performance.

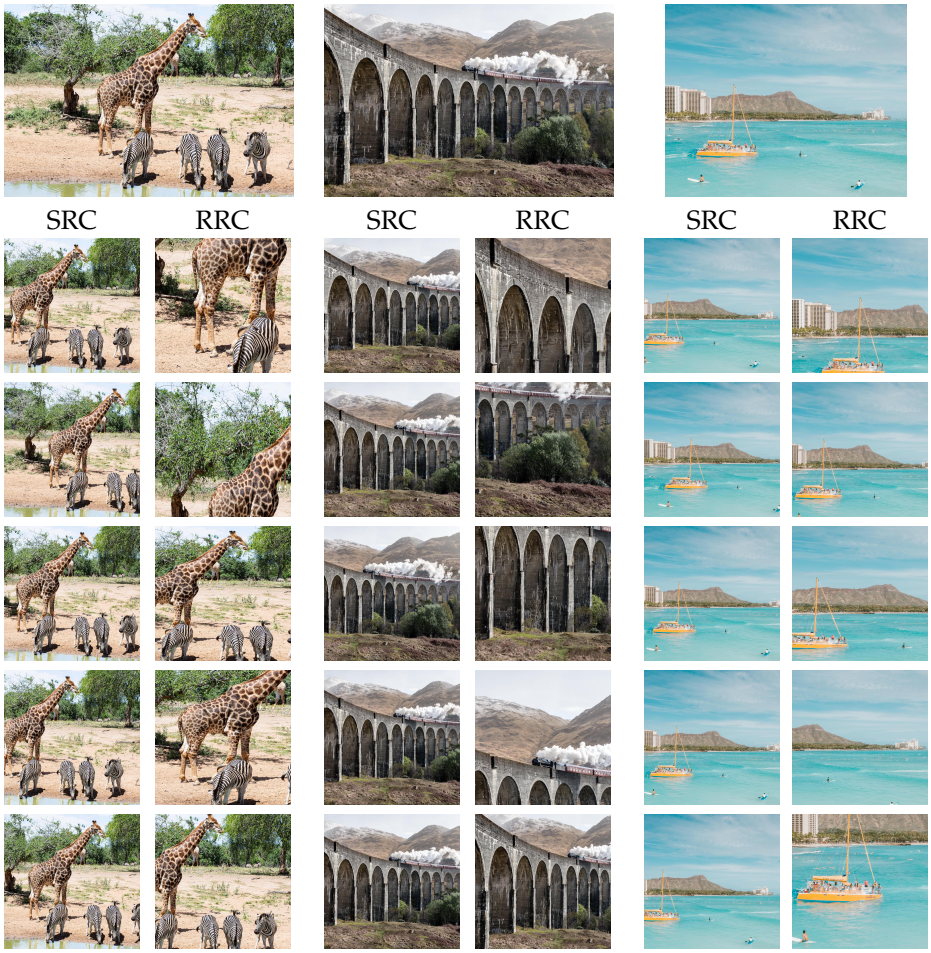

Figure 4: Illustration of Random Resized Crop (RRC) and Simple Random Crop (SRC). The usual RRC is a more aggressive data-augmentation than SRC: It has a more important regularizing effect and avoids overfitting by giving more variability to the images. At the same time it introduces a discrepancy of scale and aspect-ratio. It also leads to labeling errors, for instance when the object is not in the cropped region (e.g., train or boat). On Imagenet1k this regularization is overall regarded as beneficial. However our experiments show that it is detrimental on Imagenet21k, which is less prone to overfitting.

4 Experiments

This section includes multiple experiments in image classification, with a special emphasis on Imagenet1k [9, 39, 40]. We also report results for downstream tasks in fine-grained classification and segmentation. We include a large number of ablations to better analyze different effects, such as the importance of the training resolution and longer training schedules. We provide additional results in the appendices.

4.1 Baselines and default settings

The main task that we consider in this paper for the evaluation of our training procedure is image classification. We train on Imagenet1k-train and evaluate on Imagenet1k-val, with results on ImageNet-V2 to control overfitting. We also consider the case where we can pretrain on ImageNet-21k, Finally, we report transfer learning results on 6 different datasets/benchmarks.

Default setting. When training on ImageNet-1k only, by default we train during 400 epochs with a batch size 2048, following prior works [51, 60]. Unless specified otherwise, both the training and evaluation are carried out at resolution 224×224 (even though we recommend to train at a lower resolution when targeting 224×224 at inference time).

When pre-training on ImageNet-21k, we pre-train by default during 90 epochs at resolution 224×224, followed by a finetuning of 50 epochs on on ImageNet-1k. In this context we consider two fine-tuning resolutions: 224×224 and 384×384.

4.2 Ablations

4.2.1 Impact of training duration

In Figure 5 we provide an ablation on the number of epochs, which show that ViT models do not saturate as rapidly as the DeiT training procedure [48] when we increase the number of epochs beyond the 400 epochs adopted for our baseline.

For ImageNet-21k pre-training, we use 90 epochs for pre-training as in a few works [31, 49]. We finetune during 50 epochs on ImageNet-1k [49] and marginally adapt the stochastic depth parameter. We point out that this choice is mostly for the sake of consistency across models: we observe that training 30 epochs also provides similar results.

4.2.2 Data-Augmentation

In Table 3 we compare our handcrafted data-augmentation 3-Augment with existing learned augmentation methods. With the ViT architecture, our data-augmentation is the most effective while being simpler than the other approaches. Since previous augmentations were introduced on convnets, we also provide results for a ResNet-50. In this case previous augmentation policies have similar (RandAugment, Trivial-Augment) or better results (Auto-Augment) on the validation set.

