A Comprehensive Study of Transfer Learning under Constraints

Tom Pégeot
Université Paris-Saclay, CEA, LIST
F-91120 Palaiseau, France
tom.pegeot@cea.fr

Inna Kucher
Université Paris-Saclay, CEA, LIST
F-91120 Palaiseau, France
inna.kucher@cea.fr

Adrian Popescu
Université Paris-Saclay, CEA, LIST
F-91120 Palaiseau, France
adrian.popescu@cea.fr

Bertrand Delezoide
Amanda
F-75008, Paris, France
bertrand.delezoide@amanda.com

Abstract

Pre-training on an upstream task is widely used in deep learning to boost performance of downstream tasks. Recent studies analyzed pre-training with large datasets and large deep neural network architectures. However, pre-training is very useful in practice when downstream tasks have scarce data and are trained under computational constraints. To assess pre-training performance in this setting, we train different deep architectures with 1M parameters. We create different subsets of ImageNet to study the influence of upstream dataset in detail by varying the total size, but also the ratio between number of classes and samples per class for a constant total size. Then, we use the resulting models in transfer toward six diversified downstream tasks using linear probing and full fine tuning for downstream training. Experimental results confirm previous ones regarding performance saturation in downstream tasks, but we find that saturation occurs faster for compact deep architectures. The use of different ImageNet subsets leads to globally similar performance when enough data is included, regardless of the dataset structure. The comparison of downstream training strategies shows that linear probing can be competitive, particularly for few-shot settings. This is at odds with previous reports, which assert the superiority of full fine tuning. Finally, we observe that the type of deep architecture has a significant effect on results, but that their relative performance varies depending on the downstream training strategy.

1. Introduction

Deep neural networks are known to be data hungry [22], particularly when they include a large number of parameters. Transfer learning alleviates this problem by pre-training an upstream dataset to improve performance in downstream task, or accelerate the training process [29, 7]. Pre-training is also useful when the target domain data are not sufficient to learn an effective model from scratch, and the gain obtained from the upstream model is larger than the loss of representativeness due to domain shift [25]. The importance of pre-training grew with the advent of deep neural networks, whose learned representations are transferable [29]. Recent studies of transfer learning [4, 5, 8, 15, 39] focused on pre-training models with increasingly large number of parameters and amounts of data. They conclude that increasing the size of models and of data improves the performance in target tasks, at least until saturation is reached [1].

While interesting, these studies disregard the fact that transfer learning is often useful when training and inference capacity are limited [13]. In this work, we investigate transferability under constraints by analyzing the effects of core factors which drive this process. During pre-training, we notably test the influence of: (1) pre-training for compact deep architectures, which are likely to be used in transfer learning for constrained environments [35]; (2) deep neural network architectures since they are known to influence both the upstream and downstream performance [14]; (3) the amount of available training data, as well the ratio between number of classes and samples per class for a fixed-size upstream dataset, since downstream accuracy saturation was already analyzed for large deep architectures [1, 6, 16], but not for compact ones. During inference, we analyze the influence of: (1) the type of downstream training strategy, with the deployment of linear probing and full fine tuning, since the depth of the fine tuning process leads determines the degree to which features are adapted to the downstream task or preserved from the upstream model [39]; (2) the number of images per class in

the target datasets to assess transferability in four few-shot settings [5] and full dataset availability scenario since they are all important in practice.

We run experiments using different subsets of ImageNet [3] as upstream dataset, four deep architectures, and with six downstream datasets designed for diverse visual tasks. The empirical study reported here concludes that:

- Downstream performance saturation is reached much faster with the compact deep architectures compared to the large architectures analyzed in previous studies [1, 16]. This finding indicates that very large pre-training datasets are not needed to obtain good downstream performance with compact deep architectures.
- The structure of the upstream dataset (number of classes, samples per class) has a small influence on downstream accuracy once there are enough data in it.
- The type of architecture makes a difference, particularly when linear probing is used for downstream training. In this setting, architectures with higher-dimensional output features are clearly a better choice.
- The performance of the full fine tuning and of linear probing depends on the downstream configuration. The latter strategy is competitive when the domain shift between upstream and downstream tasks is small and/or in case of low-shot settings. Since linear probing training is much simpler, it should be considered for deployment in these cases.

As a whole, the reported results give a comprehensive view of transfer learning under constraints. They provide a sound baseline for future work performed by both researchers and engineers.

2. Related Work

Transfer learning is important for practical applications of deep learning, and is the subject of a large number of existing studies. We discuss the most relevant studies for transfer learning under constraints, which is in focus here. Prior works are further put into perspective when analyzing the results of the different experiments.

Past examination of pre-training tend to show that increasing the size of the upstream dataset has a positive effect on downstream accuracy [20, 32, 38]. However, recent studies, such as [1, 6], find that the improvement tends to saturate, and this phenomenon occurs faster for self-supervised pre-training. The works cited above focus on the total size of the dataset in terms of samples, and they give less importance to the structure of the dataset in terms of the number of classes and of samples per class. The importance of the dataset structure was highlighted for domain adaptive transfer learning [24]. The authors of this study conclude that adding more data, including more classes, can have a deleterious effect on downstream performance. In this study, the pre-training has the prior knowledge about the target task. In contrast, we pre-train models without any assumption regarding the content of downstream tasks in order to avoid meta-overfitting [39].

