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

Anonymous ICCV submission

Paper ID 9655

1. Description of Data

1.1. Datasets

ILSVRC2012 [13] Figure 1a. A dataset of natural images of 1000 diverse categories, the most commonly used Imagenet dataset, primarily released for 'Large Scale Visual Recognition Challenge'. We use the ILSVRC-2012 version as the original dataset for the classification challenge has not been modified since. The dataset has a little more than 1.2 million (1,281,167 to be precise) images with each class consisting of images ranging from 732 to 1300.

CUB-200-2011 Birds [20] Figure 1b. A dataset for fine-grained classification of 200 different bird species, an extended version of the CUB-200 dataset. The total number of images in the dataset is 11,788 with mostly 60 images per class.

FGVC-Aircraft [9] Figure 1c. A dataset of images of aircrafts spanning 102 model variants with 10,200 total images and 100 images per class.

FC100 [11] Figure 1d. A dataset curated for few-shot learning based on the popular CIFAR100 [7] includes 100 classes and 600 32×32 color images per class. It offers a more challenging scenario with lower image resolution.

Omniglot [8] Figure 1e. A dataset of images of 1623 handwritten characters from 50 different alphabets. We consider each character as a separate class. The total number of images is 32,460 with 20 examples per class.

Texture [2] Figure 1f. A dataset consists of 5640 images, organized according to 47 categories inspired from human perception. There are 120 images for each category. Image sizes range between 300x300 and 640x640, and the images contain at least 90% of the surface representing the category attribute.

Traffic Sign [5] Figure 1g. A dataset of German Traffic Sign Recognition benchmark consisting of more than 50,000 images across 43 classes.

FGCVx Fungi [14] Figure 1h. A dataset of wild mushrooms species which have been spotted and photographed by the general public in Denmark, containing over 100,000 images across 1,394 classes.

Quick Draw [6] Figure 1i. A dataset of 50 million drawings across 345 categories. We take the simplified drawings, which are 28x28 gray-scale bitmaps and aligned to the center of the drawing's bounding box. Considering the size of the full dataset, we randomly sample 1,000 images from each category.

VGG Flower [10] Figure 1j. A dataset consisting of 8189 images among 102 flower categories that commonly occuring in the United Kingdom. There are between 40 to 258 images in each category.

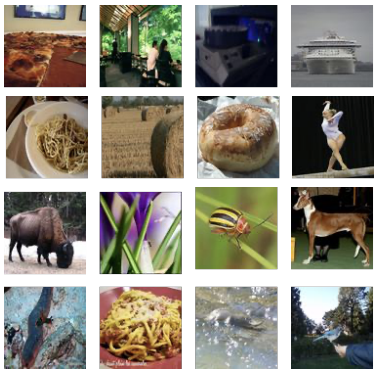
1.2. Data Preparation

For all the datasets, we resize each image into 256 x 256 then crop 224×224 from the center (except quick draw which is already aligned to the center of the drawing's bounding box, so we directly resize quick draw images to 224×224).

1.3. Test Protocol

In this work, for all the methods the training process is performed solely on ILSVRC dataset. For our library based methods, this is followed by hyperparameter validation on the CUB birds dataset. After that, each method is tested on the remaining eight datasets without further tuning.

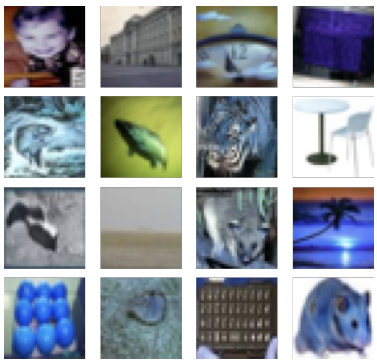
To be more specific, for the library based methods we only use the pre-trained (on ILSVRC dataset) models. While for the meta-learning based methods, we randomly split ILSVRC into a base (training) set of 900 classes for