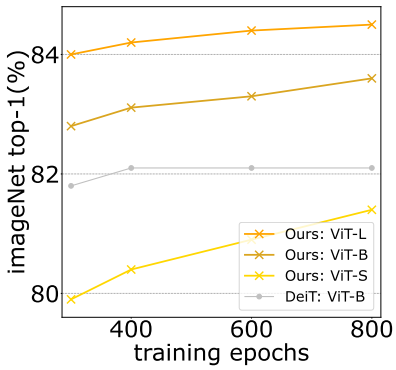

Figure 5: Top-1 accuracy on ImageNet-1k only at resolution 224×224 with our training recipes and a different number of epochs

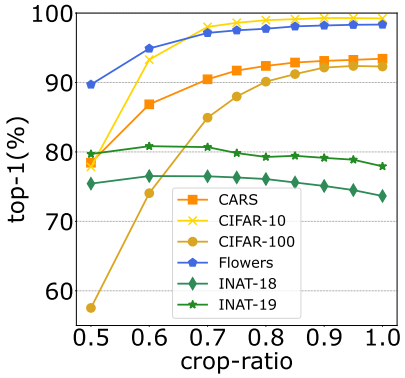

Figure 6: Transfer learning performance on 6 datasets with different test-time crop ratio. ViT-B pre-trained on ImageNet-1k at resolution 224.

Method	Learned augm. methods	# Nb of DA	Model	ImageNet-1k Val	Real	V2
Auto-Augment [7]	✓	14	ResNet50	79.7	85.6	67.9
			ViT-B	82.8	87.5	71.9
			ViT-L	84.0	**88.6**	74.0
RandAugment [6]	✓	14	ResNet50	79.5	85.5	67.6
			ViT-B	82.7	87.4	72.2
			ViT-L	84.0	88.3	73.8
Trivial-Augment [34]	✓	14	ResNet50	79.5	85.4	67.6
			ViT-B	82.3	87.0	71.2
			ViT-L	83.6	88.1	73.7
3-Augment (Ours)	✗	3	ResNet50	79.4	85.5	67.8
			ViT-B	**83.1**	**87.7**	**72.6**
			ViT-L	**84.2**	**88.6**	**74.3**

Table 3: Comparison of some existing data-augmentation methods with our simple 3-Augment proposal inspired by data-augmentation used with self-supervised learning.

This is no longer the case when evaluating on the independent set V2, for which the Auto-Augment better accuracy is not significant.

4.2.3 Impact of training resolution

In Table 6 we report the evolution of the performance according to the training resolution. We observe that we benefit from the FixRes [53] effect. By training at resolution 192×192 (or 160×160) we get a better performance at 224 after a slight fine-tuning than when training from scratch at 224×224.

We observe that the resolution has a regularization effect. While it is known that it is best to use a smaller resolution at training time [53], we also observe in the training curves that this show reduces the overfitting of the larger models. This is also illustrated by our results Table 6 with ViT-H and ViT-L. This is especially important with longer training, where models overfit without a stronger regularisation. This smaller resolution implies that there are less patches to be processed,

Model	Loss	LayerScale	Data Aug.	Epochs	ImageNet-1k val	real	v2
ViT-S	CE	✗	RandAugment	300	79.8	85.3	68.1
	BCE	✗	RandAugment	300	79.8	85.9	68.2
	BCE	✓	RandAugment	300	80.1	**86.1**	69.1
	BCE	✓	RandAugment	400	**80.7**	86.0	69.3
	BCE	✓	3-Augment	400	80.4	**86.1**	**69.7**
ViT-B	CE	✗	RandAugment	300	80.9	85.5	68.5
	BCE	✗	RandAugment	300	82.2	87.2	71.4
	BCE	✓	RandAugment	300	82.5	87.5	71.4
	BCE	✓	RandAugment	400	82.7	87.4	72.2
	BCE	✓	3-Augment	400	**83.1**	**87.7**	**72.6**
ViT-L	BCE	✗	RandAugment	300	83.0	87.9	72.4
	BCE	✗	RandAugment	400	83.3	87.7	72.5
	BCE	✓	RandAugment	400	84.0	88.3	73.8
	BCE	✓	3-Augment	400	**84.2**	**88.6**	**74.3**

Table 4: Ablation on different training component with training at resolution 224×224 on ImageNet-1k. We perform avlations with ViT-S, ViT-B and ViT-L. We report top-1 accuracy (%) on ImageNet validation set , ImageNet real and ImageNet v2.

Crop.	LS	Mixup	Aug. policy	#Imnet21k epochs	finetuning resolution	Imagenet-1k val top-1 ViT-S	ViT-B	ViT-L	Imagenet-1k v2 top-1 ViT-S	ViT-B	ViT-L
RRC	✗	0.8	RA	90	224^2	81.6	84.6	86.0	70.7	74.7	76.4
SRC	✗	0.8	RA	90	224^2	82.1	84.8	86.3	71.8	75.0	76.7
SRC	✓	0.8	RA	90	224^2	82.4	85.0	86.4	72.4	75.7	77.4
SRC	✓	✗	RA	90	224^2	82.3	85.1	86.5	72.4	75.6	77.2
SRC	✓	✗	3A	90	224^2	82.6	85.2	86.8	72.6	76.1	78.3
SRC	✓	✗	3A	240	224^2	**83.1**	**85.7**	**87.0**	**73.8**	**76.5**	**78.6**
SRC	✓	✗	3A	240	384^2	*84.8*	*86.7*	*87.7*	75.1	77.9	79.1

Table 5: Ablation path: **augmentation and regularization** with ImageNet-21k pre-training (at resolution 224×224) and ImageNet-1k fine-tuning. We measure the impact of changing Random Resize Crop (RRC) to Simple Random Crop (SRC), adding LayerScale (LS), removing Mixup, replacing RandAugment (RA) by 3-Augment (3A), and finally employing a longer number of epochs during the pre-training phase on ImageNet-21k. All experiments are done with Seed 0 with fixed hparams except the drop-path rate of stochastic depth, which depends on the model and is increased by 0.05 for the longer pre-training. We report 2 digits top-1 accuracy but note that the standard standard deviation is around 0.1 on our ViT-B baseline. Note that all these changes are neutral w.r.t. complexity except in the last row, where the fine-tuning at resolution 384×384 significantly increases the complexity.

