DOMAIN ADAPTIVE TRANSFER LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Transfer learning is a widely used method to build high performing computer vision models. In this paper, we study the efficacy of transfer learning by examining how the choice of data impacts performance. We find that more pre-training data does not always help, and transfer performance depends on a judicious choice of pre-training data. These findings are important given the continued increase in dataset sizes. We further propose domain adaptive transfer learning, a simple and effective pre-training method using importance weights computed based on the target dataset. Our methods achieve state-of-the-art results on multiple fine-grained classification datasets and are well-suited for use in practice.

1 INTRODUCTION

Transfer learning using pre-trained models is one of the most successfully applied methods in the field of computer vision. In practice, a model is first trained on a large labeled dataset such as ImageNet (Russakovsky et al., 2015), and then fine-tuned on a target dataset. During fine-tuning, a new classification layer is learned from scratch, but the parameters for the rest of the network layers are initialized from the ImageNet pre-trained model. This method to initialize training of image models has proven to be highly successful and is now a central component of object recognition (Razavian et al., 2014), detection (Girshick, 2015; Ren et al., 2015; Huang et al., 2017), and segmentation (Shelhamer et al., 2017; Chen et al., 2018; He et al., 2017).

By initializing the network with ImageNet pre-trained parameters, models train with higher accuracy and converge faster, requiring less training time. They have also achieved good performance when the target dataset is small. Most prior work have considered only ImageNet as the source of pre-training data due its large size and availability. In this work, we explore how the choice of pre-training data can impact the accuracy of the model when fine-tuned on a new dataset.

To motivate the problem, consider a target task where the goal is to classify images of different food items (e.g., ‘hot dog’ v.s. ‘hamburger’) for a mobile application (Anglade, 2017). A straight-forward approach to applying transfer learning would be to employ an ImageNet pre-trained model fine-tuned on a food-specific dataset. However, we might wonder whether the pre-trained model, having learned to discriminate between irrelevant categories (e.g., ‘dogs’ vs. ‘cats’), would be helpful in this case of food classification. More generally, if we have access to a large database of images, we might ask: is it more effective to pre-train a classifier on all the images, or just a subset that reflect food-like items?

Furthermore, instead of making a hard decision when selecting pre-training images, we can consider a soft decision that weights each example based on their relevancy to the target task. This could be estimated by comparing the distributions of the source pre-training data and the target dataset. This approach has parallels to the covariate shift problem often encountered in survey and experimental design (Shimodaira, 2000).

We study different choices of source pre-training data and show that a judicious choice can lead to better performance on all target datasets we studied. Furthermore, we propose domain adaptive transfer learning - a simple and effective pre-training method based on importance weights computed based on the target dataset.

1.1 Summary of findings

More pre-training data does not always help. We find that using the largest pre-training dataset does not always result in the best performance. By comparing results of transfer learning on different subsets of pre-training data, we find that the best results are obtained when irrelevant examples are discounted. This effect is particularly pronounced with fine-grained classification datasets.

Matching to the target dataset distribution improves transfer learning. We demonstrate a simple and computationally-efficient method to determine relevant examples for pre-training. Our method computes importance weights for examples on a pre-training dataset and is competitive with hand-curated pre-training datasets. Using this method, we obtain state-of-the-art results on the fine-grained classification datasets we studied (e.g., Birdsnap, Oxford Pets, Food-101).

Fine-grained target tasks require fine-grained pre-training. We find that transfer learning performance is dependent on whether the pre-training data captures similar discriminative factors of variations to the target data. When features are learned on coarse grained classes, we do not observe significant benefits transferred to fine-grained datasets.

2 Related Work

The success of applying convolution neural networks to the ImageNet classification problem (Krizhevsky et al., 2012) led to the finding that the features learned by a convolutional neural network perform well on a variety of image classification problems (Razavian et al., 2014; Donahue et al., 2014). Further fine-tuning of the entire model was found to improve performance (Agrawal et al., 2014).

Yosinski et al. (2014) conducted a study of how transferable ImageNet features are, finding that the higher layers of the network tend to specialize to the original task, and that the neurons in different layers in a network were highly co-adapted. They also showed that distance between tasks matters for transfer learning and examined two different subsets (man-made v.s. natural objects). Azizpour et al. (2016) also examined different factors of model design such as depth, width, data diversity and density. They compared data similarity to ImageNet based on the task type: whether it was classification, attribute detection, fine-grained classification, compositional, or instance retrieval.

