2021 IEEE/CVF International Conference on Computer Vision (ICCV) | 978-1-6654-2812-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCV48922.2021.00931

Few-shot Image Classification: Just Use a Library of Pre-trained Feature Extractors and a Simple Classifier

Arkabandhu Chowdhury[1], Mingchao Jiang[1], Swarat Chaudhuri[2], and Chris Jermaine[1]

[1]Rice University, [2]University of Texas, Austin

Abstract

Recent papers have suggested that transfer learning can outperform sophisticated meta-learning methods for few-shot image classification. We take this hypothesis to its logical conclusion, and suggest the use of an ensemble of high-quality, pre-trained feature extractors for few-shot image classification. We show experimentally that a library of pre-trained feature extractors combined with a simple feed-forward network learned with an L2-regularizer can be an excellent option for solving cross-domain few-shot image classification. Our experimental results suggest that this simple approach far outperforms several well-established meta-learning algorithms.

1. Introduction

There has been a lot of recent interest in few-shot image classification [5, 15, 17, 6, 7, 12, 19, 16]. Various papers have explored different formulations of the problem, but in one general formulation, we are given a data set $\mathcal{D}_{trn}$ of (image, label) pairs sampled from a distribution $\mathbb{P}_{trn}$. The goal is to devise a method that uses $\mathcal{D}_{trn}$ to learn a function f that is *itself* a few-shot learner. The few-shot learner f takes as input a new labeled data set $\mathcal{D}_{few}$ consisting of a set of samples from a new distribution $\mathbb{P}_{few} \neq \mathbb{P}_{trn}$. f then returns a classification function g, which is targeted at classifying samples from the distribution $\mathbb{P}_{few}$.

The process of learning f is often referred to as *meta-learning* in the literature. Learning to classify samples from $\mathbb{P}_{few}$ is a "few shot" problem when $\mathcal{D}_{few}$ is small, perhaps having only one example for each class produced by $\mathbb{P}_{few}$. In the most difficult and generally applicable variant of the few-shot problem—which we consider in this paper—$\mathbb{P}_{few}$ has no known relationship to $\mathbb{P}_{trn}$ (this is "cross-domain" few-shot learning) and $\mathcal{D}_{few}$ is not available while learning f. Thus, the meta-learning process has no access to information about the eventual application. The only information we have about $\mathbb{P}_{few}$ is the set $\mathcal{D}_{few}$, and this is available only when constructing g. We cannot, for example, choose any hyperparameters controlling the learning of g using information not extracted from $\mathcal{D}_{few}$.

Our goal is to devise a learner f that works well, out-of-the-box, on virtually any new distribution $\mathbb{P}_{few}$. We argue that in such a scenario, developing novel meta-learning methods to learn f from scratch on given $\mathcal{D}_{trn}$ may not be the most productive direction of inquiry. Because no information about $\mathbb{P}_{few}$ is available during meta-learning, it makes sense to choose a $\mathcal{D}_{trn}$ that has many different types of images, so it is likely to contain some images with features similar to those produced by $\mathbb{P}_{few}$, whatever form this distribution takes. Fortunately, in computer image classification, the standard benchmark data set is now ILSVRC2012, a 1000-class version of the full ImageNet [22]. ILSVRC2012 consists of a wide variety of images, and it has become quite standard for researchers who design and train new image classifiers to publish classifiers trained on ILSVRC2012. Such published artifacts represent thousands of hours of work by researchers who are well-versed in the "black art" of wringing every last percent of accuracy out of a deep CNN. Instead of developing new meta-learning methods, it may be more productive to simply fix $\mathcal{D}_{trn}$ = ILSVRC2012, and attempt to leverage all of the effort that has gone into learning deep CNNs over ILSVRC2012, by using those CNNs as the basis for the few-shot learner f. As other, even more wide-ranging and difficult benchmark data sets become prevalent (such as the full, 20,000+ class ImageNet), high-quality classifiers trained using that data set may be preferred.

We first show that it is possible to use any of a number of published, high-quality, deep CNNs, learned over ILSVRC2012, as the basis for a few-shot learner that significantly outperforms state-of-the art methods. The way to do this is embarrassingly simple: remove the classifier from the top of the deep CNN, fix the weights of the remaining deep feature extractor, and replace the classifier with a simple MLP that is trained using L_2 regularization to prevent over-fitting. We call these "library-based" learners because

978-1-6654-2812-5/21/$31.00 ©2021 IEEE
DOI 10.1109/ICCV48922.2021.00931

they are based on standard, published feature extractors.

Next, we ask: if a published deep CNN can be used to produce a state-of-the-art, few-shot learner, can we produce an even higher-quality few-shot learner by combining together *many* high-quality, deep CNNs? We call such a learner a "full library" learner.

Then, we note that other researchers have suggested the utility of re-using high-quality, pre-trained feature extractors for few-shot image classification. In particular, the authors of the "Big Transfer" paper [13] argue that a very large network trained on a huge data set (the JFT-300M data set [26], with 300 million images) can power an exceptionally accurate transfer-learning-based few-shot learning. Unfortunately, the authors have not made their largest JFT-300M-trained network public, and so we cannot experiment with it directly. However, they *have* made public several versions of their "Big Transfer" network, trained on the full, 20,000+ class ImageNet benchmark public. Interestingly, we find that "big" may not be as important as "diverse": a single, few-shot classifier comprised of many different high-quality ILSVRC2012-trained deep CNNs seems to be a better option than a single few-shot classifier built on top of any of the Google-trained CNNs. Finally, we investigate why a full library learner works so well. We postulate two reasons for this. First, having a very large number of features ($>$ 10,000) does not seem to be a problem for few-shot learning. Second, there seems to be strength in diversity, in the sense that different CNNs appear useful for different tasks.