Model	epochs		Resolution		ImageNet top-1 acc		
	Train.	FT	Train.	FT	val	real	v2
ViT-B	400	20	128×128	224×224	83.2	**88.1**	73.2
			160×160		83.3	88.0	**73.4**
			192×192		**83.5**	88.0	72.8
		-	224×224		83.1	87.7	72.6
	800	20	128×128	224×224	83.5	**88.3**	73.4
			160×160		83.6	88.2	73.5
			192×192		**83.8**	88.2	**73.6**
		-	224×224		83.7	88.1	73.1
ViT-L	400	20	128×128	224×224	83.9	**88.8**	74.3
			160×160		84.4	**88.8**	74.3
			192×192		**84.5**	**88.8**	**75.1**
		-	224×224		84.2	88.6	74.3
	800	20	128×128	224×224	84.5	**88.9**	74.7
			160×160		84.7	**88.9**	**75.2**
			192×192		**84.9**	88.7	75.1
		-	224×224		84.5	88.8	75.0
ViT-H	400	20	126×126	224×224	84.7	89.2	75.2
			154×154		**85.1**	**89.3**	75.3
			182×182		**85.1**	89.2	**75.4**
		-	224×224		84.8	89.1	75.3
	800	20	126×126	224×224	85.1	**89.2**	75.6
			154×154		**85.2**	**89.2**	**75.9**
			182×182		85.1	88.9	**75.9**
		-	224×224		84.6	88.5	74.9
ViT-H-52	400	20	126×126	224×224	84.9	89.2	75.6
ViT-H-26×2	400	20	126×126	224×224	84.9	89.1	75.3

Table 6: We compare ViT architectures pre-trained on ImageNet-1k only with different training resolution followed by a fine-tuning at resolution 224×224. We benefit from the FixRes effect [53] and get better performance with a lower training resolution (e.g resolution 160×160 with patch size 16 represent 100 tokens vs 196 for 224×224. This represents a reduction of 50% of the number of tokens).

and therefore it reduces the training cost and increases the performance. In that respect it effect is comparable to that of MAE [19]. We also report results with ViT-H 52 layers and ViT-H 26 layers parallel [50] models with 1B parameters. Due to the lower resolution training it is easier to train these models.

4.2.4 Comparison with previous training recipes for ViT

In Figure 1, we compare training procedures used to pre-train the ViT architecture either on ImageNet-1k and ImageNet-21k. Our procedure outperforms existing recipes with a large margin. For instance, with ImageNet-21k pre-training we have an improvement of $+3.0\%$ with ViT-L in comparison to the best approach. Similarly, when training from scratch on ImageNet-1k we improve the accuracy by $+2.1\%$ for ViT-H compared to the previous best approach, and by $+4.3\%$ with the best approach that does not use EMA. See also detailed results in our appendices.

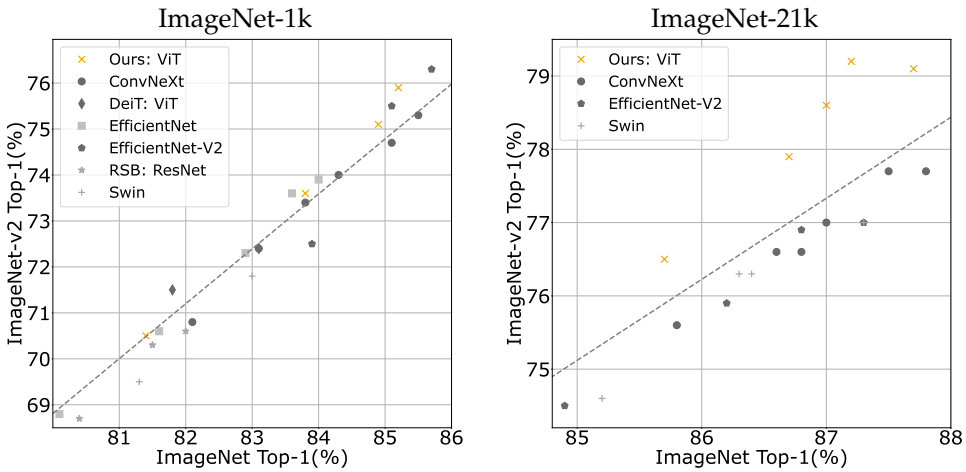

Figure 7: Generalization experiment: top-1 accuracy on ImageNet1k-val versus ImageNet-v2 for models in Table 7 and Table 8. We display a linear interpolation of all points in order to compare the generalization capability (or level of overfitting) for the different models.

4.3 Image Classification

ImageNet-1k. In Table 7 we compare ViT architectures trained with our training recipes on ImageNet-1k with other architectures. We include a comparison with the recent SwinTransformers [31] and ConvNeXts [32].

Overfitting evaluation. The comparison between Imagenet-val and -v2 is a way to quantify overfitting [54], or at least the better capability to generalize in a nearby setting without any fine-tuning[1]. In Figure 7 we plot ImageNet-val top-1 accuracy vs ImageNet-v2 top-1 accuracy in order to evaluate how the models performed when evaluated on a test set never seen at validation time. Our models overfit significantly than all other models considered, especially on ImageNet-21k. This is a good behaviour that validates the fact that our restricted choice of hyper-parameters and variants in our recipe does not lead to (too much) overfitting.

ImageNet-21k. In Table 8 we compare ViT architecture pre-trained on ImageNet-21k with our training recipe then finetuned on ImageNet-1k. We can observe that the findings are similar to what we obtained on ImageNet-1k only.