Pre-training on weakly labeled or noisy data was also found to be effective for transfer learning. Krause et al. (2016) obtained additional noisy training examples by searching the web with the class labels. We note that our method does not use the class labels to collect additional data. Mahajan et al. (2018) were able to attain impressive ImageNet performance by pre-training on 3 billion images from Instagram. Notably, they found that it was important to appropriately select hash-tags (used as weak labels) for source pre-training.

Understanding the similarity between datasets based on their content was studied by Cui et al. (2018), who suggest using the Earth Mover's Distance (EMD) as a distance measure between datasets. They constructed two pre-training datasets by selecting subsets of ImageNet and iNaturalist, and showed that selecting an appropriate pre-training subset was important for good performance. Ge & Yu (2017) used features from filter bank responses to select nearest neighbor source training examples and demonstrated better performance compared to using the entire source dataset. Zamir et al. (2018) define a method to compute transferability *between tasks* on the same input; our work focuses on computing relationships *between different input datasets*.

In a comprehensive comparison, Kornblith et al. (2018) studied fine-tuning a variety of models on multiple datasets, and showed that performance on ImageNet correlated well with fine-tuning performance. Notably, they found that transfer learning with ImageNet was ineffective for small, fine-grained datasets.

Our approach is related to domain adaptation which assumes that the *training* and *test* set have differing distributions (Shimodaira, 2000). We adopt similar ideas of importance weighting examples (Sugiyama et al., 2007; Saerens et al., 2002; Zhang et al., 2013) and adapt them to the *pre-training* step instead, showing that this is an effective approach.

In this work, we show that transfer learning to fine-grained datasets is sensitive to the choice of pre-training data, and demonstrate how to select pre-training data to significantly improve transfer learning performance. We build on the work of (Cui et al., 2018; Ge & Yu, 2017), demonstrating the effectiveness of constructing pre-training datasets. Furthermore, we present a simple, scalable, and computationally-efficient way to construct pre-training datasets.

3 TRANSFER LEARNING SETUP

We use the ANON[1] (Anonymous) and ImageNet (Russakovsky et al., 2015) datasets as our source pre-training data and consider a range of target datasets for fine-tuning (Section 3.2). For each target dataset, we consider different strategies for selecting pre-training data, and compare the fine-tuned accuracy. We do not perform any label alignment between the source and target datasets. During fine-tuning, the classification layer in the network is trained from random initialization. The following sections describe the datasets and experiments in further detail.

3.1 SOURCE PRE-TRAINING DATA

The ANON dataset has 300 million images and 18,291 classes. Each image can have multiple labels and on average, each image has 1.26 labels. The large number of labels include many fine-grained categories, for example, there are 1,165 different categories for animals. While the labels are noisy and often missing, we do not find this to a be a problem for transfer learning in practice. The labels form a semantic hierarchy: for example, the label 'mode of transport' includes the label 'vehicle', which in turn includes 'car'.

The semantic hierarchy of the labels suggests a straight-forward approach to constructing different subsets of ANON as source pre-training data. Given a label, we can select all of its child labels in the hierarchy to form a label set, with the corresponding set of training examples. We created 7 subsets of ANON across a range of labels[2] (Table 1).

Table 1: ANON subsets by hand-selecting labels.

Top Ancestor Label	# Examples	# Classes
Entire Dataset	300M	18,291
Animal	33.5M	2,992
Bird	5.4M	403
Car	27.9M	2,959
Aircraft	3.1M	418
Vehicle	43.7M	3,969
Transport	45.0M	3,987
Food	18.4M	3,532

However, creating subsets using the label hierarchy can be limiting for several reasons: (a) the number of examples per label are pre-defined by the ANON dataset; (b) not all child labels may be relevant; (c) a union over different sub-trees of the hierarchy may be desired; and (d) not all source datasets have richly-defined label hierarchies. In section 3.3, we discuss a domain adaptive transfer learning approach that automatically selects and weights the relevant pre-training data.

3.2 TARGET TRAINING DATASET

We evaluate the performance of transfer learning on a range of classification datasets (Table 2) that include both general and fine-grained classification problems. Using the same method as Krause

[1] Dataset anonymized for ICLR submission.

[2] The following parent-child relationships exists in the label hierarchy: bird $\subset$ animal; car $\subset$ vehicle $\subset$ transport; aircraft $\subset$ vehicle $\subset$ transport. We note that Anonymous excluded classes with too few training examples during training, while we include all classes available.

et al. (2016), we ensured that the source pre-training data did not contain any of the target training data by removing all near-duplicates of the target training and test data from the ANON dataset[3].