2. High Accuracy of Library-Based Learners

2.1. Designing a Library-Based Learner

We begin by asking: what if we eschew advanced meta-learning methods, and instead simply use a very high-quality deep CNN pulled from a library, trained on the ILSVRC2012 data set, as the basis for a few-shot classifier?

Specifically, we are given a high-quality, pre-trained deep CNN, from a library of pre-trained networks; we take the CNN as-is, but remove the topmost layers used for classification. This results in a function that takes an image, and returns an embedding. We then use that embedding to build a classifier in an elementary fashion: we feed the embedding into a multi-layer perceptron with a single hidden layer; a softmax is used to produce the final classification. Given a few-shot classification problem, two weight matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ are learned; the first connecting the embedding to the hidden layer, the second connecting the hidden layer to the softmax. To prevent over-fitting during training, simple L_2 regularization is used on the weight matrices.

2.2. Evaluation

To evaluate this very simple few-shot learner, we first identify nine, high-quality deep CNNs with published models, trained on ILSVRC2012: ResNet18, ResNet34, ResNet50, ResNet101, ResNet152 (all of the ResNet implementations are the ones from the original ResNet designers [8]), DenseNet121, DenseNet161, DenseNet169, and DenseNet201 (all of the DenseNet implementations are also from the original designers [10]).

Our goal is to produce an "out-of-the-box" few-shot learner that can be used on any (very small) training set $\mathcal{D}_{few}$ without additional data or knowledge of the underlying distribution. We are very careful not to allow validation or parameter tuning on testing data domains, so all parameters and settings need to be chosen apriori. If it is possible to build such a few-shot learner, it would be the most widely applicable: simply produce a few training images for each class, apply the learner. Thus, we perform an extensive hyper-parameter search, solely using the Caltech-UCSD Birds 200 set [32] as a validation data set, and then use the best hyperparameters from that data set in all of our experiments. Hyperparameters considered were learning rate, number of training epochs, regularization penalty weight, the number of neurons in the MLP hidden layer, and whether to drop the hidden layer altogether. A separate hyper-parameter search was used for 5-way, 20-way, and 40-way classification.

We then test the resulting few-shot learner—one learner per deep CNN—on eight different data sets, FGVC-Aircraft [18], FC100 [21], Omniglot [14], Traffic Sign [9], FGCVx Fungi [24], Quick Draw [11], and VGG Flower [20]. To evaluate a few-shot learner on a data set, for an "m-way n-shot" classification problem, we randomly select m different classes, and then randomly select n images from each class for training; the remainder are used for testing. As this is "out-of-the-box" learning, no validation is allowed.

We performed this evaluation for m in $\{5, 20, 40\}$ and n in $\{1, 5\}$. Due to space constraints, the full results are presented as supplementary material, and we give only a synopsis here (Table 2 and Table 3). In Table 1, we show the best and worst accuracy achieved across the 9 learners, for each of the 8 data sets, for m in $\{5, 20, 40\}$ and $n = 1$.

To give the reader an idea of how this accuracy compares to the state-of-the-art, we compare these results with a number of few-shot learners from the literature. We compare against Baseline and Baseline++ [1], MAML [6], MatchingNet [30], ProtoNet [25], RelationNet [28], Meta-transfer [27], FEAT [33], and SUR [4]. When a deep CNN classifier must be chosen for any of these methods, we use a ResNet18. For methods that require a pre-trained CNN (FEAT, Meta-transfer, and SUR), we use the ResNet18 trained by the ResNet designers [8]. Lest the reader be concerned that we chose the worst option (ResNet18), we point out that of the library-based few-shot learners, on the 5-way, 5-shot problem ResNet18 gave the best accuracy out of all of the ResNet-based learners for two of the data sets (see Ta-

	Aircraft	FC100	Omniglot	Texture	Traffic	Fungi	Quick Draw	VGG Flower
				5-way, 1-shot				
Worst	40.9 ± 0.9 RN18	50.8 ± 0.9 DN121	77.2±0.9 RN152	59.1 ± 0.9 DN169	55.5 ± 0.8 RN152	53.0 ± 0.9 DN201	57.3 ± 0.9 RN101	79.7 ± 0.8 RN18
Best	46.2 ± 1.0 DN161	61.2±0.9 RN152	86.5 ± 0.7 DN121	65.1 ± 0.9 RN101	66.6 ± 0.9 DN201	56.6 ± 0.9 DN121	62.8 ± 0.9 RN18	83.5 ± 0.8 DN161
				20-way, 1-shot				
Worst	20.1 ± 0.3 RN101	27.8 ± 0.4 DN121	56.2 ± 0.5 RN101	38.0 ± 0.4 RN18	29.7 ± 0.3 RN101	31.7 ± 0.4 RN101	33.2 ± 0.5 RN101	62.4 ± 0.5 RN101
Best	24.3 ± 0.3 DN161	36.4 ± 0.4 RN152	69.1 ± 0.5 DN121	42.5 ± 0.4 RN152	38.5 ± 0.4 DN201	33.9 ± 0.5 DN161	39.5± 0.5 DN201	70.0 ± 0.5 DN161
				40-way, 1-shot				
Worst	14.2 ± 0.2 RN34	19.6 ± 0.2 RN18	47.3 ± 0.3 RN152	28.9 ± 0.2 RN18	22.2 ± 0.2 RN152	23.7 ± 0.3 RN34	26.4 ± 0.3 RN152	53.1 ± 0.3 RN34
Best	17.4 ± 0.2 DN161	27.2 ± 0.3 RN152	61.6 ± 0.3 DN201	33.2 ± 0.3 DN152	29.5 ± 0.2 DN201	26.8 ± 0.3 DN161	31.2 ± 0.3 DN201	62.8 ± 0.3 DN161

Table 1: Accuracy obtained using library deep CNNs for few-shot learning.