The strategy used for downstream training has a strong influence on performance. Past studies [16, 39] tested pre-training for full downstream datasets, but also in few-shot learning settings. They showed that full fine tuning of downstream models is better than linear probing, which consists in retraining only the final fully-connected layer of the model. This finding seems intuitive since fine tuning adapts the features of downstream models to the characteristics of the downstream tasks. A nuance was brought by [18], a study which shows that linear probing is actually better than fine tuning when testing with out-of-distribution data for downstream tasks. However, past results were reported for the pre-training with large deep models. It is interesting to study whether they hold for smaller models, which are in focus here. Importantly, we run a more systematic study of few-shot settings compared to [16, 39] in order to have a fine-grained analysis of the merits and limitations of the two strategies. We note that there exist more refined transfer strategies. Image-level adaptation of the strategy is proposed in [10], adaptive fine tuning is explored in [9], while a combination of features from different layers is used in [7]. While interesting, they are out of the immediate scope of this work, which focuses on two opposite strategies.

Previous works focused on transfer learning for computationally-constrained devices showed the benefits of freezing part of the networks [33, 34]. However, they focused on hardware optimization [33] in order to reduce the overall energetic footprint of the implemented deep models, or architecture quantization [34] to reduce their parametric footprint. Here, we take a complementary approach and pay more attention to the upstream and downstream data, and use network scaling to preserve the precision of downstream representations.

3. Study Setup

3.1. Datasets

Pre-training datasets. Following the common practice [7, 16, 39], we transfer data from a single upstream dataset to all downstream tasks in order to assess the generalization capacity of the upstream model. The authors of [39] underline the importance of mitigating meta-overfitting when transferring knowledge. They advise to create the upstream model independently of any knowledge about downstream data. Therefore, we generate different versions of pre-training datasets by sampling Ima-

geNet21k [3]. The classes included in these datasets are selected randomly from the set of leaves classes that have enough samples per classes. A first series of tests use a variable number of classes from 100 to 6000 and fixes the number of 500 samples per class. These subsets are used to assess if downstream performance continues to increase or saturates when adding new classes. A second series of tests simultaneously vary the number of classes and samples to keep the total number of samples in the dataset constant. The size of the dataset is 1M images and the number of classes varies from 1000 to 6000. This experiment could not be carried out with fewer classes since ImageNet does not contain enough richly-represented leaf classes to reach the target dataset size. This setting corresponds to an upstream training on a fixed budget. These subsets are used to assess whether class diversity or individual class representations are more important. Note that there is no assumption made regarding the similarity between the upstream dataset and the downstream ones. This is important in order to simulate a situation in which pre-training is done without knowledge of the downstream tasks, and thus ensure the generalization of the proposed transfer scheme.

Downstream datasets. A thorough evaluation of the usefulness of pre-training requires the use of multiple and diversified downstream datasets [1]. We follow this observation and transfer upstream models toward six downstream tasks: Oxford-IIIT Pet [26] is designed for pet race recognition, Describable Textures Dataset (DTD) [2] provides different types of textures as perceived by humans, GTSRB [31], Street View House Numbers (SVHN) [23] includes house number images, FGVC-Aircraft (FGVC) [21] is designed for aircraft model recognition and Cifar100 [17] includes commonsense-level classes [27]. These datasets cover a wide range of visual tasks, and the conclusions drawn from a study of pre-training involving all of them are robust. Their main statistics are presented in Table 1. Images are resized to match the input size used during pre-training which is 224x224. Standard data augmentation [17, 11] which includes random cropping and random horizontal flipping is applied for four datasets out of six. Horizontal flipping is deactivated for GTSRB and SVHN because they mainly represent classes that depends on the orientation.

3.2. Downstream data availability

It is important to study the influence of the amount of data available for downstream tasks since pre-training is most needed when downstream data are scarce. We first run experiments with the full datasets, and then test with four few-shot learning regimes. For this we limit the number of samples per class in the downstream task to 1, 5, 10 and 25. This is a finer-grained investigation of the influence of data availability compared to [39], where a single few-shot learning setting were used. To mitigate data selection bias, we follow a standard procedure in few-shot learning and sample training images five times for each regime.

3.3. Downstream training strategies.

Following [39, 18], we run experiments with fine tuning and linear probing, two opposite strategies. Fine tuning retrains all the layers of downstream models, while linear probing only retrains the final layer. Fine tuning is usually preferred [16, 18, 39] since the full retraining of the downstream model adapts it to the domain of the downstream task. Linear probing [29] is less adaptive since it exploits pre-trained features as such. The latter can be interesting if the computational capacity of the device is limited [33] and/or when the amount of available training data is insufficient to learn a full model in an efficient manner. We note the existence of the other downstream training strategies [9, 10, 7], but their usage is out of the immediate scope here.

3.4. Training details

Training parametrization is done using a procedure which is inspired by the lightweight sweep mode proposed in [39]. Fixed values are used for most hyperparameters across network architectures and tasks. While not fully optimized for each task, this mode allows a fair comparison in a constrained environment.