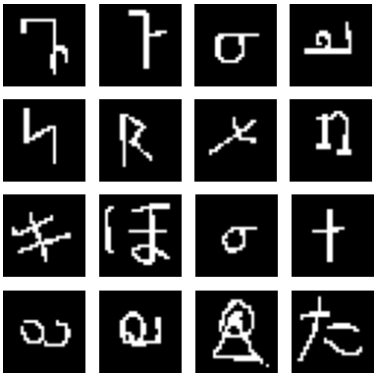
(a) ILSVRC-2012

(b) CUB-200-2011 Birds

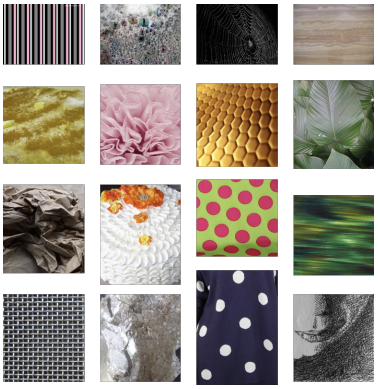
(c) Aircraft



(d) FC100



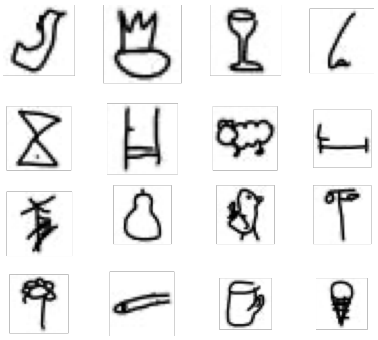
(e) Omniglot



(f) Texture

(g) Traffic Sign

(h) Fungi



(i) Quick Draw

(j) VGG Flower

Figure 1: Images sampled from the data sets used in our experiments.

meta-training and a validation set of the remaining 100 classes.

In order to evaluate a few-shot learner on a data set, for an "n-way m-shot" classification problem, we randomly select n different classes, and then randomly select m images from each class for training (equivalent to 'support' images in meta-learning literature). We then randomly select k images for rest of the images from each class for testing (equivalent to 'query' images in meta-learning literature). We perform this evaluation for n in $\{5, 20, 40\}$ and m in $\{1, 5\}$.

For the library based methods, the query size k is set to 15 (except FGCVx Fungi dataset). For the meta-learning based methods, due to GPU memory constraints, for each class in a task we used 15 query images for 5-way, 10 query images for 10-way, and 5 query images for 40-way problems, with only exception being the Fungi dataset. Fungi dataset has several classes with a very small number of images (6 being the minimum). Hence for Fungi dataset, we use 5 query images per class for 1-shot and 1 query image per class for 5-shot problems. Finally, for every problem, we report the mean of 600 randomly generated test tasks along with the 95% confidence intervals.

For the FC100 data set, there is a small portion overlaps with ILSVRC2012 data set but we still think that the FC100 as a testing set is interesting, as (1) all of the few-shot learners benefitted from the overlapping classes, and (2) this shows how the methods work in the case that the test classes are close to something that the few-shot learner has seen before.

2. Hyper-parameters for Library-Based Learners

In order to achieve an "out-of-the-box" few-shot learner that can be used on any (very small) training set D_{few} without additional data or knowledge of the underlying distribution, an extensive hyper-parameter search is performed on the validation dataset (CUB birds). The hyper-parameters that are found to be the best are then applied in the few-shot training and testing phase for each of our library based methods. The hyper-parameters we are considering include learning rate, number of training epochs, L2-regularization penalty constant, the number of neurons in the MLP hidden layer, and whether to drop the hidden layer altogether. A separate hyper-parameter search is used for 5-way, 20-way, and 40-way 1-shot classification. 5-shot problems are using the same hyper-parameters as 1-shot ones. Experiments suggest that dropping the hidden layer did not significantly help performance, but did occasionally hurt performance significantly on the validation set; as a result, we always use a hidden layer. We train all our methods using Adam optimizer. The hyper-parameters details can be found in Table 8.

3. Competitive Methods

For methods that require a pre-trained CNN (FEAT, Meta-transfer, and SUR), we use the pre-trained ResNet18 pytorch library as the backbone. We follow the hyper-parameter setting from [16], [21], and [3]. For the FEAT and Meta-transfer methods, we perform meta-training on ILSVRC dataset[13] before testing on eight different datasets. For the SUR method, we follow [3] and build a multi-domain representation by pre-training multiple ResNet18 on Meta-Dataset [18] (one per data set). To evaluate SUR on data set X, we use feature extractors trained on the rest of the data sets in {Omniglot, Aircraft, Birds, Texture, Quickdraw, Flowers, Fungi,and ILSVRC}. Traffic Sign and FC100 data sets are reserved for testing only. To be more specific, the meta-training setting are as follows:

Meta-transfer Learning
The base-learner is optimized by batch gradient descent with the learning rate of 10^{-2} and gets updated every 50 steps. The meta learner is optimized by Adam optimizer with an initial learning rate of 10^{-3}, and decaying 50% every 1,000 iterations until 10^{-4}.

