Efficient Conditional Pre-training for Transfer Learning

Shuvam Chakraborty
Stanford University

Burak Uzkent[*]
Stanford University

Kumar Ayush[*]
Stanford University

Kumar Tanmay
IIT Kharagpur

Evan Sheehan
Stanford University

Stefano Ermon
Stanford University

Abstract

Almost all the state-of-the-art neural networks for computer vision tasks are trained by (1) pre-training on a large-scale dataset and (2) finetuning on the target dataset. This strategy helps reduce dependence on the target dataset and improves convergence rate and generalization on the target task. Although pre-training on large-scale datasets is very useful for new methods or models, its foremost disadvantage is high training cost. To address this, we propose efficient filtering methods to select relevant subsets from the pre-training dataset. Additionally, we discover that lowering image resolutions in the pre-training step offers a great trade-off between cost and performance. We validate our techniques by pre-training on ImageNet in both the unsupervised and supervised settings and finetuning on a diverse collection of target datasets and tasks. Our proposed methods drastically reduce pre-training cost and provide strong performance boosts. Finally, we improve the current standard of ImageNet pre-training by 1-3% *by tuning available models on our subsets and pre-training on a dataset filtered from a larger scale dataset.*

A. Method Visualization

We present visual depictions for clustering based filtering in Figure 1 and for the domain classifier in Figure 2.

B. Additional Methods

B.1. Active Learning

Active learning is a research field concentrating on understanding which samples in a pool of samples should be given priority for annotation. One of the most common and simple active learning method relies on training a model on a labeled dataset and finding the entropy of the unseen samples by running them through the trained model. Next, top N unseen samples w.r.t their entropy (assigned by the current model) are listed in descending order. Usually, there is a single data distribution for labelled and unlabelled data, however, for our task we consider two data distributions: pre-training and target, which can be similar or completely different. For this reason, we apply two variations of active learning to conditional pre-training. First, we train a network f_t on the target dataset $\mathcal{D}_t$ and run images x_s^i in source dataset through the network f_t to get the entropy of the predictions. Next, we list the images x_s^i by ascending or descending entropy and choose the top $\mathcal{N}'$ images. Choosing high entropy samples can be interpreted as standard active learning, and we call the method that chooses low entropy images *Inverse Active Learning*.

Dataset	#classes	#train	#test
Stanford Cars [11]	196	8143	8041
Caltech Birds [8]	200	6000	2788
Functional Map of the World [2]	62	18180	10609

Table 1. We use three challenging visual categorization datasets to evaluate the proposed pre-training strategies on target classification tasks.

B.2. Experimental Setup

For classification tasks, we train the linear classification layer from scratch and finetune the pre-trained backbone weights. We give basic details about our classification datasets in Table 1.

Methods We experiment with clustering based filtering, using $K = 200$ clusters and both average and min distance to cluster centers, as well as our domain classifier method, using ResNet-18 [6] as our classifier. Furthermore, we combine our filtering methods with downsizing pre-training im-

[*] Equal Contribution. Contact: {shuvamc, buzkent, kayush}@cs.stanford.edu

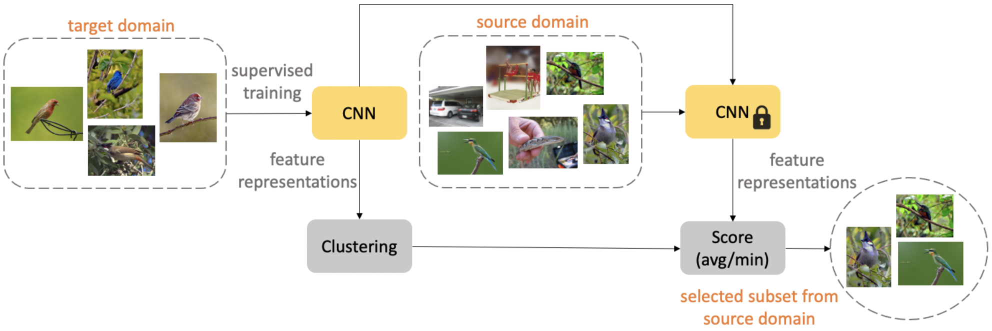

Figure 1. Schematic overview of clustering based filtering. We first train a model on the target domain to extract representations, which we use to cluster the target domain. We score source images with either average or min distance to cluster centers and then filter.