The resolution of the input images used for training was classically set to 224x224 [11].

Upstream training. To make the results comparable we used the same hyper-parameters for each training. All of them were trained during 110 epochs using the "1cycle" learning rate scheduler [30]. We chose this scheduler because it allows fast training [30], which was important in order to reduce the time needed to pre-train all the networks on all the different subsets of ImageNet21K. The batch-size is set to 128, and the maximum learning rate for the OneCycleLR is set to 0.005. Also even if recent works demonstrate that increasing the weight-decay on the head of the network lead to better downstream performances [38], we use a constant weight-decay of 5e-4 for all the layers to avoid any side effects.

Downstream training for full dataset. Two common training strategies for transfer are considered here. Linear probing is deployed because it is well adapted for constrained environments [33]. We use a fully-connected layer for classification, which receives the features provided by the upstream model. This final layer is trained for 100 epochs, with a ReduceLrOnPlateau[1] learning rate scheduler based on the loss metrics with a patience of 5. This allow us to stop the training if the learning rate reaches 10^{-8}. The

[1] https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html

Dataset	Oxford-IIIT Pet [26]	DTD [2]	GTSRB [31]	SVHN [23]	FGVC [21]	Cifar100 [17]
# classes	37	47	43	10	100	100
# training/class	99.432	39.979	619.512	7325.7	33.34	500
stdev training/class	1.534	0.144	457.377	2800.661	0.474	0
# test/class	99.135	39.979	293.721	2603.2	33.33	100

Table 1: Downstream datasets statistics.

initial learning rate is set to 0.01. The weight decay is set to a constant 5e-4 over the whole network for the same reason as the upstream training.

Full fine tuning adapts all the layers of the architecture during downstream training, and past studies indicate that it outperforms linear probing, even in few-shot learning scenarios [16, 39]. While it requires more computational power than linear probing, it can be implemented on edge devices after optimization [33]. During this training we used the same parameters as for linear probing except for the initial learning rate which is set to 0.001 to avoid damaging the pre-trained features in the first steps.

Downstream training in a few-shot setting. A recent work pointed out that training for a large number of epochs can be beneficial if downstream datasets are small [6]. However, overfitting sometimes occurs if this process is run until its end. To accommodate these two observations, we fine-tune for a large number of epochs (2500 for single shot), but stop the process if the learning rate value is too low (10^{-7}). Following [6], when increasing the number of samples per class we divide the number of epochs by the number of samples per class to keep the same number of updates during the different training. The learning rate is again reduced on plateau.

3.5. Deep Network Architectures

We choose MobileNetv2 [28], ShuffleNetv2 [19], Resnet18 [11] architectures for our experiments. To make the results obtained with these architectures comparable they are all downscaled to reach a size of 1M parameters. When scaling strategies are presented in the original papers, as it is the case for MobileNetv2 [28] and ShuffleNetv2 [19], we follow them here. We use a similar strategy for ResNet18, and also create a second version of MobileNetv2 to study the effect of embedding sizes.

MobileNetv2. We selected MobileNetv2 [28] because it was designed for computationally constrained environments. We test two version of the model scaling. The first version, MobileNet_{868}, is scaled using the scaling method from the original paper [28, 12] with a width multiplier of 0.678. The number of channels in the output of the inverted residual blocks are 11, 16, 22, 43, 65, 108 and 217 and the size of the vector in output of the feature extractor is 868. The second version, MobileNet_{151}, is downscaled by fixing the embedding size of the extractor to 151 and before using the strategy from [28] to adapt the rest of the network. We created MobileNet_{151} to match the embedding size of ResNet_{151}, and also test the influence of the embedding size against MobileNet_{868}.

ShuffleNetv2. We used the scaling method proposed in the original paper [19], to downsize this architecture to 1M parameters. The output channels of each stage are multiplied by a subunit factor (0.866), while leaving the first and the last convolution unchanged. Since the output of the extractor is 1024, we will refer to the downsized architecture as ShuffleNet_{1024}.

ResNet18. ResNet18 [11] is a generic architecture which is often used in literature. It has over 11M parameters in its full version and we downscale it to reach 1M parameters. The number of channels in each residual block is reduced uniformly, using a 0.295 width multiplication factor.

4. Experiments

4.1. Effect of a larger pre-training dataset

Past studies of pre-training [1, 8, 20] showed that larger upstream datasets translated into higher downstream performance. However, it was noted that saturation occurs beyond a certain point, and adding supplementary data is not useful anymore [1]. Given that past studies were focused on large deep neural networks, it is interesting to analyze the behavior of smaller models with respect to the number of upstream classes. We keep the number of images per class constant at 500, regardless of the total number of classes included in the pre-training dataset.