FEAT
The vanilla stochastic gradient descent with Nesterov acceleration is applied. The initial rate is set to 2×10^{-4} and decreases every 40 steps with a decay rate of 5×10^{-4} and momentum of 0.9. The learning rate for the scheduler is set to 0.5.

SUR
We follow [3] and apply SGD with momentum during optimization. The learning rate is adjusted with cosine annealing. The initial learning rate, the maximum number of training iterations ("Max Iter.") and annealing frequency ("Annealing Freq.") are adjusted individually according to each data set (Table 1). Data augmentation is also deployed to regularize the training process, which includes random crops and random color augmentations with a constant weight decay of 7×10^{-4}.

Baselines
We train the two pre-training based methods, Baseline and Baseline++ [1] following the hyper-parameters suggested by the original authors. However, since we train them on ILSVRC data as opposed to mini-imagenet [12]. During the training stage, we train 50 epochs with a batch size of 16. In the paper, the authors have trained 400 epochs on the base set of mini-imagenet consisting of 64 classes.

	Learning Rate	Weight Decay	Max Iter.	Annealing Freq.	Batch Size
ILSVRC	3×10^{-2}	7×10^{-4}	480,000	48,000	64
Omniglot	3×10^{-2}	7×10^{-4}	50,000	3,000	16
Aircraft	3×10^{-2}	7×10^{-4}	50,000	3,000	8
Birds	3×10^{-2}	7×10^{-4}	50,000	3,000	16
Textures	3×10^{-2}	7×10^{-4}	50,000	1,500	32
Quick Draw	1×10^{-2}	7×10^{-4}	480,000	48,000	64
Fungi	3×10^{-2}	7×10^{-4}	480,000	15,000	32
VGG Flower	3×10^{-2}	7×10^{-4}	50,000	1,500	8

Table 1: Hyper-parameter settings for SUR individual feature networks on MetaDataset.

Mini-imagenet has 600 images per class, whereas ILSVRC has an average of around 1,200 images per class. So, the total number of batches trained in our baselines is $50 \times (1000 \times 1,200)/16 = 3,750,000$, as opposed to $400 \times (64 \times 600)/16 = 960,000$ in the original paper.

Metric-Learning Methods and MAML
For the three most popular metric-learning methods, MatchingNet [19], ProtoNet [15] and RelationNet [17], again we followed the implementation and hyper-parameters provided by [1].

All the metric-learning methods and MAML [4] are trained using the Adam optimizer with initial learning rate of 10^{-3}. For MAML, the inner learning rate is kept at 10^{-2} as suggested by the original authors. And following [1], we do the following modifications: For MatchingNet, we use an FCE classification layer without fine-tuning and also multiply the cosine similarity by a constant scalar. For RelationNet, we replace the L2 norm with a softmax layer to expedite training. For MAML, we use a first-order approximation in the gradient for memory efficiency. Even then, we could not train MAML for 40-way in a single GPU due to memory shortage. Hence, we drop MAML for 40-way experiments.

During the meta-training stage of all these methods, we train 150,000 episodes for 5-way 1-shot, 5-way 5-shot, and 20-way 1-shot problems, and 50,000 episodes for all other problems on the base split of ILSVRC. Here an episode refers to one meta-learning task that includes training on the 'support' images and testing on the 'query' images. The stopping episode number is chosen based on no significant increase in validation accuracy. In the original paper, the authors trained 60,000 episodes for 1-shot and 40,000 episodes for 5-shot tasks. We meticulously observe that those numbers are too low for certain problems in terms of validation accuracy. Hence, we allow more episodes. We select the best accuracy model on the validation set of ILSVRC for meta-testing on other datasets.

4. Complete Results

Here we conduct additional experiments which are not reported in our main paper.

Single Library Learners VS. Competitive Methods
Table 2, 3, 4, 5 show the performance comparison of single library learners and the competitive methods including Baseline, Baseline++, MAML, MatchingNet, ProtoNet, RelationNet, Meta-transfer Learning, and FEAT. The comparison is conducted on problems of 1-shot under 5-way, 20-way, and 40-way and 5-shot under 40-way.