Comparison with BerT-like pre-training. In Table 9 we compare ViT models trained with our training recipes with ViT trained with different BerT-like approaches. We observe that for an equivalent number of epochs our approach gives comparable performance on ImageNet-1k and better on ImageNet-v2 as well as in segmentation on Ade. For BerT like pre-training we compare our method with MAE [19] and BeiT [2] because they remain relatively simple approaches with very good performance. As our approach does not use distillation or multi-crops we

[1]Caveat: The measures are less robust with -V2 as the number of test images is 10000 instead of 50000 for Imagenet-val. This translates to a higher standard deviation (0.2%).

Table 7: **Classification with Imagenet1k training.** We compare architectures with comparable FLOPs and number of parameters. All models are trained on ImageNet1k only without distillation nor self-supervised pre-training. We report Top-1 accuracy on the validation set of ImageNet1k and ImageNet-V2 with different measure of complexity: throughput, FLOPs, number of parameters and peak memory usage. The throughput and peak memory are measured on a single V100-32GB GPU with batch size fixed to 256 and mixed precision. For ResNet [20] and RegNet [38] we report the improved results from Wightman et al. [57]. Note that different models may have received a different optimization effort. ↑R indicates that the model is fine-tuned at the resolution R and -R indicates that the model is trained at resolution R.

Architecture	nb params ($\times 10^6$)	throughput (im/s)	FLOPs ($\times 10^9$)	Peak Mem (MB)	Top-1 Acc.	V2 Acc.
"Traditional" ConvNets						
ResNet-50 [20, 57]	25.6	2587	4.1	2182	80.4	68.7
ResNet-101 [20, 57]	44.5	1586	7.9	2269	81.5	70.3
ResNet-152 [20, 57]	60.2	1122	11.6	2359	82.0	70.6
RegNetY-4GF [38, 57]	20.6	1779	4.0	3041	81.5	70.7
RegNetY-8GF [38, 57]	39.2	1158	8.0	3939	82.2	71.1
RegNetY-16GF [38, 48]	83.6	714	16.0	5204	82.9	72.4
EfficientNet-B4 [44]	19.0	573	4.2	10006	82.9	72.3
EfficientNet-B5 [44]	30.0	268	9.9	11046	83.6	73.6
EfficientNetV2-S [45]	21.5	874	8.5	4515	83.9	74.0
EfficientNetV2-M [45]	54.1	312	25.0	7127	85.1	75.5
EfficientNetV2-L [45]	118.5	179	53.0	9540	85.7	76.3
Vision Transformers derivative						
PiT-S-224 [21]	23.5	1809	2.9	3293	80.9	_
PiT-B-224 [21]	73.8	615	12.5	7564	82.0	_
Swin-T-224 [31]	28.3	1109	4.5	3345	81.3	69.5
Swin-S-224 [31]	49.6	718	8.7	3470	83.0	71.8
Swin-B-224 [31]	87.8	532	15.4	4695	83.5	_
Swin-B-384 [31]	87.9	160	47.2	19385	84.5	_
Vision MLP & Patch-based ConvNets						
Mixer-B/16 [46]	59.9	993	12.6	1448	76.4	63.2
ResMLP-B24 [47]	116.0	1120	23.0	930	81.0	69.0
PatchConvNet-S60-224 [49]	25.2	1125	4.0	1321	82.1	71.0
PatchConvNet-B60-224 [49]	99.4	541	15.8	2790	83.5	72.6
PatchConvNet-B120-224 [49]	188.6	280	29.9	3314	84.1	73.9
ConvNeXt-B-224 [32]	88.6	563	15.4	3029	83.8	73.4
ConvNeXt-B-384 [32]	88.6	190	45.0	7851	85.1	74.7
ConvNeXt-L-224 [32]	197.8	344	34.4	4865	84.3	74.0
ConvNeXt-L-384 [32]	197.8	115	101.0	11938	85.5	75.3
Our Vanilla Vision Transformers						
ViT-S	22.0	1891	4.6	987	81.4	70.5
ViT-S↑384	22.0	424	15.5	4569	83.4	73.1
ViT-B	86.6	831	17.5	2078	83.8	73.6
ViT-B↑384	86.9	190	55.5	8956	85.0	74.8
ViT-L	304.4	277	61.6	3789	84.9	75.1
ViT-L↑384	304.8	67	191.2	12866	85.8	76.7
ViT-H	632.1	112	167.4	6984	85.2	75.9

Table 8: **Classification with Imagenet-21k training.** We compare architectures with comparable FLOPs and number of parameters. All models are trained on ImageNet-21k without distillation nor self-supervised pre-training. We report Top-1 accuracy on the validation set of ImageNet-1k and ImageNet-V2 with different measure of complexity: throughput, FLOPs, number of parameters and peak memory usage. The throughput and peak memory are measured on a single V100-32GB GPU with batch size fixed to 256 and mixed precision. For Swin-L we decrease the batch size to 128 in order to avoid out of memory error and re-estimate the memory consumption. ↑R indicates that the model is fine-tuned at the resolution R.