We present the results obtained with different architectures in Figures 1a and 1b with a linear probing and full fine tuning of downstream tasks, respectively. Performance increases a lot when the total number of classes used for pre-training is small. An important gain is observed between 100 and 500 classes, particularly for linear probing. The relative gain starts to decline between 500 and 1000 classes, and even more between 1000 and 2000 classes. Then, performance starts to saturate beyond 2000 classes. Some performance variability is observed in the 1000 to 6000 classes range for all tested architectures and both training strategies when increasing the number of classes, but they do not exceed 3 accuracy points between the lowest and highest points. This finding is important insofar it indicates that increasing the number of classes is not useful for deep architectures designed for constrained environments. Performance saturation occurs for much smaller volumes of data compared to previous studies [1, 8, 20], which fo-

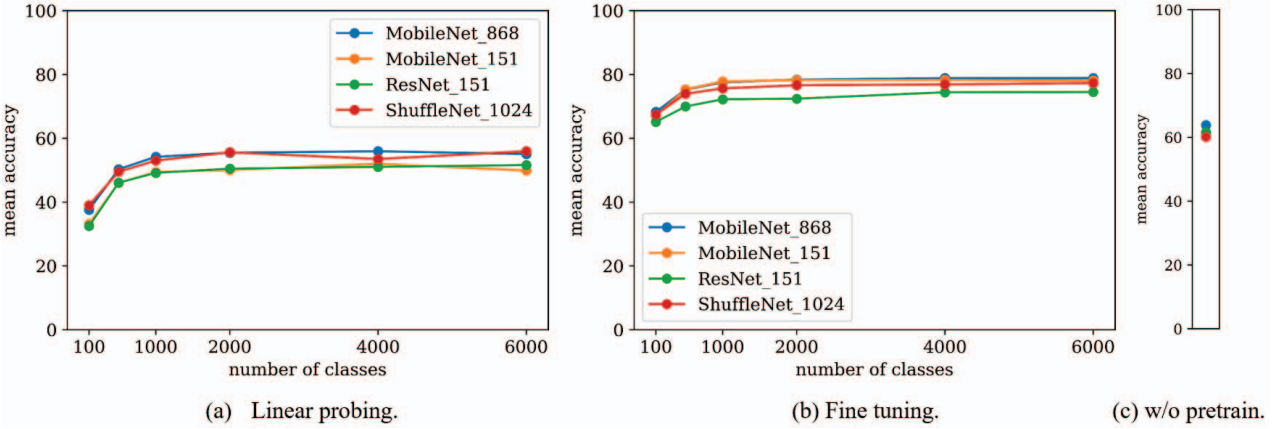

Figure 1: Mean accuracy on the downstream tasks as a function of the number of classes, and using 500 images per class for all tested deep architectures.

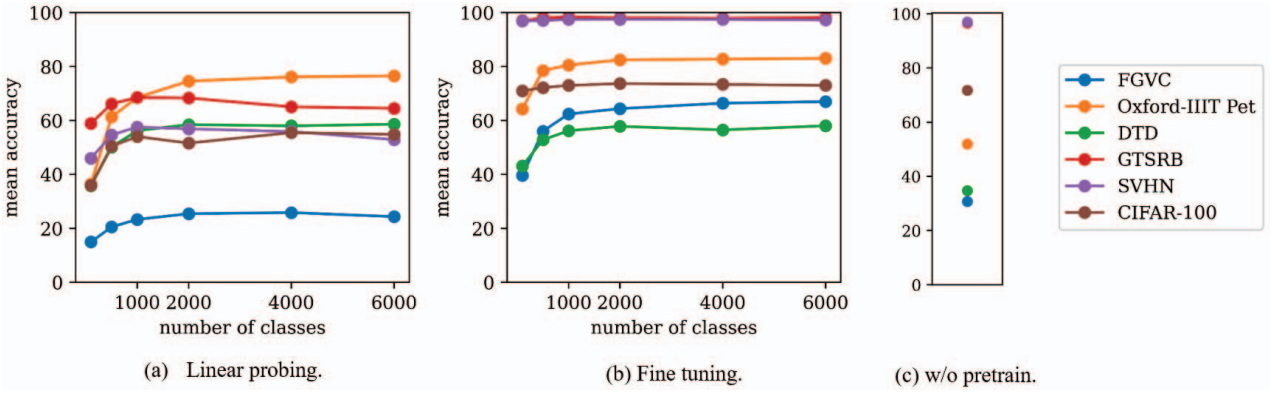

Figure 2: Detailed accuracy for each downstream task with MobileNet_{868}.

cused on larger deep architectures and tested much larger upstream datasets. The conclusion is that pre-training of compact deep architectures is effective with an upstream dataset which includes approximately 1M diversified images.

An interesting observation is that fine-tuning-based training is clearly better than a direct use of features learned upstream via linear probing. The accuracy gain when using the first strategy is over 15 points for all tested numbers of classes of the upstream dataset, and all backbone architectures. A similar finding was already reported in literature [16, 18] for larger deep architectures, and is confirmed here for compact architectures, which are adapted for computationally-constrained environments. We also note that the gain offered by fine tuning over linear probing is larger when upstream training is done with a low number of classes (up to 1000). This is explained by the stronger sensitivity of linear probing to the quality of the upstream features, due to the direct use of features versus an adaptation of them for downstream tasks during fine tuning.