Full Library VS. Google BiT Methods
Table 6 shows the performance comparison of the full library method and the Google BiT methods. The problems addressed here are 1-shot under 5-way, 20-way, and 40-way. The full library method utilizes a library of nine, ILSVRC-trained CNNs, while the Google BiT methods individually use three deep CNNs trained on the full ImageNet.

Full Library VS. Hard and Soft Ensemble Methods
Table 7 compares the performances of the full library method and the hard and soft ensemble methods (bagging) for 1-shot problems under 5-way, 20-way, and 40-way.

	Aircraft	FC100	Omniglot	Texture	Traffic	Fungi	Quick Draw	VGG Flower
Baseline	36.6 ± 0.7	40.3 ± 0.8	65.2 ± 1.0	42.9 ± 0.7	57.6 ± 0.9	35.9 ± 0.9	49.0 ± 0.9	67.9 ± 0.9
Baseline++	33.9 ± 0.7	39.3 ± 0.7	59.8 ± 0.7	40.8 ± 0.7	58.8 ± 0.8	35.4 ± 0.9	46.6 ± 0.8	58.6 ± 0.9
MAML	26.5 ± 0.6	39.4 ± 0.8	50.7 ± 1.0	38.1 ± 0.9	45.6 ± 0.8	34.8 ± 1.0	46.0 ± 1.0	54.7 ± 0.9
MatchingNet	29.9 ± 0.6	38.2 ± 0.8	51.7 ± 1.0	39.3 ± 0.7	55.2 ± 0.8	38.1 ± 0.9	50.2 ± 0.8	51.8 ± 0.9
ProtoNet	31.8 ± 0.6	40.9 ± 0.8	79.2 ± 0.8	42.0 ± 0.8	54.4 ± 0.9	36.7 ± 0.9	55.8 ± 1.0	59.1 ± 0.9
RelationNet	31.2 ± 0.6	46.3 ± 0.9	69.6 ± 0.9	41.0 ± 0.8	54.6 ± 0.8	36.8 ± 1.0	52.5 ± 0.9	55.5 ± 0.9
Meta-transfer	30.4 ± 0.6	57.6 ± 0.9	78.9 ± 0.8	50.1 ± 0.8	62.3 ± 0.8	45.8 ± 1.0	58.4 ± 0.9	73.2 ± 0.9
FEAT	33.2 ± 0.7	42.1 ± 0.8	69.8 ± 0.9	51.8 ± 0.9	49.0 ± 0.8	46.9 ± 1.0	53.1 ± 0.8	75.3 ± 0.9
SUR	33.5 ± 0.6	42.1 ± 1.0	93.4 ± 0.5	42.8 ± 0.8	45.3 ± 0.9	44.1 ± 1.0	54.3 ± 0.9	72.6 ± 1.0
Worst library-based	40.9 ± 0.9 RN18	50.8 ± 0.9 DN121	77.2 ± 0.8 RN152	59.0 ± 0.9 DN169	55.5 ± 0.8 RN152	53.0 ± 0.9 DN201	57.3 ± 0.9 RN101	79.7 ± 0.8 RN18
Best library-based	**46.1 ± 1.0** DN161	**61.2 ± 0.9** RN152	**86.5 ± 0.6** DN121	**65.1 ± 0.9** RN101	**66.6 ± 0.9** DN201	**56.6 ± 0.9** DN121	**62.8 ± 0.9** RN18	**83.2 ± 0.8** DN201

Table 2: Comparing competitive methods with the simple library-based learners, on the 5-way, 1-shot problem.