Table 2: Target datasets for fine-tuning.

Target Dataset	# Training Examples	# Test Examples	# Classes
CIFAR-10 (Krizhevsky & Hinton, 2009)	50,000	10,000	10
Birdsnap (Berg et al., 2014)	47,386	2,443	78
Stanford Cars (Krause et al., 2013)	8,144	8,041	196
FGVC Aircraft (Maji et al., 2013)	6,667	3,333	100
Oxford-IIIT Pets (Parkhi et al., 2012)	3,680	3,369	37
Food-101 (Bossard et al., 2014)	75,750	25,250	101

3.3 Domain adaptive transfer learning by importance weighting

In this section, we propose domain adaptive transfer learning, a simple and effective way to weight examples during pre-training. Let us start by considering a simplified setting where our source and target datasets are over the same set of values in pixels x, and labels y; we will relax this assumption later in this section.

During pre-training, we usually minimize parameters θ over a loss function $\mathbb{E}_{x,y\sim D_s}[L(f_\theta(x),y)]$ computed empirically over a source dataset D_s. $L(f_\theta(x),y)$ is often the cross entropy loss between the predictions of the model $f_\theta(x)$ and the ground-truth labels y. However, the distribution of source pre-training dataset D_s may differ from the target dataset D_t. This could be detrimental as the model may emphasize features which are not relevant to the target dataset. We will mitigate this by up-weighting the examples that are most relevant to the target dataset. This is closely related[4] to prior probability shift (Saerens et al., 2002; Storkey, 2009) also known as target shift (Zhang et al., 2013).

We start by considering optimizing the loss function over the target dataset, D_t instead:

$$\mathbb{E}_{x,y\sim D_t}\big[L(f_\theta(x),y)\big] = \sum_{x,y} P_t(x,y)L(f_\theta(x),y)$$

where we use P_s and P_t to denote distributions over the source and target datasets respectively. We first reformulate the loss to include the source dataset D_s:

$$= \sum_{x,y} P_s(x,y)\frac{P_t(x,y)}{P_s(x,y)}L(f_\theta(x),y) = \sum_{x,y} P_s(x,y)\frac{P_t(y)P_t(x|y)}{P_s(y)P_s(x|y)}L(f_\theta(x),y)$$

Next, we make the assumption that $P_s(x|y) \approx P_t(x|y)$, that is the distribution of examples given a particular label in the source dataset is approximately the same as that of the target dataset. We find this assumption reasonable in practice: for example, the distribution of 'bulldog' images from a large natural image dataset can be expected to be similar to that of a smaller animal-only dataset. This assumption also allows us to avoid having to directly model the data distribution $P(x)$.

Cancelling out the terms, we obtain:

$$\approx \sum_{x,y} P_s(x,y)\frac{P_t(y)}{P_s(y)}L(f_\theta(x),y) = \mathbb{E}_{x,y\sim D_s}\big[\frac{P_t(y)}{P_s(y)}L(f_\theta(x),y)\big]$$

Intuitively, $P_t(y)$ describes the distribution of labels in the target dataset, and $P_t(y)/P_s(y)$ reweights classes during source pre-training so that the class distribution statistics match $P_t(y)$. We refer to

[3] We used a CNN-based duplicate detector and chose a conservative threshold for computing near-duplicates to err on the side of ensuring that duplicates were removed. We removed a total of 48k examples from ANON, corresponding to duplicates that were found in target datasets.

[4] Prior work on prior probability shift usually considered shifts between train and test set, while we instead consider differences between the pre-training and training datasets.

$P_t(y)/P_s(y)$ as importance weights and call this approach of pre-training *Domain Adaptive Transfer Learning*.

For this approach to be applicable in practice, we need to relax the earlier assumption that the source and target datasets share the same label space. Our goal is to estimate $P_t(y)/P_s(y)$ for each label in the source dataset. The challenge is that the source and target datasets have different sets of labels. Our solution is to estimate both $P_t(y)$ and $P_s(y)$ for labels in the source domain. The denominator $P_s(y)$ is obtained by dividing the number of times a label appears by the total number of source dataset examples. To estimate $P_t(y)$, we use a classifier to compute the probabilities of *labels from source dataset* on *examples from the target dataset*.

Concretely, we first train an image classification model on the entire source dataset. Next, we feed only the images from the target dataset into this model to obtain a prediction for each target example. The predictions are averaged across target examples, providing an estimate of $P_t(y)$, where y is specified over the source label space. We emphasize that this method does not use the target labels when computing importance weights.