The performance obtained with the four tested architectures varies for both downstream training strategies (linear probing in Figure 1a, fine tuning in Figure 1b). Globally, MobileNetv2 and ShuffleNetv2 behave better than ResNet after scaling to 1M parameters. This is somewhat expected since the first two types of architectures were designed purposely for computationally-constrained environments. Interestingly, the difference between MobileNet_{868} and MobileNet_{151} is much smaller when the upstream models are fine tuned (Figure 1b) compared to linear probing (Figure 1a). This indicates that models which have a wider output are more adequate for linear probing if the overall number of parameters is equivalent. An explanation resides in the higher dimensionality of the frozen features produced by MobileNet_{868}, which favors the separability of classes downstream. This finding is in line with the well-known result reported for wide residual networks [37].

We propose a per-dataset view of results obtained with linear probing and with fine tuning in Figure 2. These results are reported with a MobileNet_{868}, which provides the

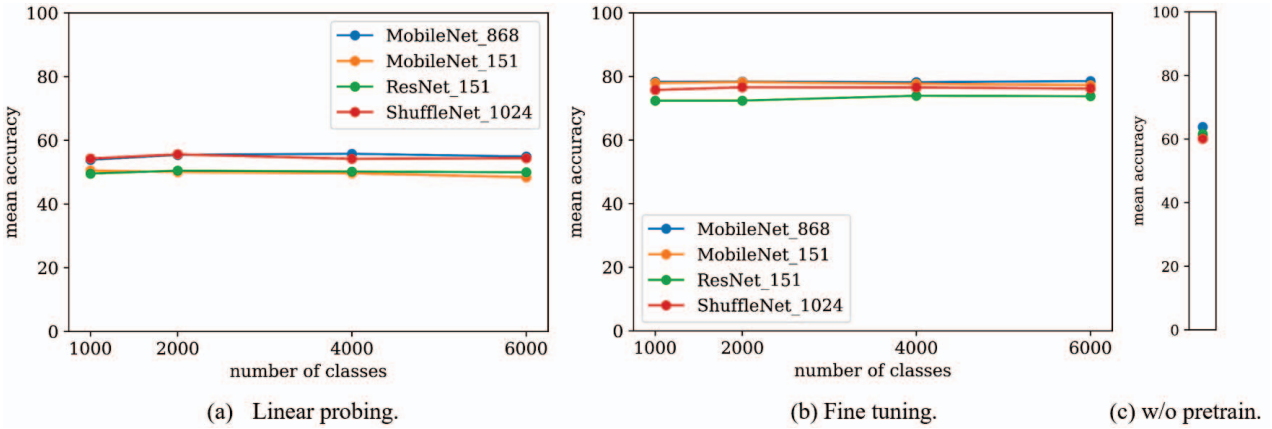

Figure 3: Mean accuracy on the downstream tasks when the total size of the dataset is constant (1M images) and the number of images per class decreases when the number of classes increases. The minimum number of classes is 1000 because ImageNet does not contain enough leaf classes with enough images to run experiments with 100 and 500 classes.

best overall results in Figures 1a and 1b. Fine tuning is better than linear probing for five datasets out of six and number of classes included in the upstream datasets. The differences are much stronger for downstream tasks whose domain shift compared to the upstream task is larger. This is the case of FGVC, SVHN and GTSRB, three datasets focused on aircrafts, house number plates and street signs. The domains are not well represented in ImageNet and the retraining of all weights during fine tuning is clearly needed. Linear probing is better than fine tuning only for DTD . This result might be explained by the low total size of this dataset, which includes only 1600 training images, combined with the large domain shift between ImageNet and this texture-focused dataset.

We also report performance with downstream tasks without pre-training on average and per dataset to assess the overall effect of pre-training. The difference between the best and worst of the four tested architectures is 4 points in Figure 1c, but there are strong differences between individual datasets (Figure 2c). The global comparison shows that the use of pre-training for fine tuning brings a significant improvement compared to training from scratch. The dataset-level analysis (Figure 2) gives more insight into the merits and limitations of pre-training with the two downstream training strategies. Linear probing is effective for small domain shifts between upstream and downstream tasks (Oxford-IIIT Pet) or when the dataset size is small (DTD), but provides lower performance in the other cases. This is expected since the features are not adapted to each task. Fine tuning provides similar performance to that of training from scratch for easy tasks, such as GTSRB and SVHN, and brings important improvements for FGVC, Oxford-IIIT Pet and DTD.

4.2. Effect of pre-training with a constant-size dataset

We complement the analysis from Subsection 4.1 with experiments run with a dataset which total size is kept fixed at 1M images. Here, the number of images per class decreases when the number of selected classes increases. The dataset is balanced, meaning the samples are distributed evenly between classes. This corresponds to an upstream training with a fixed sample budget.

The figures 3a and 3b show the mean accuracy on the six downstream task with linear probing and full fine-tuning. The global trends are similar to those observed in Figure 1, as is the accuracy obtained with linear probing and fine tuning in different configurations.

We observe a performance gain of up to two points when the number of classes in the dataset changes from 1000 to 2000. Beyond 2000 classes, performance seems to oscillate, and is even slightly decreasing for linear probing (Figure 3a). For this strategy, the obtained accuracy decreases in all tested configurations except one when the number of classes increases from 4000 to 6000. This result can be explained by 2 opposite phenomena. While a more diversified pre-training dataset is likely to lead to a better representation, a larger number of classes also makes the upstream task more difficult. Our finding is consistent with the saturation of downstream performance beyond a certain point, even when upstream performance is improved [1]. Here, the improvement of the representation brought by the addition of new classes is degraded by the growth of the complexity of the task, and by the scarcer representation of each class when the total number of classes increases.