	Aircraft	FC100	Omniglot	Texture	Traffic	Fungi	Quick Draw	VGG Flower
Baseline	11.4 ± 0.2	15.4 ± 0.3	38.9 ± 0.6	17.6 ± 0.3	27.8 ± 0.4	13.1 ± 0.3	21.9 ± 0.4	38.8 ± 0.4
Baseline++	10.1 ± 0.2	15.8 ± 0.3	39.2 ± 0.4	18.1 ± 0.3	31.5 ± 0.3	13.7 ± 0.3	22.5 ± 0.3	33.1 ± 0.4
MAML	7.6 ± 0.1	17.2 ± 0.3	29.1 ± 0.4	14.8 ± 0.2	19.4 ± 0.3	11.5 ± 0.3	19.7 ± 0.3	21.8 ± 0.3
MatchingNet	7.7 ± 0.1	17.0 ± 0.3	44.6 ± 0.5	19.8 ± 0.3	26.2 ± 0.3	16.1 ± 0.4	26.7 ± 0.4	29.9 ± 0.4
ProtoNet	11.5 ± 0.2	20.1 ± 0.3	58.8 ± 0.5	20.0 ± 0.3	29.5 ± 0.4	16.2 ± 0.4	30.7 ± 0.4	40.7 ± 0.4
RelationNet	11.0 ± 0.2	18.7 ± 0.3	51.3 ± 0.5	18.6 ± 0.3	28.5 ± 0.3	15.9 ± 0.4	29.1 ± 0.4	35.5 ± 0.4
Meta-transfer	12.8 ± 0.2	30.9 ± 0.4	58.6 ± 0.5	27.2 ± 0.4	35.8 ± 0.4	23.1 ± 0.4	35.2 ± 0.4	52.0 ± 0.5
FEAT	15.1 ± 0.3	18.9 ± 0.4	51.1 ± 0.6	28.9 ± 0.5	31.7 ± 0.4	25.7 ± 0.4	28.7 ± 0.5	56.5 ± 0.6
SUR	14.2 ± 0.3	19.1 ± 0.4	84.2 ± 0.4	21.0 ± 0.3	26.2 ± 0.3	21.7 ± 0.4	34.2 ± 0.4	57.1 ± 0.5
Worst library-based	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	33.3 ± 0.5 RN101	62.4 ± 0.5 RN101
Best library-based	**24.3 ± 0.3** DN161	**36.4 ± 0.4** RN152	**69.1 ± 0.4** DN121	**42.5 ± 0.4** DN152	**38.5 ± 0.4** DN201	**33.9 ± 0.5** DN161	**39.5 ± 0.5** DN201	**70.0 ± 0.5** DN161

Table 3: Comparing competitive methods with the simple library-based learners, on the 20-way, 1-shot problem.

	Aircraft	FC100	Omniglot	Texture	Traffic	Fungi	Quick Draw	VGG Flower
Baseline	6.9 ± 0.2	9.7 ± 0.1	27.1 ± 0.4	11.2 ± 0.1	20.0 ± 0.3	8.2 ± 0.2	14.8 ± 0.3	28.7 ± 0.3
Baseline++	6.1 ± 0.1	10.1 ± 0.1	29.9 ± 0.3	12.1 ± 0.2	23.2 ± 0.2	8.7 ± 0.2	15.9 ± 0.2	24.2 ± 0.3
MatchingNet	5.1 ± 0.1	11.6 ± 0.2	32.5 ± 0.4	15.0 ± 0.2	19.3 ± 0.2	11.0 ± 0.2	19.0 ± 0.3	26.9 ± 0.3
ProtoNet	7.2 ± 0.2	13.8 ± 0.2	47.2 ± 0.4	14.4 ± 0.2	21.6 ± 0.3	11.3 ± 0.2	22.9 ± 0.3	31.0 ± 0.3
RelationNet	6.2 ± 0.2	13.8 ± 0.2	45.2 ± 0.4	11.4 ± 0.2	18.2 ± 0.2	10.7 ± 0.2	21.1 ± 0.3	28.1 ± 0.3
Meta-transfer	7.6 ± 0.2	20.6 ± 0.2	37.5 ± 0.3	19.2 ± 0.2	24.4 ± 0.2	10.2 ± 0.2	22.2 ± 0.2	40.4 ± 0.3
FEAT	10.1 ± 0.2	12.9 ± 0.3	35.7 ± 0.4	21.8 ± 0.3	24.9 ± 0.3	18.0 ± 0.3	19.4 ± 0.3	47.6 ± 0.4
SUR	9.9 ± 0.1	13.3 ± 0.2	78.1 ± 0.3	15.0 ± 0.2	19.7 ± 0.2	15.6 ± 0.3	26.7 ± 0.3	48.8 ± 0.3
Worst library-based	14.2 ± 0.2 RN34	19.6 ± 0.2 RN18	47.3 ± 0.3 RN152	28.8 ± 0.2 RN18	22.2 ± 0.2 RN152	23.7 ± 0.3 RN34	26.4 ± 0.3 RN152	53.1 ± 0.3 RN34
Best library-based	**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 4: Comparing competitive methods with the simple library-based learners, on the 40-way, 1-shot problem.