Our approach is in contrast to Ge & Yu (2017), which is computationally expensive as they compute a similarity metric between every pair of images in the source dataset and target dataset. It is also more adaptive than Cui et al. (2018), which suggests selecting appropriate labels to pretrain on, without specifying a weight on each label.

4 Experiments

We used the Inception v3 (Szegedy et al., 2016), and AmoebaNet-B (Real et al., 2018) models in our experiments.

For Inception v3 models, we pre-train from random initialization for 2,000,000 steps using Stochastic Gradient Descent (SGD) with Nesterov momentum. Each mini-batch contained 1,024 examples. The same weight regularization and learning rate parameters were used for all pre-trained models and were selected based on a separate hold-out dataset. We used a learning rate schedule that first starts with a linear ramp up for 20,000 steps, followed by cosine decay.

AmoebaNet-B models followed a similar setup with pre-training from random initialization for 250,000 steps using SGD and Nesterov momentum. We used larger mini-batches of 2,048 examples to speed up training. The same weight regularization and learning rate parameters were used for all models, and matched the parameters that Real et al. (2018) used for ImageNet training. We chose to use AmoebaNet-B with settings (N=18, F=512), resulting in over 550 million parameters when trained on ImageNet, so as to evaluate our methods on a large model.

During fine-tuning, we used a randomly initialized classification layer in place of the pre-trained classification layer. Models were trained for 20,000 steps using SGD with momentum. Each mini-batch contained 256 examples. The weight regularization and learning rate parameters were determined using a hold-out validation set. We used a similar learning rate schedule with a linear ramp for 2,000 steps, followed by cosine decay.

For domain adaptive transfer learning, we found that adding a smooth prior when computing $P_t(y)$ helped performance with ImageNet as a source pre-training data. Hence, we used a temperature[5] of 2.0 when computing the softmax predictions for the computation of the importance weights.

4.1 Pre-training setup

While it is possible to directly perform pre-training with importance weights, we found it challenging as the importance weights varied significantly. When pre-training on a large dataset, this means that it is possible to have batches of data that are skewed in their weights with many examples weighted lightly. This is also computationally inefficient as the examples with very small weights contribute little to the gradients during training.

Hence, we created pre-training datasets by sampling examples from the source dataset using the importance weights. We start by choosing a desired pre-training dataset size, often large. We then

[5] The logits are divided by the temperature before computing the softmax.

Table 3: Transfer learning results with Inception v3. Each row corresponds to a pre-training method. Adaptive transfer refers to our proposed method described in section 3.3. Each column corresponds to one target dataset. Results reported are top-1 accuracy for all datasets except Oxford-IIIT Pets, where we report mean accuracy per class. All results are averaged over 5 fine-tuning runs. Adaptive transfer is better or competitive with the hand selected subsets.

	Target Dataset					
Pre-training Method	Birdsnap	Oxford-IIIT Pets	Stanford Cars	FGVC Aircraft	Food-101	CIFAR-10
Entire ANON Dataset	74.2	92.5	94.0	88.2	88.6	97.6
ANON - Bird	80.7	86.4	88.1	74.9	87.5	96.9
ANON - Animal	77.8	96.7	89.1	78.2	89.2	98.1
ANON - Car	73.4	79.8	**96.0**	82.1	86.1	93.0
ANON - Aircraft	73.4	78.7	88.2	91.1	87.1	96.1
ANON - Vehicle	74.2	79.6	95.8	86.8	81.6	96.4
ANON - Transport	74.4	78.4	95.9	88.4	86.9	96.2
ANON - Food	74.9	81.1	90.3	85.6	93.5	96.4
ANON - Adaptive Transfer	**81.7**	**97.1**	95.7	**94.1**	**94.1**	**98.3**
ImageNet - Entire Dataset	77.2	93.3	91.5	88.8	88.7	97.4
ImageNet - Adaptive Transfer	76.6	94.1	92.1	87.8	88.9	97.7
Random Initialization	75.2	80.8	92.1	88.3	86.4	95.7

sample examples from the source dataset at a rate proportional to the importance weights, repeating examples as needed. We report results that construct a pre-training dataset of 80 million examples for ANON, and 2 million examples for ImageNet. We used the same sampled pre-training dataset with both the Inception v3 and AmoebaNet-B experiments.