Architecture	nb params ($\times 10^6$)	throughput (im/s)	FLOPs ($\times 10^9$)	Peak Mem (MB)	Top-1 Acc.	V2 Acc.
"Traditional" ConvNets						
R-101x3↑384 [25]	388	_	204.6	_	84.4	_
R-152x4↑480 [25]	937	_	840.5	_	85.4	_
EfficientNetV2-S↑384 [45]	21.5	874	8.5	4515	84.9	74.5
EfficientNetV2-M↑480 [45]	54.1	312	25.0	7127	86.2	75.9
EfficientNetV2-L↑480 [45]	118.5	179	53.0	9540	86.8	76.9
EfficientNetV2-XL↑512 [45]	208.1	_	94.0	_	87.3	77.0
Patch-based ConvNets						
ConvNeXt-B [32]	88.6	563	15.4	3029	85.8	75.6
ConvNeXt-B↑384 [32]	88.6	190	45.1	7851	86.8	76.6
ConvNeXt-L [32]	197.8	344	34.4	4865	86.6	76.6
ConvNeXt-L↑384 [32]	197.8	115	101	11938	87.5	77.7
ConvNeXt-XL [32]	350.2	241	60.9	6951	87.0	77.0
ConvNeXt-XL↑384 [32]	350.2	80	179.0	16260	87.8	77.7
Vision Transformers derivative						
Swin-B [31]	87.8	532	15.4	4695	85.2	74.6
Swin-B↑384 [31]	87.9	160	47.0	19385	86.4	76.3
Swin-L [31]	196.5	337	34.5	7350	86.3	76.3
Swin-L↑384 [31]	196.7	100	103.9	33456	87.3	77.0
Vanilla Vision Transformers						
ViT-B/16 [42]	86.6	831	17.6	2078	84.0	_
ViT-B/16↑384 [42]	86.7	190	55.5	8956	85.5	_
ViT-L/16 [42]	304.4	277	61.6	3789	84.0	_
ViT-L/16↑384 [42]	304.8	67	191.1	12866	85.5	_
Our Vanilla Vision Transformers						
ViT-S	22.0	1891	4.6	987	83.1	73.8
ViT-B	86.6	831	17.6	2078	85.7	76.5
ViT-B↑384	86.9	190	55.5	8956	86.7	77.9
ViT-L	304.4	277	61.6	3789	87.0	78.6
ViT-L↑384	304.8	67	191.2	12866	87.7	79.1
ViT-H	632.1	112	167.4	6984	87.2	79.2

Pretrained data	Model	Method	# pre-training epochs	# finetuning epochs	ImageNet val	ImageNet Real	ImageNet V2
INET-1k	ViT-B	BeiT	300	$100^{(1k)}$	82.9	-	-
			800	$100^{(1k)}$	83.2	-	-
		MAE*	1600	$100^{(1k)}$	<u>83.6</u>	<u>88.1</u>	<u>73.2</u>
		Ours	$400^{(1k)}$	$20^{(1k)}$	83.5	88.0	72.8
			$800^{(1k)}$	$20^{(1k)}$	**83.8**	**88.2**	**73.6**
	ViT-L	BeiT	800	$30^{(1k)}$	<u>85.2</u>	-	-
		MAE	400	$50^{(1k)}$	84.3	-	-
			800	$50^{(1k)}$	84.9	-	-
			1600	$50^{(1k)}$	85.1	-	-
		MAE*	1600	$50^{(1k)}$	**85.9**	**89.4**	**76.5**
		Ours	$400^{(1k)}$	$20^{(1k)}$	84.5	<u>88.8</u>	<u>75.1</u>
			$800^{(1k)}$	$20^{(1k)}$	84.9	88.7	<u>75.1</u>
INET-21k	ViT-B	BeiT	150	$50^{(1k)}$	83.7	88.2	73.1
			150 + $90^{(21k)}$	$50^{(1k)}$	<u>85.2</u>	<u>89.4</u>	75.4
		Ours	$90^{(21k)}$	$50^{(1k)}$	<u>85.2</u>	<u>89.4</u>	<u>76.1</u>
			$240^{(21k)}$	$50^{(1k)}$	**85.7**	**89.5**	**76.5**
	ViT-L	BeiT	150	$50^{(1k)}$	86.0	89.6	76.7
			150 + $90^{(21k)}$	$50^{(1k)}$	**87.5**	**90.1**	**78.8**
		Ours	$90^{(21k)}$	$50^{(1k)}$	86.8	89.9	78.3
			$240^{(21k)}$	$50^{(1k)}$	<u>87.0</u>	<u>90.0</u>	<u>78.6</u>

Table 9: Comparison of self-supervised pre-training with our approach. As our approach is fully supervised, this table is given as an indication. All models are evaluated at resolution 224×224. We report Image classification results on ImageNet val, real and v2 in order to evaluate overfitting. $^{(21k)}$ indicate a finetuning with labels on ImageNet-21k and $^{(1k)}$ indicate a finetuning with labels on ImageNet-1k. * design the improved setting of MAE using pixel (w/ norm) loss.

have not made a comparison with approaches such as PeCo [12] which use an auxiliary model as a psycho-visual loss and iBoT [66], which uses multi-crop and an exponential moving average of the model.

4.4 Downstream tasks and other architectures

4.4.1 Transfer Learning

In order to evaluate the quality of the ViT models learned through our training procedure we evaluated them with transfer learning tasks. We focus on the performance of ViT models pre-trained on ImageNet-1k only at resolution 224×224 during 400 epochs on the 6 datasets shown in Table 14. Our results are presented in Table 10. In Figure 6 we measure the impact of the crop ratio at inference time on transfer learning results. We observe that on iNaturalist this parameter has a significant impact on the performance. As recommended in the paper Three Things [50] we finetune only the attention layers for transfer learning experiments on Flowers, this improves performance by 0.2%.

Table 10: We compare Transformers based models on different transfer learning tasks with ImageNet-1k pre-training. We report results with our default training on ImageNet-1k (400 epochs at resolution 224×224). We also report results with convolutional architectures for reference. For consistency we keep our crop ratio equal to 1.0 on all datasets. Other works use 0.875, which is better for iNat-19 and iNat-18, see Figure 6.