	Aircraft	FC100	Omniglot	Texture	Traffic	Fungi	Quick Draw	VGG Flower
Baseline	17.0 ± 0.2	29.4 ± 0.3	80.0 ± 0.3	27.1 ± 0.2	49.1 ± 0.3	24.0 ± 0.5	41.5 ± 0.3	68.3 ± 0.3
Baseline++	12.3 ± 0.2	24.4 ± 0.3	67.0 ± 0.3	25.1 ± 0.3	44.1 ± 0.3	19.7 ± 0.5	35.2 ± 0.3	53.9 ± 0.3
MatchingNet	8.6 ± 0.2	22.1 ± 0.3	59.1 ± 0.4	23.3 ± 0.3	37.1 ± 0.3	19.0 ± 0.5	31.5 ± 0.3	46.7 ± 0.4
ProtoNet	13.8 ± 0.2	28.0 ± 0.3	80.0 ± 0.3	29.6 ± 0.3	39.7 ± 0.3	23.6 ± 0.5	42.6 ± 0.4	61.6 ± 0.3
RelationNet	10.7 ± 0.2	26.1 ± 0.3	77.0 ± 0.3	18.7 ± 0.2	29.6 ± 0.3	18.3 ± 0.5	40.6 ± 0.3	47.0 ± 0.3
Meta-transfer	9.7 ± 0.2	29.9 ± 0.2	48.1 ± 0.3	29.8 ± 0.2	33.3 ± 0.2	12.2 ± 0.3	31.6 ± 0.2	55.4 ± 0.3
FEAT	16.2 ± 0.3	27.1 ± 0.4	58.5 ± 0.4	36.8 ± 0.3	37.3 ± 0.3	32.9 ± 0.6	35.6 ± 0.4	74.0 ± 0.4
SUR	15.5 ± 0.2	32.6 ± 0.3	94.2 ± 0.1	25.8 ± 0.1	37.0 ± 0.2	26.3 ± 0.6	45.0 ± 0.3	69.8 ± 0.3
Worst library-based	28.4 ± 0.2 RN34	37.3 ± 0.2 RN18	79.9 ± 0.3 RN152	48.4 ± 0.2 RN18	47.2 ± 0.2 RN152	46.6 ± 0.3 RN34	49.8 ± 0.3 RN152	81.4 ± 0.2 RN34
Best library-based	**35.9 ± 0.2** DN161	**48.2 ± 0.3** RN152	**89.4 ± 0.2** DN201	**55.4 ± 0.2** DN161	**57.5 ± 0.2** DN201	**52.1 ± 0.3** DN161	**55.5 ± 0.3** DN201	**88.9 ± 0.2** DN161

Table 5: Comparing competitive methods with the simple library-based learners, on the 40-way, 5-shot problem.