	Aircraft	FC100	Omniglot	Texture	Traffic	Fungi	Quick Draw	VGG Flower
Baseline	47.6 ± 0.7	66.9 ± 0.7	96.5 ± 0.2	62.5 ± 0.7	82.1 ± 0.7	57.6 ± 1.7	75.6 ± 0.7	90.9 ± 0.5
Baseline++	40.9 ± 0.7	59.8 ± 0.8	90.6 ± 0.4	59.9 ± 0.7	79.6 ± 0.8	54.1 ± 1.7	68.4 ± 0.7	83.1 ± 0.7
MAML	33.1 ± 0.6	62.0 ± 0.8	82.6 ± 0.7	56.9 ± 0.8	67.4 ± 0.9	48.3 ± 1.8	77.5 ± 0.8	78.0 ± 0.7
MatchingNet	33.5 ± 0.6	59.4 ± 0.8	89.7 ± 0.5	54.7 ± 0.7	73.7 ± 0.8	55.7 ± 1.7	70.4 ± 0.8	74.2 ± 0.8
ProtoNet	41.5 ± 0.7	64.7 ± 0.8	95.5 ± 0.3	62.9 ± 0.7	75.0 ± 0.8	53.1 ± 1.8	74.9 ± 0.7	86.7 ± 0.6
RelationNet	37.5 ± 0.7	64.8 ± 0.8	91.2 ± 0.4	60.0 ± 0.7	68.6 ± 0.8	58.7 ± 1.8	71.9 ± 0.7	80.6 ± 0.7
Meta-transfer	46.2 ± 0.7	75.7 ± 0.8	93.5 ± 0.4	70.5 ± 0.7	80.0 ± 0.8	66.1 ± 0.8	77.7 ± 0.7	90.5 ± 0.6
FEAT	46.2 ± 0.9	60.5 ± 0.8	85.3 ± 0.7	70.7 ± 0.7	70.5 ± 0.8	67.3 ± 1.7	69.9 ± 0.7	92.1 ± 0.4
SUR	45.2 ± 0.8	67.2 ± 1.0	**98.7 ± 0.1**	59.6 ± 0.7	70.6 ± 0.8	60.0 ± 1.8	73.5 ± 0.7	90.8 ± 0.5
Worst library-based	61.0 ± 0.9 RN34	71.9 ± 0.8 DN121	94.0 ± 0.4 RN152	79.3 ± 0.6 RN18	78.8 ± 0.7 RN152	77.1 ± 0.8 RN34	77.8 ± 0.7 RN152	95.3 ± 0.4 RN34
Best library-based	**66.0 ± 0.9** DN161	**80.0 ± 0.6** RN152	96.7 ± 0.2 DN201	**83.4 ± 0.6** DN161	**85.3 ± 0.7** DN201	**79.1 ± 0.7** DN121	**81.8 ± 0.6** DN201	**96.8 ± 0.3** DN161

Table 2: Comparing competitive methods with the simple library-based learners, on the 5-way, 5-shot problem.

ble 4). Further, these competitive methods tend to be quite expensive to train—MAML, for example, requires running gradient descent over a gradient descent—and for such a method, the shallower ResNet18 is a much more reasonable choice than the deeper models (even using a ResNet18, we could not successfully train first-order MAML for 5-shot, 40-way classification, due to memory constraints).

For the competitive methods (other than SUR) we follow the same procedure as was used for the library-based few-shot classifiers: any training that is necessary is performed on the ILSVRC2012 data set, and hyperparameter validation is performed using the Caltech-UCSD Birds 200 data set. Each method is then used without further tuning on the remaining eight data sets. To evaluate SUR on data set X, we use feature extractors trained on the data sets in $\{$Omniglot, Aircraft, Birds, Texture, Quickdraw, Flowers, Fungi,and ILSVRC$\} - X$. Traffic Sign and FC100 datasets are reserved for testing only.

A comparison of each of these competitive methods with the the best and worst-performing library-based learners on the 5-way, 5-shot learning problem is shown in Table 2; a comparison on 20-way, 5-shot learning in Table 3. A more complete set of results is in the supplementary material.

2.3. Discussion

There are a few key takeaways from these results. The best library-based learner *always* beat every one of the other methods tested, with the only exception of SUR when test-

	Aircraft	FC100	Omniglot	Texture	Traffic	Fungi	Quick Draw	VGG Flower
Baseline	24.2 ± 0.3	40.0 ± 0.4	87.5 ± 0.3	37.0 ± 0.3	59.9 ± 0.4	32.5 ± 0.8	52.8 ± 0.4	76.7 ± 0.4
Baseline++	18.4 ± 0.3	33.8 ± 0.3	76.2 ± 0.2	34.8 ± 0.3	55.3 ± 0.4	28.2 ± 0.8	45.5 ± 0.4	64.0 ± 0.4
MAML	11.8 ± 0.2	25.7 ± 0.3	46.5 ± 0.4	21.9 ± 0.3	27.0 ± 0.3	17.3 ± 0.7	30.7 ± 0.3	32.9 ± 0.3
MatchingNet	11.9 ± 0.2	31.6 ± 0.3	64.6 ± 0.6	31.6 ± 0.3	46.5 ± 0.4	28.1 ± 0.8	41.2 ± 0.4	53.7 ± 0.5
ProtoNet	22.1 ± 0.3	38.9 ± 0.4	88.1 ± 0.2	38.9 ± 0.3	46.9 ± 0.4	33.0 ± 0.9	33.0 ± 0.9	70.9 ± 0.4
RelationNet	17.1 ± 0.3	39.1 ± 0.4	79.7 ± 0.3	32.1 ± 0.3	41.9 ± 0.4	27.8 ± 0.8	47.5 ± 0.4	62.5 ± 0.4
Meta-transfer	19.1 ± 0.3	48.0 ± 0.4	75.3 ± 0.4	45.5 ± 0.4	52.0 ± 0.4	38.6 ± 0.5	52.6 ± 0.4	74.0 ± 0.4
FEAT	23.1 ± 0.5	34.4 ± 0.5	66.4 ± 0.6	47.5 ± 0.6	43.4 ± 0.6	43.9 ± 0.8	47.0 ± 0.6	80.5 ± 0.5
SUR	21.8 ± 0.3	42.9 ± 0.5	$\mathbf{96.3 \pm 0.1}$	35.5 ± 0.4	46.7 ± 0.4	34.4 ± 0.9	54.2 ± 0.4	77.1 ± 0.4
Worst library-based	37.5 ± 0.4 RN18	47.1 ± 0.4 RN18	84.3 ± 0.3 RN101	58.7 ± 0.4 RN18	55.9 ± 0.4 RN152	56.1 ± 0.5 RN34	57.2 ± 0.4 RN101	86.8 ± 0.3 RN101
Best library-based	$\mathbf{44.6 \pm 0.4}$ DN161	$\mathbf{58.8 \pm 0.4}$ RN152	92.0 ± 0.2 DN201	$\mathbf{65.1 \pm 0.4}$ DN161	$\mathbf{66.0 \pm 0.4}$ DN201	$\mathbf{60.8 \pm 0.5}$ DN161	$\mathbf{63.9 \pm 0.4}$ DN161	$\mathbf{91.6 \pm 0.2}$ DN161