Model	CIFAR-10	CIFAR-100	Flowers	Cars	iNat-18	iNat-19
Grafit ResNet-50 [52]	-	-	98.2	92.5	69.8	75.9
ResNet-152 [4]	-	-	-	-	69.1	-
ViT-B/16 [13]	98.1	87.1	89.5	-	-	-
ViT-L/16 [13]	97.9	86.4	89.7	-	-	-
ViT-B/16 [42]	-	87.8	96.0	-	-	-
ViT-L/16 [42]	-	86.2	91.4	-	-	-
DeiT-B	99.1	90.8	98.4	92.1	73.2	77.7
Ours ViT-S	98.9	90.6	96.4	89.9	67.1	72.7
Ours ViT-B	99.3	92.5	98.6	93.4	73.6	78.0
Ours ViT-L	**99.3**	**93.4**	**98.9**	**94.5**	**75.6**	**79.3**

4.4.2 Semantic segmentation

We evaluate our ViT baselines models (400 epochs schedules for ImageNet-1k models and 90 epochs for ImageNet-21k models) with semantic segmentation experiments on ADE20k dataset [65]. This dataset consists of 20k training and 5k validation images with labels over 150 categories. For the training, we adopt the same schedule as in Swin: 160k iterations with UperNet [59]. At test time we evaluate with a single scale and multi-scale. Our UperNet implementation is based on the XCiT [16] repository. By default the UperNet head uses an embedding dimension of 512. In order to save compute, for small and tiny models we set it to the size of their working dimension, i.e. 384 for small and 192 for tiny. We keep the 512 by default as it is done in XCiT for other models. Our results are reported in Table 11. We observe that vanilla ViTs trained with our training recipes have a better FLOPs-accuracy trade-off than recent architectures like XCiT or Swin.

4.4.3 Training with others architectures

In Table 12 we measure the top-1 accuracy on ImageNet-val, ImageNet-real and ImageNet-v2 with different architecture train with our training procedure at resolution 224×224 on ImageNet-1k only. We can observe that for some architectures like PiT or CaiT our training method will improve the performance. For some others like TNT our approach is neutral and for architectures like Swin it decreases the performance. This is consistent with the findings of Wightman et al. [57] and illustrates the need to improve the training procedure in conjunction to the architecture to obtain robust conclusions. Indeed, adjusting these architectures while keeping the training procedure fixed can probably have the same effect as keeping the architecture fixed and adjusting the training procedure. That means that with a fixed training procedure we can have an overfitting of an architecture for a given training procedure. In order to take overfitting into account we perform our measurements on the ImageNet val and ImageNet-v2 to quantify the amount of overfitting.

Table 11: **ADE20k semantic segmentation** performance using UperNet [59] (in comparable settings [11, 16, 31]). All models are pre-trained on ImageNet-1k except models with † symbol that are pre-trained on ImageNet-21k. We report the pre-training resolution used on ImageNet-1k and ImageNet-21k.

Backbone	Pre-training resolution	UperNet #params ($\times 10^6$)	UperNet FLOPs ($\times 10^9$)	UperNet Single scale mIoU	UperNet Multi-scale mIoU
ResNet50	224×224	66.5	-	42.0	-
DeiT-S	224×224	52.0	1099	-	44.0
XciT-T12/16	224×224	34.2	874	41.5	-
XciT-T12/8	224×224	33.9	942	43.5	-
Swin-T	224×224	59.9	945	44.5	46.1
Our ViT-T	224×224	10.9	148	40.1	41.8
Our ViT-S	224×224	41.7	588	**45.6**	**46.8**
XciT-M24/16	224×224	112.2	1213	47.6	-
XciT-M24/8	224×224	110.0	2161	48.4	-
PatchConvNet-B60	224×224	140.6	1258	48.1	48.6
PatchConvNet-B120	224×224	229.8	1550	49.4	50.3
MAE ViT-B	224×224	127.7	1283	48.1	-
Swin-B	384×384	121.0	1188	48.1	49.7
Our ViT-B	224×224	127.7	1283	49.3	50.2
Our ViT-L	224×224	353.6	2231	**51.5**	**52.0**
PatchConvNet–B60†	224×224	140.6	1258	50.5	51.1
PatchConvNet-L120†	224×224	383.7	2086	52.2	52.9
Swin-B† (640×640)	224×224	121.0	1841	50.0	51.6
Swin-L† (640×640)	224×224	234.0	3230	-	53.5
Our ViT-B†	224×224	127.7	1283	51.8	52.8
Our ViT-B†	384×384	127.7	1283	53.4	54.1
Our ViT-L†	224×224	353.6	2231	53.8	54.7
Our ViT-L†	320×320	353.6	2231	**54.6**	**55.6**

5 Conclusion

This paper makes a simple contribution: it proposes improved baselines for vision transformers trained in a supervised fashion that can serve (1) either as a comparison basis for new architectures; (2) or for other training approaches such as those based on self-supervised learning. We hope that this stronger baseline will serve the community effort in making progress on learning foundation models that could serve many tasks. Our experiments have also gathered a few insights on how to train ViT for larger models with reduced resources without hurting accuracy, allowing us to train a one-billion parameter model with 4 nodes of 8 GPUs.

Acknowledgement. We thank Ishan Misra for his valuable feedback.

Model	Params ($\times 10^6$)	Flops ($\times 10^9$)	ImageNet-1k orig.	val	real	v2
ViT-S [48]	22.0	4.6	79.8	80.4	86.1	69.7
ViT-B [13, 48]	86.6	17.6	81.8	83.1	87.7	72.6
PiT-S [21]	23.5	2.9	80.9	80.4	86.1	69.2
PiT-B [21]	73.8	12.5	82.0	82.4	86.8	72.0
TNT-S [62]	23.8	5.2	81.5	81.4	87.2	70.6
TNT-B [62]	65.6	14.1	82.9	82.9	87.6	72.2
ConViT-S [8]	27.8	5.8	81.3	81.3	87.0	70.3
ConViT-B [8]	86.5	17.5	82.4	82.0	86.7	71.3
Swin-S [31]	49.6	8.7	83.0	82.1	86.9	70.7
Swin-B [31]	87.8	15.4	83.5	82.2	86.7	70.7
CaiT-B12 [51]	100.0	18.2	-	83.3	87.7	73.3

Table 12: We report the performance reached with our training recipe with 400 epochs at resolution 224×224 for other transformers architectures. We have not performed an extensive grid search to adapt the hyper-parameters to each architecture. Our results are overall similar to the ones achieved in the papers where these architectures were originally published (reported in column 'orig.'), except for Swin Transformers, for which we observe a drop on ImageNet-val.