4.2 Transfer learning results

Domain adaptive transfer learning is better. When the source pre-training domain matches the target dataset, such as in ANON-Bird to Birdsnap or ANON-Cars to Stanford Cars, transfer learning is most effective (Table 3). However, when the domains are mismatched, we observe negative transfer: ANON-Cars fine-tuned on Birdsnap performs poorly. Strikingly, this extends to categories which are intuitively close: aircrafts and cars. The features learned to discriminate between types of cars does not extend to aircrafts, and vice-versa.

More data is not necessarily better. Remarkably, more data during pre-training can hurt transfer learning performance. In all cases, the model pre-trained on the entire ANON dataset did worse than models trained on more specific subsets. These results are surprising as common wisdom suggests that more pre-training data should improve transfer learning performance if generic features are learned. Instead, we find that it is important to determine how relevant additional data is.

The ImageNet results with Domain Adaptive Transfer further emphasize this point. For ImageNet with Adaptive Transfer, each pre-training dataset only has around 450k unique examples. While this is less than half of the full ImageNet dataset of 1.2 million examples, the transfer learning results are slightly better than using the full ImageNet dataset for many of the target datasets.

Domain adaptive transfer is effective. When pre-training with ANON and ImageNet, we find that the domain adaptive transfer models are better or competitive with manually selected labels from the hierarchy. For datasets that are composed of multiple categories such as CIFAR-10 which includes animals and vehicles, we find further improved results since the constructed dataset includes multiple different categories.

In Figure 1, we observe that the distributions are much more concentrated with FGVC Aircraft and Stanford Cars: this arises from the fact that ImageNet has only coarse-grained labels for aircraft and cars. In effect, ImageNet captures less of the discriminative factors of variation that is captured in either FGVC Aircraft and Stanford Cars. Hence, we observe that transfer learning only improves the results slightly.

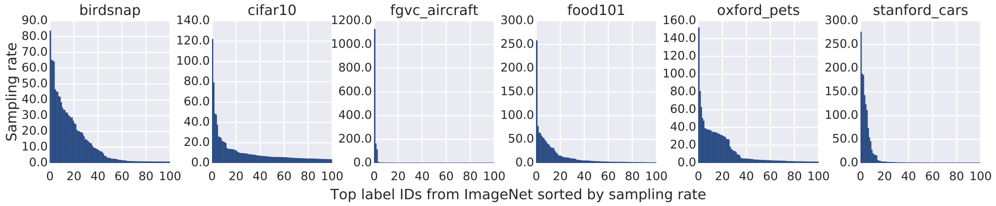

Figure 1: Distribution of importance weights for each target dataset when using ImageNet as a source pre-training dataset. The horizontal axis represents the top 100 ImageNet labels sorted by importance weight for each dataset; each dataset has a different order. The distributions vary widely between target datasets. FGVC Aircraft selects only a few labels that turn out to be coarse grained, whereas Oxford Pets selects a wider variety of fine-grained labels. This reflects the inherent bias in the ImageNet dataset.

4.3 Comparing pre-training sampling mechanisms

In section 4.1, we described a method to construct pre-training datasets from sampling the source dataset. This process also allows us to study the effect of different distributions. Rather than sampling with replacement, as we did earlier, we could also sample *without* replacement when constructing the pre-training dataset. When sampling without replacement, we deviate from the importance weights assigned, but gain more unique examples to train on. We compare these two methods of sampling: (a) sampling with replacement - 'same distribution matcher', and (b) sampling without replacement - 'elastic distribution matcher'. Details of the methods are elaborated in the appendix.

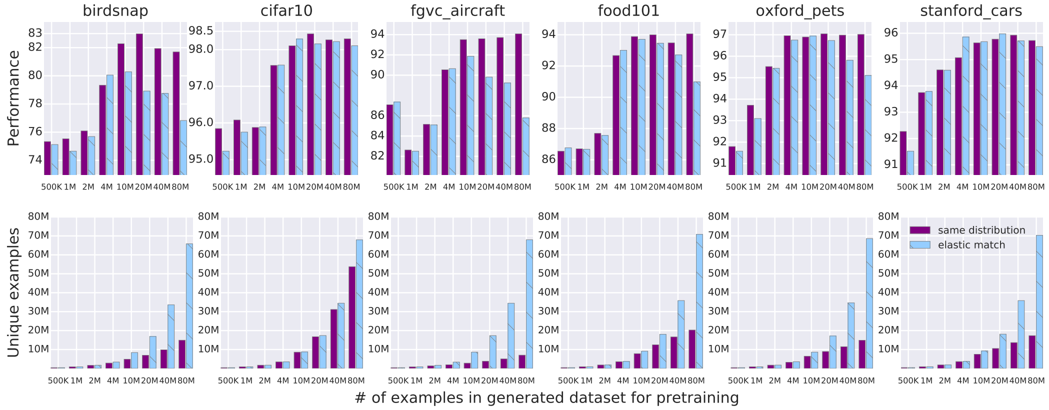

Figure 2: Performance (top) and unique examples (bottom) of the same distribution matcher and elastic distribution matcher at different sampled dataset sizes. We see that when dataset size increases, the performance of same distribution matcher increases and then saturates, while that of elastic distribution matcher drops after a peak. Notice that the elastic distribution matcher also has significantly more unique examples than same distribution matcher as the dataset size increases.