Table 3: Comparing competitive methods with the simple library-based learners, on the 20-way, 5-shot problem.

ing on Omniglot data set. For the other data sets, the gap only grows as the number of ways increases. In fact, for the 20-way problem, the *worst* library-based learner always beat all of the other methods tested (except SUR on Omniglot). The gap can be quite dramatic, especially on the 20-way problem. The best non-transfer based few-shot learner (MAML, Proto Nets, Relation Nets, and Matching Nets fall in this category) was far worse than even the worst library-based learner: 39% accuracy for Proto Nets vs. 59% accuracy for a classifier based on RestNet18 on Texture, 48% accuracy for Relation Nets vs. 57% accuracy for classifier based on ResNet101 on Quick Draw. There have been a large number of non-transfer-based methods proposed in the literature (with a lot of focus on improving MAML in particular [6, 7, 12, 19, 16]) but the gap between MAML and the library-based classifiers is very large.

We also note that of the rest non-library methods, Meta-transfer, Baseline, and FEAT were generally the best. We note that Meta-transfer, Baseline, and FEAT use the pre-trained ResNet18 without modification. This tends to support the hypothesis at the heart of this paper: starting with a state-of-the-art feature extractor, trained by experts, may be the most important decision in few-shot learning.

3. A Simple Full Library Classifier

3.1. Extreme Variation in Few-Shot Quality

There is not a clear pattern to which of the library-based classifiers tends to perform best, and which tends to perform worst. Consider the complete set of results, over all of the library-based few-shot learners, for the 5-way, 5-shot problem, shown in Table 4. For "out-of-the-box" use, where no validation data are available, it is very difficult to see any sort of pattern that might help with picking a particular library-based classifier. The DenseNet variations sometimes do better than ResNet (on the Aircraft data set, for example), but sometimes they do worse than the ResNet variations (on the FC100 data set, for example). And within a family, it is unclear which library-based CNN to use. As mentioned before, ResNet18 provides the best ResNet-based few-shot learner for two of the data sets, but it forms the basis of the worst ResNet-based learner on another two.

3.2. Combining Library-Based Learners

Given the relatively high variance in accuracies obtained using the library deep CNNs across various data sets, it is natural to ask: Is it perhaps possible to use all of these library feature extractors in concert with one another, to develop a few-shot learner which consistently does as well as (or even better then) the best library CNN?

Given the lack of training data in few-shot learning, the first idea that one might consider is some simple variant of ensembling: given a few-shot learning problem, simply train a separate neural network on top of each of the deep CNNs, and then use a majority vote at classification time (hard ensembling) or we can average the class weights at classification time (soft ensembling).

Another option is to take all of the library deep CNNs together, and view them as a single feature extractor. Using the nine models considered thus far in this paper in this way results in 13,984 features. We then train an MLP on top of this, using L_2 regularization. Again using the Caltech-UCSD Birds 200 data set for validation, we perform hyperparameter search to build a few-shot learner for 5-way, 20-way, and 40-way classification problems.

We test both of these options over the eight test sets, and give a synopsis of the results in Table 5. We find that across all 24 test cases (eight data sets, three classification tasks), the best single learner was *never* able to beat the best method that used all nine deep CNNs. All of the tested

	Aircraft	FC100	Omniglot	Texture	Traffic	Fungi	Quick Draw	VGG Flower
DenseNet121	64.7 ± 0.9	71.9 ± 0.8	$\mathbf{96.7 \pm 0.3}$	82.1 ± 0.6	$\mathbf{85.0 \pm 0.7}$	$\mathbf{79.1 \pm 0.7}$	81.3 ± 0.7	96.0 ± 0.4
DenseNet161	$\mathbf{66.0 \pm 0.9}$	73.7 ± 0.7	96.6 ± 0.3	$\mathbf{83.4 \pm 0.6}$	83.9 ± 0.7	78.4 ± 0.8	81.3 ± 0.6	$\mathbf{96.8 \pm 0.3}$
DenseNet169	63.6 ± 0.9	73.5 ± 0.7	95.0 ± 0.3	82.3 ± 0.6	84.6 ± 0.7	78.4 ± 0.8	80.6 ± 0.7	96.1 ± 0.3
DenseNet201	62.6 ± 0.9	75.1 ± 0.7	$\mathbf{96.7 \pm 0.6}$	83.2 ± 0.6	85.3 ± 0.7	78.0 ± 0.7	$\mathbf{81.8 \pm 0.6}$	96.5 ± 0.3
ResNet18	61.2 ± 0.9	72.1 ± 0.8	95.4 ± 0.3	79.3 ± 0.6	83.2 ± 0.7	77.7 ± 0.7	81.7 ± 0.6	95.3 ± 0.4
ResNet34	61.0 ± 0.9	76.2 ± 0.7	94.9 ± 0.3	82.5 ± 0.6	81.3 ± 0.7	77.1 ± 0.7	79.7 ± 0.6	95.3 ± 0.4
ResNet50	62.3 ± 0.9	73.9 ± 0.8	94.3 ± 0.4	83.2 ± 0.6	79.4 ± 0.8	77.9 ± 0.8	78.1 ± 0.8	95.6 ± 0.4
ResNet101	62.4 ± 0.9	79.2 ± 0.7	95.5 ± 0.3	82.8 ± 0.6	81.3 ± 0.7	78.6 ± 0.8	78.6 ± 0.7	95.8 ± 0.3
ResNet152	61.7 ± 1.0	$\mathbf{80.0 \pm 0.6}$	94.0 ± 0.4	83.0 ± 0.6	78.8 ± 0.7	79.0 ± 0.8	77.8 ± 0.7	95.6 ± 0.3