References

[1] Apex. https://nvidia.github.io/apex/index.html, accessed: 2022-01-01

[2] Bao, H., Dong, L., Wei, F.: Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254 (2021)

[3] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision (2020)

[4] Chu, P., Bian, X., Liu, S., Ling, H.: Feature space augmentation for long-tailed data. arXiv preprint arXiv:2008.03673 (2020)

[5] Cordonnier, J.B., Loukas, A., Jaggi, M.: On the relationship between self-attention and convolutional layers. arXiv preprint arXiv:1911.03584 (2019)

[6] Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: RandAugment: Practical automated data augmentation with a reduced search space. arXiv preprint arXiv:1909.13719 (2019)

[7] Cubuk, E.D., Zoph, B., Mané, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation policies from data. arXiv preprint arXiv:1805.09501 (2018)

[8] d'Ascoli, S., Touvron, H., Leavitt, M.L., Morcos, A.S., Biroli, G., Sagun, L.: Convit: Improving vision transformers with soft convolutional inductive biases. In: ICML (2021)

[9] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)

[10] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)

[11] Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., Guo, B.: Cswin transformer: A general vision transformer backbone with cross-shaped windows. arXiv preprint arXiv:2107.00652 (2021)

[12] Dong, X., Bao, J., Zhang, T., Chen, D., Zhang, W., Yuan, L., Chen, D., Wen, F., Yu, N.: Peco: Perceptual codebook for bert pre-training of vision transformers. arXiv preprint arXiv:2111.12710 (2021)

[13] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)

[14] El-Nouby, A., Izacard, G., Touvron, H., Laptev, I., Jegou, H., Grave, E.: Are large-scale datasets necessary for self-supervised pre-training? arXiv preprint arXiv:2112.10740 (2021)

[15] El-Nouby, A., Neverova, N., Laptev, I., Jégou, H.: Training vision transformers for image retrieval. arXiv preprint arXiv:2102.05644 (2021)

[16] El-Nouby, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., et al.: Xcit: Cross-covariance image transformers. arXiv preprint arXiv:2106.09681 (2021)

[17] Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., Feichtenhofer, C.: Multiscale vision transformers. arXiv preprint arXiv:2104.11227 (2021)

[18] Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jégou, H., Douze, M.: Levit: a vision transformer in convnet's clothing for faster inference. arXiv preprint arXiv:2104.01136 (2021)

[19] He, K., Chen, X., Xie, S., Li, Y., Doll'ar, P., Girshick, R.B.: Masked autoencoders are scalable vision learners. arXiv preprint arXiv:2111.06377 (2021)

[20] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Conference on Computer Vision and Pattern Recognition (2016)

[21] Heo, B., Yun, S., Han, D., Chun, S., Choe, J., Oh, S.J.: Rethinking spatial dimensions of vision transformers. arXiv preprint arXiv:2103.16302 (2021)

[22] Horn, G.V., Mac Aodha, O., Song, Y., Shepard, A., Adam, H., Perona, P., Belongie, S.J.: The iNaturalist species classification and detection dataset. arXiv preprint arXiv:1707.06642 (2017)

[23] Horn, G.V., Mac Aodha, O., Song, Y., Shepard, A., Adam, H., Perona, P., Belongie, S.J.: The inaturalist challenge 2018 dataset. arXiv preprint arXiv:1707.06642 (2018)

[24] Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.Q.: Deep networks with stochastic depth. In: European Conference on Computer Vision (2016)

[25] Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., Houlsby, N.: Big transfer (bit): General visual representation learning. arXiv preprint arXiv:1912.11370 **6**, 3 (2019)

[26] Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: IEEE Workshop on 3D Representation and Recognition (2013)

[27] Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. In: NeurIPS (2012)

[28] Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep., CIFAR (2009)

[29] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)

[30] LingChen, T.C., Khonsari, A., Lashkari, A., Nazari, M.R., Sambee, J.S., Nascimento, M.A.: Uniformaugment: A search-free probabilistic data augmentation approach. arXiv preprint arXiv:2003.14348 (2020)

[31] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021)

[32] Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. arXiv preprint arXiv:2201.03545 (2022)

[33] Loshchilov, I., Hutter, F.: Fixing weight decay regularization in adam. arXiv preprint arXiv:1711.05101 (2017)

[34] Müller, S., Hutter, F.: Trivialaugment: Tuning-free yet state-of-the-art data augmentation. arXiv preprint arXiv:2103.10158 (2021)

[35] Neimark, D., Bar, O., Zohar, M., Asselmann, D.: Video transformer network. arXiv preprint arXiv:2102.00719 (2021)

[36] Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing (2008)

[37] Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: Conference on Computer Vision and Pattern Recognition (2014)

[38] Radosavovic, I., Kosaraju, R.P., Girshick, R.B., He, K., Dollár, P.: Designing network design spaces. Conference on Computer Vision and Pattern Recognition (2020)

[39] Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do ImageNet classifiers generalize to ImageNet? In: International Conference on Machine Learning (2019)

[40] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015)

[41] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)

[42] Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., Beyer, L.: How to train your vit? data, augmentation, and regularization in vision transformers. arXiv preprint arXiv:2106.10270 (2021)

[43] Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Conference on Computer Vision and Pattern Recognition (2015)

[44] Tan, M., Le, Q.V.: EfficientNet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946 (2019)

[45] Tan, M., Le, Q.V.: Efficientnetv2: Smaller models and faster training. In: International Conference on Machine Learning (2021)

[46] Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Keysers, D., Uszkoreit, J., Lucic, M., Dosovitskiy, A.: MLP-Mixer: An all-MLP architecture for vision. arXiv preprint arXiv:2105.01601 (2021)