The comparison of the results for 4000 and 6000 classes from Figures 1 and 3 is interesting because the total size of the upstream dataset is smaller in the latter configuration.

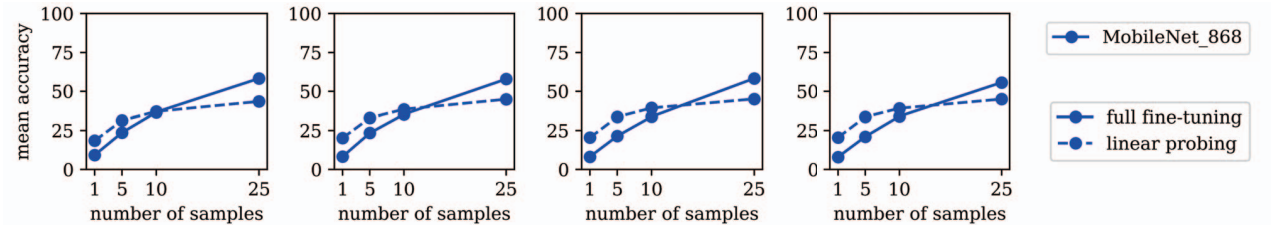

Figure 4: Mean accuracy on the downstream tasks in low-shots four settings. The number of image per class is constant (500) in the upstream dataset. It includes 1000, 2000, 4000, 6000 classes, from left to right.

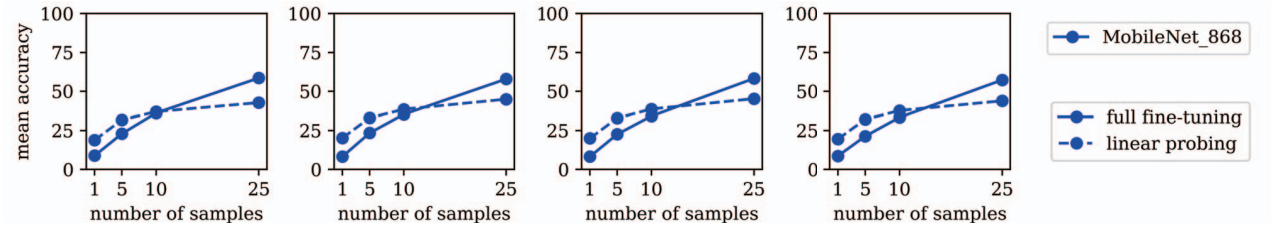

Figure 5: Mean accuracy on the downstream tasks in low-shots four settings. The total size of the upstream dataset stays constant (1M images). It includes 1000, 2000, 4000, 6000 classes, from left to right.

There are 2M and 3M images for 4000 and 6000 classes in Figure 1, but only 1M in Figure 3. This finding shows that a representation of upstream classes with fewer images does not have a significant impact on downstream performance.

4.3. Effect of pre-training in few-shot scenarios

Pre-training is particularly useful when only few samples are available per class since deep neural networks are data hungry [5]. Performance is reported for pre-training with MobileNet_{868} for 1M parameters, the configuration which works best for downstream training with all data. We plot accuracy for pre-training with 1000 and 6000 classes for each model to also assess the influence of this parameter. We investigate the performance on downstream tasks for different few-shot learning regimes. We again report results with two upstream pre-training strategies: the number of images per class is constant in Figure 4 and the total size of the dataset is constant 5. The observation that performance is very similar in the two upstream dataset configurations remains valid for all tested few-shot settings. Interestingly, the obtained results indicate that linear probing is significantly better than full fine tuning up to 10 samples per class. The full training of the downstream models is difficult with very few samples due to the occurrence of overfitting [36]. The gap between the two training strategies narrows when the number of samples increases. Fine tuning becomes better than linear probing when 25 samples are available. Naturally, this tendency is even clearer when all samples are available for downstream tasks, as we discussed in Subsection 4.1. Our results are at odds with those reported previously [39], regarding the superiority of fine tuning over linear probing even in a few-shot setting. The main difference comes from the scale of the networks, with much larger architectures being tested in [39]. The observations made here show that the downstream training strategy should be adapted depending on the quantity of data available for target datasets. In practice, the pre-trained model should be used as a frozen feature extractor if the number of samples in the downstream task is smaller.

Performance of few-shot learning is similar for upstream dataset variants with a constant number of images per class (Figure 4) and with a constant size dataset (Figure 5). This echoes the results obtained when using the full downstream datasets. The difference between pre-training with 1000 and 6000 datasets is larger for linear probing compared to fine tuning. The accuracy gain between these two variants of the upstream dataset reaches approximately 2, 2 and 1.5 accuracy points for 1-shot, 5-shots and 25-shots settings, respectively. This is expected since linear probing makes direct usage of upstream features, and thus benefits more from strong pre-trained representations.