Table 4: Complete results for library-based few-shot learners on the 5-way, 5-shot problem.

methods had similar performance on the 5-way classification task, though the best single learner was generally a bit worse than the other methods.

Where the difference becomes more obvious is on the classification tasks with more categories. For the 20-way and 40-way problems, the two ensemble-based methods had a small but consistent drop in accuracy, and building a single network on top of all nine deep CNNs is clearly the best. This may be somewhat surprising; for the 40-way problem, hyperparameter search on the Caltech-UCSD Birds 200 data set settled on a single network with 1024 neurons in a hidden layer; this means that more than 14 million parameters must be learned over just 200 images. Clearly, the neural network is massively over-parameterized, and yet the accuracies obtained are remarkable, with over 90% accuracy on two of the data sets.

4. Data vs. Diversity: Who Wins?

The results from the last section clearly show that an MLP learned on top of a library of high-quality, deep CNNs makes for an excellent, general-purpose, few-shot learner. This leaves open the question. When basing a few-shot learner on a fairly simple, transfer-based approach, which is more important, diversity or size, when constructing the few-shot learner? That is, is it better to have a large variety of deep CNNs to combine together, each of which is trained on a smaller $\mathcal{D}_{trn}$, or is it preferable to base a few-shot learner on a single, deep CNN that has been trained on a larger and more diverse $\mathcal{D}_{trn}$?

To answer this question, we compare the single MLP built upon all nine of the library deep CNNs with an MLP built upon a high-quality deep CNN that was *itself* constructed upon an even larger data set: the full ImageNet data set, with more than 20,000 categories. High-quality, publicly available deep CNNs for this data set are rare, but Google recently released a set of deep CNNs trained on the full ImageNet, specifically for use in transfer learning[13]. We consider three of their deep CNNs. Each is a ResNet: BiT-ResNet-101-3 ("BiT" stands for "Big Transfer"; "101-3" is a ResNet101 that has $3X$ the width of a standard ResNet), BiT-ResNet-152-4, and BiT-ResNet-50-1.

For each of these Big Transfer models, we perform a full hyperparameter search using the Caltech-UCSD Birds data set for validation, as we did for each of the deep CNNs trained on ILSVRC2012. Interestingly, the Google models tend to do better with a much larger L_2 regularization parameter weight (0.5 to 0.7) compared to the other deep CNNs trained on ILSVRC2012 (which typically performed best on the validation set using a weight of around 0.1). Results are shown in Table 6.

The headline finding is that the single model utilizing a library of nine, ILSVRC2012-trained CNNs, is the best model. It did not not always perform the best on each data set. In fact, on four of the data sets (FC100, Texture, Fungi, and VGG Flower) at least one of the Google Big Transfer models outperformed the single model consisting of nine ILSVRC2012-trained CNNs.

In each of the those cases, the performance was comparable across models, except, perhaps, for the VGG Flower data set, where the best Big Transfer models always obtained more than 99% accuracy. However, on the other data sets (Aircraft, Omniglot, Trafic Sign, and QuickDraw) the combined model far outperformed any of the Big Transfer models. The gap was often significant, and the combined model outperformed the best Big Transfer-based model by an average of more than 11% for the 40-way task.

It was also interesting that while the Big Transfer models were generally better than the ILSVRC2012-trained library CNNs, this varied across data sets. On the Aircraft and Omniglot data sets, for example, even the best *individual* ILSVRC2012-trained library CNNs outperformed the Big Transfer models.

All of this taken together seems to suggest that, when building a transfer-based few-shot learner, having a large library of deep CNNs is at least as important—and likely *more* important—than having access to a CNN that was trained on a very large $\mathcal{D}_{trn}$.