We find that the performance of the same distribution matcher increases, and then saturates. Conversely, the elastic distribution matcher performance first increases then decreases. Note that at the low end of the dataset sizes, both methods will generate similar datasets. Thus, the later decrease in performance from the elastic distribution matcher comes from diverging from the original desired distribution. This indicates that using the importance weights during pre-training is more important than having more unique examples to train on.

4.4 Results on large models

We furthered studied our method on large models to understand if large models are better able to generalize because the increased capacity enables them to capture more factors of variation. We conducted the same experiments on AmoebaNet-B, with over 550 million parameters.

Table 4: Transfer learning results with AmoebaNet-B.

	Target Fine-tuned Dataset					
Pre-training Data	Birdsnap	Oxford-IIIT Pets	Stanford Cars	FGVC Aircraft	Food-101	CIFAR-10
Entire ANON Dataset	80.3	94.5	95.3	90.5	92.0	98.6
ANON - Bird	**85.5**	90.4	92.0	86.9	90.7	97.8
ANON - Animal	84.1	96.4	93.2	90.0	92.3	**98.8**
ANON - Car	79.0	88.9	**96.2**	92.2	90.1	96.7
ANON - Aircraft	78.0	87.7	93.3	92.5	89.8	97.2
ANON - Vehicle	78.8	88.6	96.0	93.0	90.4	97.2
ANON - Transport	79.2	89.1	95.9	**93.1**	90.4	97.3
ANON - Food	79.7	89.2	92.6	88.7	95.1	97.5
ANON - Adaptive Transfer	85.1	**96.8**	95.8	92.8	**95.3**	98.6
ImageNet - Entire Dataset	80.8	94.5	94.2	90.7	91.7	98.0
ImageNet - Adaptive Transfer	80.7	95.1	93.5	89.2	91.5	98.0
Best Published Results	80.2[a,f]	94.3[b]	94.1[c]	92.9[c,f]	90.4[d]	98.5[e]

[a] Wei et al. (2018) [b] Kornblith et al. (2018) [c] Yu et al. [d] Cui et al. (2018) [e] Cubuk et al. (2018)
[f] Krause et al. (2016) achieve 83.9% on Birdsnap and 94.5% on FGVC Aircraft by adding additional bird and aircraft images during training of the source and target datasets; images were collected from Google image search using class names from the target datasets.

We found that the general findings persisted with AmoebaNet-B: (a) using the entire ANON dataset was always worse compared to an appropriate subset and (b) our domain adaptive transfer method was better or competitive with the hand selected subsets.

Furthermore, we find that the large model was also able to narrow the performance gap between the more general subsets and specific subsets: for example, the performance on Birdsnap between ANON-Bird and ANON-Animal is smaller with AmoebaNet-B compared to Inception v3. We also observe better transfer learning between the transportation datasets compared to Inception v3.

Our results are state of the art compared to the best published results (Table 4). The performance of the AmoebaNet-B was also better in all cases than Inception v3, except for the FGVC Aircraft dataset. This is consistent with Kornblith et al. (2018) who also found that Inception v3 did slightly better than NasNet-A (Zoph et al., 2017).

5 Discussion

Transfer learning appears most effective when the pre-trained model captures the discriminative factors of variation present in the target dataset. This is reflected in the significant overlap in the classes between ImageNet and other datasets such as Caltech101, CIFAR-10, etc. where transfer learning with ImageNet is successful. Our domain adaptive transfer method is also able to identify the relevant examples in the source pre-training dataset that capture these discriminative factors.

Conversely, the cases where transfer learning is less effective are when it fails to capture the discriminative factors. In the case of the "FGVC Aircraft" dataset (Maji et al., 2013), the task is to discriminate between 100 classes over manufacturer and models of aircraft (e.g., Boeing 737-700). However, ImageNet only has coarse grained labels for aircraft (e.g., airliner, airship). In this case, ImageNet models tend to learn to "group" different makes of aircraft together rather than differentiate them. It turns out that the ANON dataset has fine-grained labels for aircraft and is thus able to demonstrate better transfer learning efficacy.