The global comparison of linear probing and fine tuning in few shot scenarios, presented in Figures 4 and 5, is refined with a presentation of the accuracy per dataset. Figures 6 and 7 illustrate results for MobileNet_{868} pre-trained with 6000 and 1000 classes, respectively. Linear probing is clearly better than fine tuning for datasets which are semantically related to the content on the pre-trained models, such as Oxford-IIIT Pet and CIFAR-100. ImageNet [3], the dataset used for pre-training, includes a large number of classes which describe the natural world, which are also well represented in the three downstream datasets for which

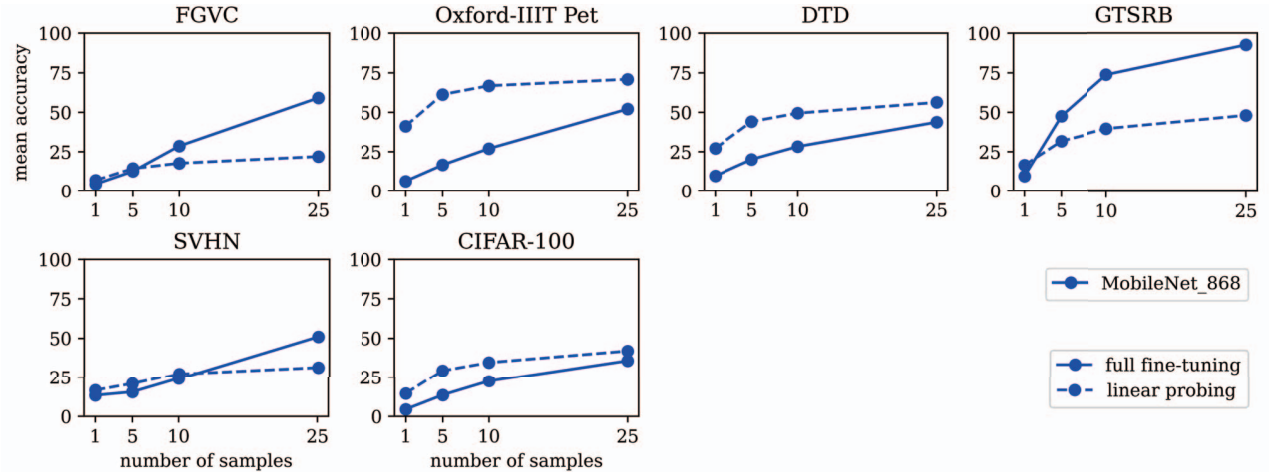

Figure 6: Downstream accuracy for each downstream task with 6000 classes and 500 images per classes for the pre-training.

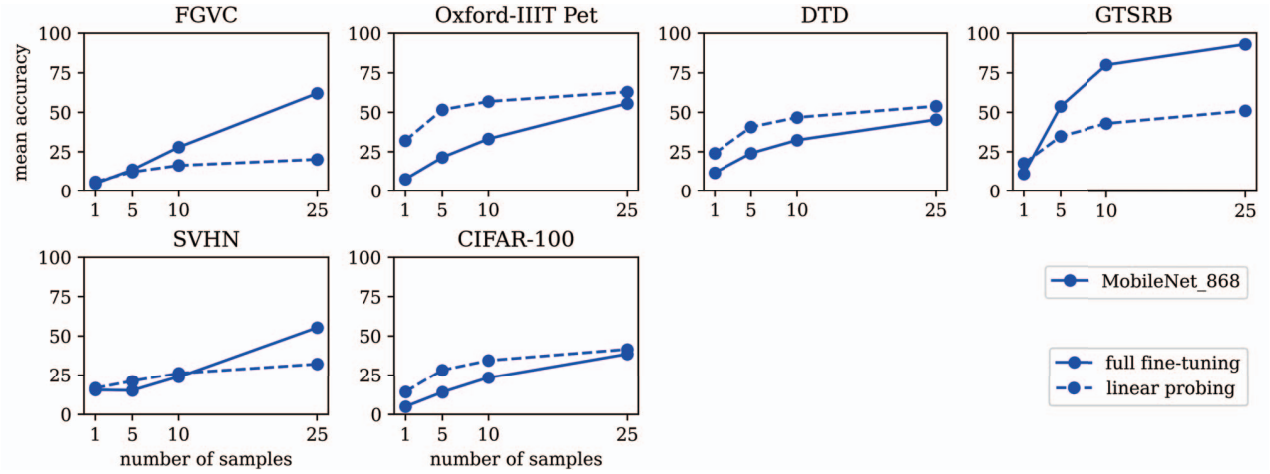

Figure 7: Downstream accuracy for each downstream task with 1000 classes and 500 images per classes for the pre-training.

probing has good results in few-shot scenarios. Linear probing accuracy is also better for DTD, a texture dataset which does not benefit from fine tuning even when all its images are available (Figure 2). Fine tuning is the better option for FGVC, SVHN, and GTSRB, the three datasets with a larger domain shift compared to ImageNet.