	Aircraft	FC100	Omniglot	Texture	Traffic	Fungi	QDraw	Flower
5-way, 1-shot								
Full Library	44.9 ± 0.9	60.9 ± 0.9	**88.4 ± 0.7**	68.4 ± 0.8	**66.2 ± 0.9**	57.7 ± 1.0	**65.4 ± 0.9**	86.3 ± 0.8
BiT-ResNet-101-3	42.2 ± 1.0	58.5 ± 0.9	76.2 ± 1.1	63.6 ± 0.9	47.4 ± 0.8	**63.7 ± 0.9**	54.9 ± 0.9	98.0 ± 0.3
BiT-ResNet-152-4	40.5 ± 0.9	61.0 ± 0.9	78.4 ± 0.9	65.6 ± 0.8	48.4 ± 0.8	62.2 ± 1.0	55.4 ± 0.9	97.9 ± 0.3
BiT-ResNet-50-1	**45.0 ± 1.0**	**61.9 ± 0.9**	79.4 ± 0.9	**68.5 ± 0.9**	56.1 ± 0.8	61.6 ± 1.0	54.0 ± 0.9	**98.5 ± 0.2**
20-way, 1-shot								
Full Library	**25.2 ± 0.3**	36.8 ± 0.4	**76.4 ± 0.4**	**46.3 ± 0.4**	**40.0 ± 0.4**	37.6 ± 0.5	**44.2 ± 0.5**	75.5 ± 0.5
BiT-ResNet-101-3	19.9 ± 0.3	34.5 ± 0.4	53.4 ± 0.6	44.2 ± 0.4	24.6 ± 0.3	**41.6 ± 0.5**	32.0 ± 0.4	**95.7 ± 0.2**
BiT-ResNet-152-4	18.2 ± 0.3	**37.2 ± 0.4**	51.0 ± 0.6	44.3 ± 0.4	23.6 ± 0.3	40.3 ± 0.5	29.0 ± 0.4	95.0 ± 0.2
BiT-ResNet-50-1	22.6 ± 0.4	36.1 ± 0.4	58.1 ± 0.5	45.7 ± 0.4	28.3 ± 0.3	39.2 ± 0.4	30.0 ± 0.4	95.4 ± 0.2
40-way, 1-shot								
Full Library	**18.6 ± 0.2**	27.3 ± 0.2	**67.7 ± 0.3**	37.0 ± 0.3	**30.3 ± 0.2**	29.3 ± 0.3	**34.5 ± 0.3**	67.6 ± 0.3
BiT-ResNet-101-3	12.7 ± 0.2	24.5 ± 0.2	19.7 ± 0.4	34.4 ± 0.3	15.9 ± 0.2	**30.7 ± 0.3**	13.5 ± 0.2	91.9 ± 0.2
BiT-ResNet-152-4	12.4 ± 0.2	26.9 ± 0.2	30.8 ± 0.5	35.8 ± 0.3	16.0 ± 0.2	**30.7 ± 0.3**	18.7 ± 0.3	91.4 ± 0.2
BiT-ResNet-50-1	15.8 ± 0.2	**27.5 ± 0.2**	47.6 ± 0.4	**37.4 ± 0.3**	19.7 ± 0.2	30.6 ± 0.3	21.7 ± 0.2	**93.4 ± 0.2**

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	FC100	Omniglot	Texture	Traffic	Fungi	Quick Draw	VGG Flower
				5-way, 1-shot				
Full Library	44.9 ± 0.9	60.9 ± 0.9	**88.4 ± 0.7**	**68.4 ± 0.8**	66.2 ± 0.9	57.7 ± 1.0	**65.4 ± 0.9**	**86.3 ± 0.8**
Hard Ensemble	45.0 ± 0.9	60.0 ± 0.9	**88.4 ± 0.6**	67.9 ± 0.9	65.2 ± 0.8	**58.1 ± 0.9**	64.7 ± 1.0	84.9 ± 0.8
Soft Ensemble	44.2 ± 0.9	61.0 ± 0.9	88.2 ± 0.6	67.4 ± 0.9	63.2 ± 0.8	57.1 ± 1.0	65.2 ± 0.9	**86.3 ± 0.7**
Best Single	**46.1 ± 1.0**	**61.2 ± 0.9**	86.5 ± 0.6	65.1 ± 0.9	**66.6 ± 0.9**	56.6 ± 0.9	62.8 ± 0.9	83.2 ± 0.8
				20-way, 1-shot				
Full Library	**25.2 ± 0.3**	**36.8 ± 0.4**	**76.4 ± 0.4**	**46.3 ± 0.4**	**40.0 ± 0.4**	**37.6 ± 0.5**	**44.2 ± 0.5**	**75.5 ± 0.5**
Hard Ensemble	23.9 ± 0.3	36.3 ± 0.4	73.4 ± 0.4	45.7 ± 0.4	38.2 ± 0.4	36.4 ± 0.4	42.7 ± 0.4	73.3 ± 0.5
Soft Ensemble	24.2 ± 0.3	36.4 ± 0.4	73.8 ± 0.4	**46.3 ± 0.5**	37.7 ± 0.4	37.1 ± 0.4	43.4 ± 0.5	74.3 ± 0.5
Best Single	24.3 ± 0.3	36.4 ± 0.4	69.1 ± 0.4	42.5 ± 0.4	38.5 ± 0.4	33.9 ± 0.5	39.5 ± 0.5	70.0 ± 0.5
				40-way, 1-shot				
Full Library	**18.6 ± 0.2**	27.3 ± 0.2	**67.7 ± 0.3**	**37.0 ± 0.3**	**30.3 ± 0.2**	**29.3 ± 0.3**	**34.5 ± 0.3**	**67.6 ± 0.3**
Hard Ensemble	17.7 ± 0.2	27.3 ± 0.2	65.8 ± 0.3	36.3 ± 0.3	29.0 ± 0.2	28.7 ± 0.3	33.7 ± 0.3	66.1 ± 0.3
Soft Ensemble	18.1 ± 0.2	**27.6 ± 0.3**	66.2 ± 0.3	**37.0 ± 0.3**	29.0 ± 0.2	29.1 ± 0.2	34.1 ± 0.3	67.4 ± 0.3
Best Single	17.4 ± 0.2	27.2 ± 0.3	61.6 ± 0.3	33.2 ± 0.3	29.5 ± 0.2	26.8 ± 0.3	31.2 ± 0.3	62.8 ± 0.3