Our results using AmoebaNet-B show that even large models transfer better when pre-trained on a subset of classes, suggesting that they make capacity trade-offs between the fine-grained classes when training on the entire dataset. This finding posits new research directions for developing large models that do not make such a trade-off.

We have seen an increase in dataset sizes since ImageNet; for example, the YFCC100M dataset (Thomee et al., 2016) has 100M examples. We have also seen developments of more efficient methods to train deep neural networks. Recent benchmarks (Coleman et al., 2018) demonstrate that it is possible to train a ResNet-50 model in half an hour, under fifty US dollars. This combination of data and compute will enable more opportunities to employ better methods for transfer learning.

REFERENCES

Pulkit Agrawal, Ross B. Girshick, and Jitendra Malik. Analyzing the performance of multilayer neural networks for object recognition. In *ECCV 2014*, volume 8695 of *Lecture Notes in Computer Science*, pp. 329–344. Springer, 2014.

Tim Anglade. How HBOs Silicon Valley built Not Hotdog with mobile TensorFlow, Keras & React Native, 2017.

Anonymous. Reference anonymized for conference submission.

Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. Factors of transferability for a generic convnet representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(9):1790–1802, 2016.

Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L. Alexander, David W. Jacobs, and Peter N. Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2019–2026, 2014.

Lukas Bossard, Matthieu Guillaumin, and Luc J. Van Gool. Food-101 - mining discriminative components with random forests. In David J. Fleet, Toms Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *ECCV 2014*, volume 8694 of *Lecture Notes in Computer Science*, pp. 446–461. Springer, 2014.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2018.

Cody A. Coleman, Deepak Narayanan, Daniel Kang, Tian Zhao, Jian Zhang, Luigi Nardi, Peter Bailis, Kunle Olukotun, Chris R, and Matei Zaharia. Stanford DAWN Deep Learning Benchmark (DAWNBench), 2018.

Ekin Dogus Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. *CoRR*, abs/1805.09501, 2018.

Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning*, 2014.

Weifeng Ge and Yizhou Yu. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Ross Girshick. Fast r-cnn. In *International Conference on Computer Vision*, pp. 1440–1448, 2015.

Kaiming He, Georgia Gkioxari, Piotr Dollr, and Ross Girshick. Mask r-cnn. In *International Conference on Computer Vision 2017*, pp. 2980–2988, 2017.

Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara Balan, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3296–3297, 2017.

Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better? *CoRR*, abs/1805.08974, 2018.

Jonathan Krause, Jia Deng, Michael Stark, and Li Fei-Fei. Collecting a large-scale dataset of fine-grained cars. In *Second Workshop on Fine-Grained Visual Categorization (FGVC2)*, 2013.

Jonathan Krause, Benjamin Sapp, Andrew Howard, Howard Zhou, Alexander Toshev, Tom Duerig, James Philbin, and Li Fei-Fei. The unreasonable effectiveness of noisy data for fine-grained recognition. In *ECCV*, 2016.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Computer Science Department, University of Toronto, Tech. Rep*, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

Dhruv Kumar Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. *CoRR*, abs/1805.00932, 2018.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *CoRR*, abs/1306.5151, 2013.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 512–519, 2014.

Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. Regularized evolution for image classifier architecture search. *CoRR*, abs/1802.01548, 2018.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision*, 115(3):211–252, December 2015.

Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 14(1):21–41, 2002.

Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, 2017.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, October 2000.

Amos J Storkey. When training and test sets are different: Characterising learning transfer. In *Dataset Shift in Machine Learning*, pp. 3–28. MIT Press, 2009.

Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems*, pp. 1433–1440, USA, 2007. Curran Associates Inc. ISBN 978-1-60560-352-0.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Commun. ACM*, 59(2):64–73, January 2016.

Xiu-Shen Wei, Chen-Wei Xie, Jianxin Wu, and Chunhua Shen. Mask-cnn: Localizing parts and selecting descriptors for fine-grained bird species categorization. *Pattern Recognition*, 76:704–714, 2018.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, volume 2, pp. 3320–3328, Cambridge, MA, USA, 2014. MIT Press.

Fisher Yu, Dequan Wang, and Trevor Darrell. Deep layer aggregation. In *IEEE Conference on Computer Vision and Pattern Recognition 2018*.

Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Kun Zhang, Bernhard Schlkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pp. 819–827, 2013.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. *CoRR*, abs/1707.07012, 2017.

6 Appendix

6.1 Distribution Matching

We describe the distribution matching methods in detail in this section.

Let us start by assuming that we have a source dataset with 100 examples with three different classes: (A: 10 examples), (B: 40 examples), and (C: 50 examples). Next, consider a scenario where the target dataset has a predicted label distribution over the source label set such that (A: 50%), (B: 30%), and (C: 20%). From this we can examine how to construct a pre-training dataset, say of size 30 examples.

With the same distribution matcher, we sample the examples at a rate proportional to the importance weight computed using the ratio of the two distributions. Hence, (A: $0.5/0.1 = 5$), (B: $0.3/0.4 = 0.75$), (C: $0.2/0.5 = 0.4$). We then adjust this based on the desired pre-training dataset size ($30/100 = 0.3$). Thus, in expectation, this results in the following number of examples per class: A: ($0.3 \times 5 \times 10 = 15$), (B: $0.3 \times 0.75 \times 40 = 9$), (C: $0.3 \times 0.4 \times 50 = 6$).

For the elastic distribution matcher, we avoid selecting each example more than once. In order to keep the distribution as similar to the desired one, we consider a sequential approach: we start with the class with the highest importance weight, in this case A, and exhaust the 10 samples available. Next, we recursively consider sampling a dataset of the remaining desired examples ($30 - 10 = 20$) from the rest of the classes. Thus, we obtain the following number of examples per class: $(A : 10)$, $(B : 12)$, $(C : 8)$. In Table 5, we show how the sampling distribution turns out to differ for the CIFAR-10 dataset when using ImageNet as source pre-training data.

6.2 Understanding the importance of the pre-training distribution

To further understand the importance of the distribution, we created 3 ANON subsets of the same size but with different distributions from top 4,000 matched labels on Oxford-IIIT Pets. The uniform distribution experiment tells us how important it is to select relevant images, and the reverse distribution experiment tells us the importance of choosing the *weighted* distribution that matches the target dataset.

We observed that their transfer performance aligns well with the degree that their distribution matches the distribution of target dataset (Table 6).

Table 5: Comparison of ImageNet labels statistics between the same distribution matcher and elastic distribution matcher for CIFAR-10.

#	ImageNet Label	# Examples	Sample Rate %	Sample Rate %
		in ImageNet	(Same Distribution)	(Elastic Match)
1	Moving van	1159	576.95%	100.00%
2	Sorrel	1300	370.50%	100.00%
3	Container ship	1300	212.25%	100.00%
4	Airliner	1300	189.85%	100.00%
5	Amphibian	1300	149.87%	100.00%
6	Thresher	1300	143.89%	100.00%
7	Hartebeest	1300	138.48%	100.00%
8	Japanese spaniel	771	201.66%	100.00%
9	Chain saw	1194	112.37%	100.00%
10	Fox squirrel	1206	110.68%	100.00%
11	Convertible	1300	92.86%	100.00%
12	Milk can	1097	95.20%	100.00%
13	Gazelle	1300	79.34%	95.39%
14	Speedboat	1300	79.25%	95.28%
15	Rock beauty	969	86.11%	100.00%
16	Yawl	1206	66.16%	79.55%
17	Can opener	1300	61.23%	73.62%
18	Walker hound	1025	69.78%	83.90%
19	Persian cat	1300	54.64%	65.69%
20	Brambling	1300	53.09%	63.83%
...				
991	Toilet seat	1300	0.36%	0.43%
992	Gown	1300	0.35%	0.42%
993	Cup	1300	0.35%	0.42%
994	Porcupine	1300	0.34%	0.41%
995	Pencil box	1300	0.34%	0.41%
996	Miniskirt	1300	0.32%	0.39%
997	Strainer	1300	0.29%	0.35%
998	Notebook	1300	0.25%	0.30%
999	Radio	1300	0.24%	0.29%
1000	Suit	1300	0.23%	0.28%

Table 6: Transfer performance on Oxford-IIIT Pets from ANON subsets of the same size (80M) but with different distribution: *Same* is the distribution of source dataset labels predicted from target dataset examples, *Uniform* is a uniform distribution on all selected labels, and *Reverse* is a distribution obtained by swapping the sampling rates between the highest and the lowest labels from the *Same* distribution.

Target Distribution	# Unique Examples	Transfer Performance
Same	14.9M	97.0
Uniform	54.7M	95.3
Reverse	10.4M	84.9