[47] Touvron, H., Bojanowski, P., Caron, M., Cord, M., El-Nouby, A., Grave, E., Joulin, A., Synnaeve, G., Verbeek, J., Jégou, H.: ResMLP: feedforward networks for image classification with data-efficient training. arXiv preprint arXiv:2105.03404 (2021)

[48] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. International Conference on Machine Learning (2021)

[49] Touvron, H., Cord, M., El-Nouby, A., Bojanowski, P., Joulin, A., Synnaeve, G., Jégou, H.: Augmenting convolutional networks with attention-based aggregation. arXiv preprint arXiv:2112.13692 (2021)

[50] Touvron, H., Cord, M., El-Nouby, A., Verbeek, J., J'egou, H.: Three things everyone should know about vision transformers. arXiv preprint arXiv:2203.09795 (2022)

[51] Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H.: Going deeper with image transformers. International Conference on Computer Vision (2021)

[52] Touvron, H., Sablayrolles, A., Douze, M., Cord, M., Jégou, H.: Grafit: Learning fine-grained image representations with coarse labels. International Conference on Computer Vision (2021)

[53] Touvron, H., Vedaldi, A., Douze, M., Jegou, H.: Fixing the train-test resolution discrepancy. Neurips (2019)

[54] Touvron, H., Vedaldi, A., Douze, M., Jégou, H.: Fixing the train-test resolution discrepancy: Fixefficientnet. arXiv preprint arXiv:2003.08237 (2020)

[55] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)

[56] Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. arXiv preprint arXiv:2102.12122 (2021)

[57] Wightman, R., Touvron, H., Jégou, H.: Resnet strikes back: An improved training procedure in timm. arXiv preprint arXiv:2110.00476 (2021)

[58] Wu, H., Xiao, B., Codella, N.C.F., Liu, M., Dai, X., Yuan, L., Zhang, L.: Cvt: Introducing convolutions to vision transformers. arXiv preprint arXiv:2103.15808 (2021)

[59] Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: European Conference on Computer Vision (2018)

[60] Xiao, T., Singh, M., Mintun, E., Darrell, T., Dollár, P., Girshick, R.: Early convolutions help transformers see better. arXiv preprint arXiv:2106.14881 (2021)

[61] You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., Song, X., Demmel, J., Keutzer, K., Hsieh, C.J.: Large batch optimization for deep learning: Training BERT in 76 minutes. In: International Conference on Learning Representations (2020)

[62] Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Tay, F., Feng, J., Yan, S.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. arXiv preprint arXiv:2101.11986 (2021)

[63] Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: CutMix: Regularization strategy to train strong classifiers with localizable features. arXiv preprint arXiv:1905.04899 (2019)

[64] Zhang, H., Cissé, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)

[65] Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. Conference on Computer Vision and Pattern Recognition (2017)

[66] Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A.L., Kong, T.: ibot: Image bert pre-training with online tokenizer. arXiv preprint arXiv:2111.07832 (2021)

Appendices

A Experimental details

Fine-tuning at higher resolution When pre-training on ImageNet-1k at resolution 224×224 we fix the train-test resolution discrepancy by finetuning at a higher resolution [53]. Our finetuning procedure is inspired by DeiT, except that we adapt the stochastic depth rate according to the model size [51]. We fix the learning reate to $lr = 1 \times 10^{-5}$ with batch-size=512 during 20 epochs with a weight decay of 0.1 without repeated augmentation. Other hyper-parameters are similar to those employed in DeiT fine-tuning.

Stochastic depth We adapt the stochastic depth drop rate according to the model size. We report stochastic depth drop rate values in Table 13.

Model	# Params ($\times 10^6$)	FLOPs ($\times 10^9$)	Stochastic depth drop-rate	
			ImageNet-1k	ImageNet-21k
ViT-T	5.7	1.3	0.0	0.0
ViT-S	22.0	4.6	0.0	0.0
ViT-B	86.6	17.5	0.1	0.1
ViT-L	304.4	61.6	0.4	0.3
ViT-H	632.1	167.4	0.5	0.5

Table 13: Stochastic depth drop-rate according to the model size. For 400 epochs training on ImageNet-1k and 90 epochs training on ImageNet-21k. See section B for further adaption with longer training.

For transfer learning experiments we evaluate our models pre-trained at resolution 224×224 on ImageNet-1k only on 6 transfer learning datasets. We give the details of these datasets in Table 14 below.

Dataset	Train size	Test size	#classes
iNaturalist 2018 [23]	437,513	24,426	8,142
iNaturalist 2019 [22]	265,240	3,003	1,010
Flowers-102 [36]	2,040	6,149	102
Stanford Cars [26]	8,144	8,041	196
CIFAR-100 [28]	50,000	10,000	100
CIFAR-10 [28]	50,000	10,000	10

Table 14: Datasets used for our different transfer-learning tasks.

B Additional Ablations

Number of training epochs In Table 15 we provide an ablation on the number of training epochs on ImageNet-1k. We do not observe a saturation when the increase

Model	epochs	ImageNet top1 acc.		
		val	real	v2
ViT-S	300	79.9	86.1	68.8
	400	80.4	86.1	69.7
	600	80.8	86.7	69.9
	800	81.4	87.0	70.5
ViT-B	300	82.8	87.6	72.1
	400	83.1	87.7	72.6
	600	83.2	87.8	73.3
	800	83.7	88.1	73.1
ViT-L	300	84.1	88.5	74.1
	400	84.2	88.6	74.3
	600	84.4	88.6	74.6
	800	84.5	88.8	75.0
ViT-H	300	84.6	89.0	74.9
	400	84.8	89.1	75.3

Table 15: Impact on the performance of the number of training epochs on ImageNet-1k.

of the number of training epochs, as observed with BerT like approaches [2, 19]. For longer training we increase the weight decay from 0.02 to 0.05 and we increase the stochastic depth drop-rate by 0.05 every 200 epochs to prevent overfitting.