	Aircraft	FC100	Omniglot	Texture	Traffic	Fungi	Quick Draw	VGG Flower
				5-way, 5-shot				
Full Library	**68.9 ± 0.9**	79.1 ± 0.8	97.5 ± 0.3	85.3 ± 0.6	**85.8 ± 0.7**	81.2 ± 0.8	**84.2 ± 0.6**	97.4 ± 0.3
Hard Ensemble	67.8 ± 0.9	79.9 ± 0.7	97.8 ± 0.2	85.4 ± 0.5	85.2 ± 0.7	**82.1 ± 0.7**	83.5 ± 0.6	97.7 ± 0.2
Soft Ensemble	68.4 ± 0.9	**80.5 ± 0.6**	**98.0 ± 0.2**	**85.7 ± 0.6**	85.2 ± 0.7	82.0 ± 0.7	84.1 ± 0.5	**97.9 ± 0.2**
Best Single	66.0 ± 0.9	80.0 ± 0.6	96.7 ± 0.2	83.4 ± 0.6	85.2 ± 0.7	79.1 ± 0.7	81.8 ± 0.6	96.8 ± 0.3
				20-way, 5-shot				
Full Library	**49.5 ± 0.4**	**61.6 ± 0.4**	**95.4 ± 0.2**	**68.5 ± 0.4**	**70.4 ± 0.4**	**65.5 ± 0.5**	**69.4 ± 0.4**	**94.3 ± 0.2**
Hard Ensemble	46.6 ± 0.4	60.1 ± 0.4	94.5 ± 0.2	67.8 ± 0.4	67.8 ± 0.4	64.4 ± 0.4	67.9 ± 0.4	93.5 ± 0.2
Soft Ensemble	47.5 ± 0.4	60.7 ± 0.4	94.9 ± 0.2	68.2 ± 0.4	68.3 ± 0.4	64.4 ± 0.4	68.8 ± 0.4	93.7 ± 0.2
Best Single	44.6 ± 0.4	58.8 ± 0.4	92.0 ± 0.2	65.1 ± 0.4	66.0 ± 0.4	60.8 ± 0.5	63.9 ± 0.4	91.6 ± 0.2
				40-way, 5-shot				
Full Library	**41.2 ± 0.3**	**51.8 ± 0.2**	**93.2 ± 0.1**	**59.3 ± 0.2**	**62.7 ± 0.2**	**57.6 ± 0.3**	**60.8 ± 0.3**	**91.9 ± 0.2**
Hard Ensemble	38.0 ± 0.3	50.2 ± 0.2	92.1 ± 0.1	58.5 ± 0.2	59.6 ± 0.2	55.6 ± 0.3	59.3 ± 0.3	90.6 ± 0.2
Soft Ensemble	39.0 ± 0.3	51.2 ± 0.3	92.5 ± 0.1	59.0 ± 0.2	60.2 ± 0.2	56.5 ± 0.3	60.1 ± 0.3	91.1 ± 0.2
Best Single	35.9 ± 0.2	48.2 ± 0.3	89.4 ± 0.2	55.4 ± 0.2	57.5 ± 0.2	52.1 ± 0.3	55.5 ± 0.3	88.9 ± 0.2

Table 5: Accuracy obtained using all nine library CNNs as the basis for a few-shot learner.

5. Why Does This Work?

5.1. Few-Shot Fine-Tuning Is Surprisingly Easy

We turn our attention to asking: why does using a full library of pre-trained feature extractors seem to work so well? One reason is that fine-tuning appears to be very easy with a library of pre-trained features. Consider the following simple experiment, designed to test whether the number of training points has much effect on the learned model.

We choose a large number of 40-way, problems, over all eight of our benchmark data sets, and train two classifiers for each. Both classifiers use all of the 13,984 features provided by the nine library feature extractors. However, the first classifier is trained as a one-shot classifier, and the second classifier is trained using all of the available data in the data set. Our goal is to see whether the sets of learned weights have a strong correspondence across the two learners; if they do, it is evidence that the number of training points has a relatively small effect. Note that in a neural network with a hidden layer consisting of hundreds or thousands of neurons, there are likely to be a large number of learned weights that are of equal quality; in fact, simply permuting the neurons in the hidden layer results in a model that is functionally identical, but that has very different weight matrices. Thus, we do not use a hidden layer in either classifier, and instead use a softmax directly on top of an output layer that linearly combines the input features. Both the one-shot and full-data classifiers were learned without regularization.

Over each of the eight benchmark data sets, for each feature, we take the L_1 norm of all of the 40 weights associated with the feature; this serves as an indication of the importance of the feature. We compute the Pearson correlation coefficient for each of the 13,984 norms obtained using the models learned for the 1-shot and full data classifiers. These correlations are shown in Table 7.

What we see is a remarkably high correlation between the two sets of learned weights; above 80% correlation in every case, except for the Traffic Sign data set. However, even in the case of the Traffic Sign data set, there was a weak correlation (the Traffic Sign data is a bit of an outlier in other ways as well, as we will see subsequently).

This would seem to indicate that there is a strong signal with only one data point per class, as the learned weights do not differ much when they are learned using the whole data set. This may be one explanation for the surprising accuracy of the full library few-shot learner. Of course, the strong correlation may gloss over significant differences, and more data *does* make a significant difference (consider the difference between the one-shot accuracies in Table 1 and the five-shot accuracies in Table 2.). But even one image per class seems to give a lot of information.

5.2. Different Problems Utilize Different Features

Another reason for the accuracy obtained by the full library method may be that different problem domains seem to utilize different sets of features, and different feature extractors. Having a large library ensures that *some* features relevant to any task are always present.

To investigate this, we construct a large number of 40-way training tasks over each of the various data sets, and for each, we learn a network without a hidden layer on top of all 13,984 features obtained using the library of deep CNNs.