Table 7: Accuracy obtained using all nine library CNNs as the basis for a few-shot learner.

	Number of Epochs	Hidden Size	Learning Rate	Regularization Constant
		5-way, 1-shot and 5-shot		
DenseNet121	200	1024	1×10^{-3}	0.2
DenseNet161	100	1024	5×10^{-4}	0.2
DenseNet169	300	1024	5×10^{-4}	0.5
DenseNet201	100	512	5×10^{-4}	0.5
ResNet18	200	512	1×10^{-3}	0.2
ResNet34	100	1024	5×10^{-4}	0.2
ResNet50	300	2048	5×10^{-4}	0.1
ResNet101	100	512	1×10^{-3}	0.1
ResNet152	300	512	5×10^{-4}	0.1
Full Library	300	1024	5×10^{-4}	0.1
BiT-ResNet-101-3	300	4096	1×10^{-3}	0.7
BiT-ResNet-152-4	300	2048	5×10^{-4}	0.7
BiT-ResNet-50-1	200	2048	5×10^{-4}	0.5
		20-way, 1-shot and 5-shot		
DenseNet121	100	1024	5×10^{-4}	0.2
DenseNet161	100	512	1×10^{-3}	0.1
DenseNet169	300	512	5×10^{-4}	0.1
DenseNet201	200	1024	5×10^{-4}	0.1
ResNet18	200	2048	5×10^{-4}	0.1
ResNet34	100	1024	5×10^{-4}	0.1
ResNet50	100	1024	5×10^{-4}	0.1
ResNet101	200	2048	5×10^{-4}	0.2
ResNet152	100	512	5×10^{-4}	0.2
Full Library	100	512	5×10^{-4}	0.1
BiT-ResNet-101-3	300	2048	5×10^{-4}	0.5
BiT-ResNet-152-4	300	1024	5×10^{-4}	0.5
BiT-ResNet-50-1	100	2048	5×10^{-4}	0.9
		40-way, 1-shot and 5-shot		
DenseNet121	100	2048	5×10^{-4}	0.1
DenseNet161	100	512	5×10^{-4}	0.1
DenseNet169	100	512	1×10^{-3}	0.2
DenseNet201	100	1024	5×10^{-4}	0.1
ResNet18	100	512	1×10^{-3}	0.1
ResNet34	100	2048	5×10^{-4}	0.2
ResNet50	100	512	5×10^{-4}	0.1
ResNet101	100	512	5×10^{-4}	0.1
ResNet152	100	1024	5×10^{-4}	0.1
Full Library	100	1024	5×10^{-4}	0.1
BiT-ResNet-101-3	300	512	5×10^{-4}	0.7
BiT-ResNet-152-4	200	1024	5×10^{-4}	0.5
BiT-ResNet-50-1	300	1024	5×10^{-4}	0.5

Table 8: Hyper-parameter settings for different backbones in 5, 20, and 40 ways.

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. 3, 4

[2] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1

[3] 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. 3

[4] 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. 4

[5] 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. 1

[6] 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. 1

[7] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1

[8] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 1

[9] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 1

[10] 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. 1

[11] 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. 1

[12] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016. 3

[13] 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, 3

[14] B Schroeder and Y Cui. Fgvcx fungi classification challenge 2018, 2018. 1

[15] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017. 4

[16] 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. 3

[17] 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. 4

[18] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples. *arXiv e-prints*, page arXiv:1903.03096, Mar. 2019. 3

[19] 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. 4

[20] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 1

[21] 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. 3