5. Conclusion

We investigate transfer learning in image recognition under constraints through a comprehensive empirical study, which analyzes the roles of the dataset used for upstream training, the performance of different deep architectures, the results obtained with two opposite training strategies. Our experiments confirm findings reported in previous studies regarding performance saturation for large deep architectures [1, 8, 20]. It shows that the phenomenon appears faster in terms of scale of the upstream dataset, due to the compactness of tested architectures. As a result, the conclusion to use increasingly larger pre-training datasets to improve performance [1, 8, 20] does not seem justified for compact deep architectures. In contrast to past studies [16, 39] which assert that full fine tuning is preferable to linear probing, our result shows which strategy is better depending on number of images available. Linear probing is a good strategy when the domain shift is small and/or when the available number of samples per class in the downstream task is low. It is also interesting due to its lower computational complexity, which is important in constrained environments [33].

We used variants of a fully-supervised dataset for pre-training. It would be useful to extend it by testing weakly-supervised and unsupervised pre-training. It would be equally interesting to explore ways to predict an adapted downstream training strategy based on an analysis of the domain shift between the upstream and downstream tasks.

References

[1] Samira Abnar, Mostafa Dehghani, Behnam Neyshabur, and Hanie Sedghi. Exploring the limits of large scale pre-training. *arXiv preprint arXiv:2110.02095*, 2021. 1, 2, 3, 4, 6, 8

[2] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 3, 4

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 3, 7

[4] 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. 1

[5] Vincent Dumoulin, Neil Houlsby, Utku Evci, Xiaohua Zhai, Ross Goroshin, Sylvain Gelly, and Hugo Larochelle. Comparing transfer and meta learning approaches on a unified few-shot classification benchmark. *arXiv preprint arXiv:2104.02638*, 2021. 1, 2, 7

[6] Alaaeldin El-Nouby, Gautier Izacard, Hugo Touvron, Ivan Laptev, Hervé Jegou, and Edouard Grave. Are large-scale datasets necessary for self-supervised pre-training? *arXiv preprint arXiv:2112.10740*, 2021. 1, 2, 4

[7] Utku Evci, Vincent Dumoulin, Hugo Larochelle, and Michael C Mozer. Head2Toe: Utilizing intermediate representations for better transfer learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 6009–6033. PMLR, 17–23 Jul 2022. 1, 2, 3

[8] Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, et al. Self-supervised pretraining of visual features in the wild. *arXiv preprint arXiv:2103.01988*, 2021. 1, 4, 8

[9] Yunhui Guo, Yandong Li, Liqiang Wang, and Tajana Rosing. Adafilter: Adaptive filter fine-tuning for deep transfer learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4060–4066, 2020. 2, 3

[10] Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris. Spottune: transfer learning through adaptive fine-tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4805–4814, 2019. 2, 3

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, CVPR, 2016. 3, 4

[12] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 4

[13] Mohammadreza Iman, Khaled Rasheed, and Hamid R Arabnia. A review of deep transfer learning and recent advancements. *arXiv preprint arXiv:2201.09679*, 2022. 1

[14] Junguang Jiang, Yang Shu, Jianmin Wang, and Mingsheng Long. Transferability in deep learning: A survey. *arXiv preprint arXiv:2201.05867*, 2022. 1

[15] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *European conference on computer vision*, pages 491–507. Springer, 2020. 1

[16] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2661–2671, 2019. 1, 2, 3, 4, 5, 8

[17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 3, 4

[18] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022. 2, 3, 5

[19] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018. 4

[20] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196, 2018. 2, 4, 8

[21] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 3, 4

[22] Gary Marcus. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*, 2018. 1

[23] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 3, 4

[24] Jiquan Ngiam, Daiyi Peng, Vijay Vasudevan, Simon Kornblith, Quoc V Le, and Ruoming Pang. Domain adaptive transfer learning with specialist models. *arXiv preprint arXiv:1811.07056*, 2018. 2

[25] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009. 1

[26] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 3, 4

[27] Eleanor Rosch. Principles of categorization. In Eleanor Rosch and B. B. Lloyd, editors, *Cognition and Categorization*, pages 27–48. Erlbaum, Hillsdale, NJ, 1978. 3

[28] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 4

[29] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014. 1, 3

[30] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019. 3

[31] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012. 3, 4

[32] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 2

[33] Paul N Whatmough, Chuteng Zhou, Patrick Hansen, Shreyas Kolala Venkataramanaiah, Jae-sun Seo, and Matthew Mattina. Fixynn: Efficient hardware for mobile computer vision via transfer learning. *arXiv preprint arXiv:1902.11128*, 2019. 2, 3, 4, 8

[34] Zheng Xie, Zhiquan Wen, Jing Liu, Zhiqiang Liu, Xixian Wu, and Mingkui Tan. Deep transferring quantization. In *European Conference on Computer Vision*, pages 625–642. Springer, 2020. 2

[35] Xiaowei Xu, Yukun Ding, Sharon Xiaobo Hu, Michael Niemier, Jason Cong, Yu Hu, and Yiyu Shi. Scaling for edge inference of deep neural networks. *Nature Electronics*, 1(4):216–222, 2018. 1

[36] Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. In *International Conference on Learning Representations (ICLR)*, 2021. 7

[37] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 5

[38] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022. 2, 3

[39] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019. 1, 2, 3, 4, 7, 8