	Aircraft	FC100	Omniglot	Texture	Traffic	Fungi	QDraw	Flower
				5-way, 5-shot				
Full Library	**68.9 ± 0.9**	79.1 ± 0.8	**97.5 ± 0.3**	85.3 ± 0.6	**85.8 ± 0.7**	81.2 ± 0.8	**84.2 ± 0.6**	97.4 ± 0.3
BiT-ResNet-101-3	54.0 ± 1.1	78.6 ± 0.8	82.5 ± 1.2	82.0 ± 0.9	69.2 ± 0.9	81.2 ± 1.2	63.7 ± 1.1	99.6 ± 0.1
BiT-ResNet-152-4	59.5 ± 1.0	**80.9 ± 0.7**	94.2 ± 0.5	**85.4 ± 0.6**	73.3 ± 0.8	**82.5 ± 0.9**	74.8 ± 0.8	**99.7 ± 0.1**
BiT-ResNet-50-1	61.9 ± 1.2	79.0 ± 0.8	87.2 ± 1.1	84.2 ± 0.6	75.6 ± 1.0	**82.5 ± 0.8**	71.5 ± 0.8	99.3 ± 0.2
				20-way, 5-shot				
Full Library	**49.5 ± 0.4**	61.6 ± 0.4	**95.4 ± 0.2**	68.5 ± 0.4	**70.4 ± 0.4**	65.5 ± 0.5	**69.4 ± 0.4**	94.3 ± 0.2
BiT-ResNet-101-3	35.8 ± 0.4	60.4 ± 0.4	87.8 ± 0.3	69.6 ± 0.4	51.1 ± 0.4	68.4 ± 0.5	57.0 ± 0.4	99.3 ± 0.1
BiT-ResNet-152-4	33.5 ± 0.4	**63.4 ± 0.4**	85.4 ± 0.4	**70.9 ± 0.4**	49.2 ± 0.4	68.1 ± 0.5	52.6 ± 0.5	**99.5 ± 0.1**
BiT-ResNet-50-1	39.6 ± 0.4	60.9 ± 0.4	83.9 ± 0.4	66.4 ± 0.4	53.5 ± 0.4	**68.7 ± 0.4**	55.0 ± 0.4	99.1 ± 0.1
				40-way, 5-shot				
Full Library	**41.2 ± 0.3**	51.8 ± 0.2	**93.2 ± 0.1**	59.3 ± 0.2	**62.7 ± 0.2**	57.6 ± 0.3	**60.8 ± 0.3**	91.9 ± 0.2
BiT-ResNet-101-3	24.6 ± 0.3	49.6 ± 0.2	56.4 ± 0.8	**61.5 ± 0.2**	40.2 ± 0.2	**60.3 ± 0.3**	28.9 ± 0.5	99.0 ± 0.1
BiT-ResNet-152-4	25.4 ± 0.2	**53.0 ± 0.3**	81.0 ± 0.3	58.6 ± 0.2	40.0 ± 0.2	53.9 ± 0.4	44.8 ± 0.3	98.8 ± 0.1
BiT-ResNet-50-1	33.0 ± 0.3	48.8 ± 0.3	84.6 ± 0.2	60.0 ± 0.2	46.9 ± 0.2	59.2 ± 0.3	48.3 ± 0.3	**99.0 ± 0.1**

Table 6: Comparing a few-shot learner utilizing the full library of nine ILSVRC2012-trained deep CNNs with the larger CNNs trained on the full ImageNet.

	Aircraft	Birds	FC100	Fungi	Omniglot	Quick Draw	Texture	Traffic	VGG Flower
Correlation	0.95	0.88	0.89	0.86	0.93	0.92	0.80	0.18	0.87

Table 7: Correlation between weight learned using one example per class, and the full data.

Again, we compute the L_1 norm of the set of weights associated with each of the features. This time, however, for each problem we then consider the features whose norms are in the top 20%; these could be considered the features most important to solving the classification task.

For each of the (data set, data set) pairs, we compute the average Jaccard similarity of these top-feature sets. Since each set consists of 20% of the features, if each set of features chosen was completely random, for n features in all, we would expect a Jaccard similarity of $\frac{0.04n}{.2n+.2n-0.04n} = \frac{0.04}{.4-0.04} = 0.111$. Anything greater indicates the sets of features selected are positively correlated; lower indicates a negative correlation. Results are in Figure 1. For each of the nine CNNs, we also compute the fraction of each CNN's features that are in the top 20% when a full library classifier is constructed. These percentages are in Figure 2.

The data in these two plots, along with the previous results, tells a consistent story: there appears to be little correspondence between data sets in terms of the set of features that are chosen as important across data sets. The largest Jaccard value in Figure 1 is less than 0.5 (observed between FC100 and Texture). This shows, in our opinion, a relatively weak correspondence. The Traffic Sign data set, had an average Jaccard of 0.108 across the other eight data sets, which is even lower than the 0.111 that would be expected under a purely random feature selection regime. One might speculate that the lack of correspondence across data sets is evidence for the hypothesis that different tasks tend to use different features, which would explain why it is so effective to use an entire library of deep CNNs for few-shot learning.

Also note that in Figure 2, we tend to see that different deep CNNs contribute "important" features at very different rates, depending on the particular few-shot classification task. This also seems to be evidence that diversity is important. In general, the DenseNets' features are preferred over the ResNets' features, but this is not universal, and there is a lot of variation. It may not be an accident that the three data sets where the selection of features from library CNNs shows the most diversity in Figure 2 (Traffic Sign, Quick Draw, and Omniglot) are the three data sets where the classifier built on top of all nine library CNNs has the largest advantage compared to the few-shot classifiers built on top of the single, "Big Transfer"-based deep CNN.

6. Related Work

While most research on few-shot learning [31, 21, 23] has focused on developing new and sophisticated methods for learning a few-shot learner, there have recently been a

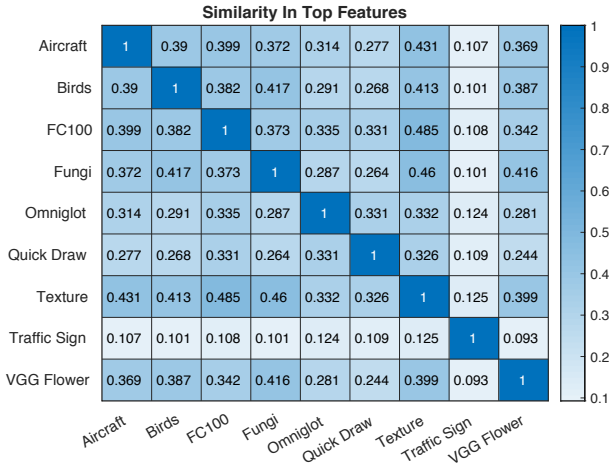

Figure 1: Jaccard similarity of sets of most important features for the various (data set, data set) pairs.

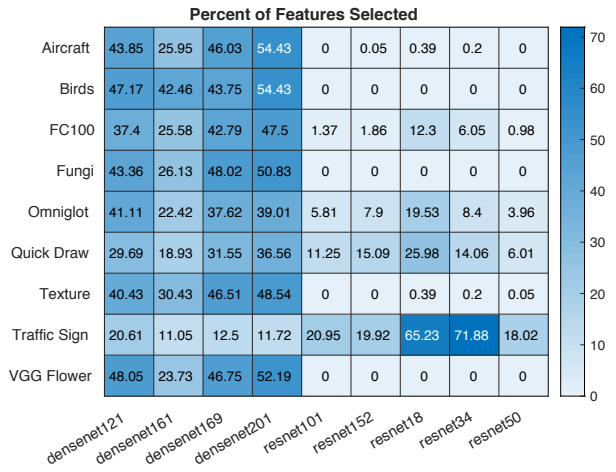

Figure 2: Percent of each deep CNN's features that appear in the top 20% of features, on each data set.

few papers that, like this work, have suggested that transfer-based methods may be the most accurate.

In Section 1, we mentioned Google's "Big Transfer" paper [13]. There, the authors argued that the best approach to few-shot learning is to concentrate on using a high-quality feature extractor rather than a sophisticated meta-learning approach. We agree with this, but also give evidence that training a huge model and a massive data set may not be the only key to few-shot image classification. We found that a library with a wide diversity of high-quality deep CNNs can lead to substantially better accuracy than a single massive CNN, even when that deep CNN is trained on a much larger data set. Tian et al. [29] make a similar argument to the Big Transfer authors, observing that a simple transfer-based approach can outperform a sophisticated meta-learner.

Chen et al. [1] were among the first researchers to point out that simple, transfer-based methods may outperform sophisticated few-shot learners. They proposed Baseline which uses a simple linear classifier on top of a pre-trained deep CNN, and Baseline++ which uses a distance-based classifier. Dhillon et al. [2] also point out the utility of transfer-based methods, and propose a transductive fine-tuner; however, such a method relies on having an appropriately large number of relevant, unlabeled images to perform the transductive learning. Sun et al. [27] consider transfer learning, but where transfer is realized via shifting and scaling of the classification parameters.

Dvornik et al. [4] propose SUR, and argue that diversity in features is best obtained through diversity in data sets. Their method trains many feature extractors, one per data set. We argue that instead, a single high-quality data set is all that is needed. Adding additional feature extractors trained on that data set is the best way to higher accuracy. This may be counter-intuitive, but there is an obvious argument for this: training on less diverse data sets such as Aircraft or VGG Flowers is apt to result in feature extractors that are highly specialized, do not generalize well, and are not particularly useful. The proposed library-based classifier generally outperformed SUR in our experiments.

Dvornik et al. [3] also consider the idea of ensembling for few-shot learning, although their idea is to simultaneously train a number of diverse deep CNNs as feature extractors during meta-learning; adding penalty terms that encourage both diversity and conformity during learning. We propose the simple idea of simply using a set of existing deep CNNs, trained by different groups of engineers.

7. Conclusions

We have examined the idea of using a library of deep CNNs as a basis for a high-quality few-shot learner. We have shown that a learner built on top of a high-quality deep CNN can have remarkable accuracy, and that a learner built upon an entire library of CNNs can significantly outperform a few-shot learner built upon any one deep CNN.

While we conjecture that it will be hard to do better than using a library of high-quality deep CNNs as the basis for a few-shot learner, there are key unanswered questions. First, future work should study if the accuracy continues to improve as the library is made even larger. Also, it may be expensive to feed a test image through a series of nine (or more) deep CNNs. It would be good to know if the computational cost of the library-based approach can be reduced. Finally, it would be good to know if there are better methods to facilitate transfer than learning a simple MLP.

Acknowledgments. The work in this paper was funded by NSF grant #1918651; by NIH award # UL1TR003167, and by ARO award #78372-CS.

References

[1] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019. 2, 8

[2] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*, 2019. 8

[3] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Diversity with cooperation: Ensemble methods for few-shot classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3723–3731, 2019. 8

[4] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Selecting Relevant Features from a Multi-domain Representation for Few-shot Classification. *arXiv e-prints*, page arXiv:2003.09338, Mar. 2020. 2, 8

[5] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006. 1

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017. 1, 2, 4

[7] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 9516–9527, 2018. 1, 4

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[9] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *The 2013 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2013. 2

[10] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 2

[11] Jonas Jongejan, Henry Rowley, Takashi Kawashima, Jongmin Kim, and Nick Fox-Gieg. The quick, draw!-ai experiment. *Mount View, CA, accessed Feb*, 17(2018):4, 2016. 2

[12] Taesup Kim, Jaesik Yoon, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. *arXiv preprint arXiv:1806.03836*, 2018. 1, 4

[13] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. *arXiv preprint arXiv:1912.11370*, 6, 2019. 2, 5, 8

[14] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 2

[15] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019. 1

[16] Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. *arXiv preprint arXiv:1801.05558*, 2018. 1, 4

[17] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2019. 1

[18] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *CoRR*, abs/1306.5151, 2013. 2

[19] Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2:2, 2018. 1, 4

[20] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008. 2

[21] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in Neural Information Processing Systems*, 31:721–731, 2018. 2, 7

[22] 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. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1

[23] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018. 7

[24] Brigit Schroeder and Yin Cui. Fgvcx fungi classification challenge 2018. https://github.com/visipedia/fgvcx_fungi_comp, 2018. 2

[25] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017. 2

[26] 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

[27] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 403–412, 2019. 2, 8

[28] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 2

[29] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? *arXiv preprint arXiv:2003.11539*, 2020. 8

[30] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 2

[31] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7278–7286, 2018. 7

[32] Peter Welinder, Steve Branson, Takeshi Mita, Wah Catherine, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 2

[33] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8808–8817, 2020. 2