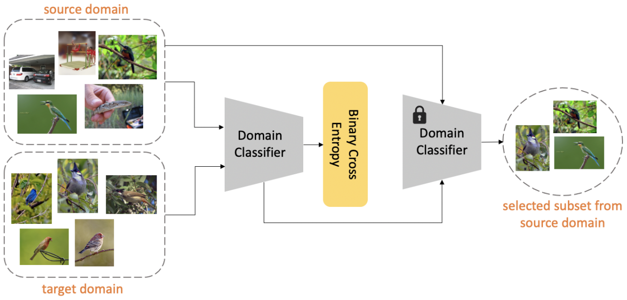

Figure 2. Depiction of the Domain Classifier. We train a simple binary classifier to discriminate between source and target domain and then use the output probabilities on source images to filter.

age resolution from 224x224 to 112x112 using bilinear interpolation. We always perform filtering on 224 resolution source images, but use it to pre-training at both resolutions to assess flexibility, as we want robust methods that do not need to be specifically adjusted to the pre-training setup.

Supervised Pre-training. For supervised pre-training, in all experiments, we utilize the ResNet-34 model [6] on 1 Nvidia-TITAN X GPU. We perform standard cropping/flipping transforms for ImageNet and the target data. For pre-training, we pretrain on the given subset of ImageNet for 90 epochs, utilizing SGD with momentum .9, weight decay of 1e-4, and learning rate .01 with a decay of 0.1 every 30 epochs. We finetune for 90 epochs with a learning rate decay of 0.1 every 30 epochs for all datasets. For Cars and Birds, we utilize SGD with momentum .9 [10], learning rate 0.1, and weight decay of 1e-4. For fMoW, we utilize the Adam optimizer [7] with learning rate 1e-4.

Unsupervised Pre-training. For unsupervised pre-training, we utilize the state of the art MoCo-v2 [4] technique using a ResNet-50 model [6] in all experiments. We train on 4 Nvidia GPUs. MoCo [3, 4] is a self-supervised learning method that utilizes contrastive learning, where the goal is to maximize agreement between different views of the same image (positive pairs) and to minimize agreement between different images (negative pairs). Our choice to use MoCo is driven by (1) performance, and (2) computational cost. Compared to other self-supervised frameworks, such as SimCLR [1], which require a batch size of 4096, MoCo uses a momentum updated queue of previously seen samples and achieves comparable performance with a batch size of just 256 [3].

We keep the same data augmentations and hyperparameters used in [4]. We finetune the MoCo pre-trained backbone on our target tasks for 100 epochs using a learning rate of 0.001, batch size of 64, SGD optimizer for Cars and Birds, and Adam optimizer for fMoW.

B.3. Low Level Tasks

Object Detection. We use a standard setup for object detection with a Faster R-CNN detector with a R50-C4 backbone as in [3, 5, 12]. We pre-train the backbone with MoCo-v2 on the full or filtered subset of ImageNet. We finetune the final layers for 24k iterations (∼ 23 epochs) on trainval2007 (∼ 5k images). We evaluate on the VOC test2007 set with the default metric AP50 and the more stringent metrics of COCO-style [9] AP and AP75. For filtering, we use the domain classifier with no modifications and for clustering we use MoCo-v2 on Pascal VOC to learn representations.

Semantic Segmentation. We use PSAnet [13] network with ResNet-50 backbone to perform semantic segmentation. We train PSAnet network with a batch size of 16 and a learning rate of 0.01 for 100 epochs and use SGD optimizer. Similar to object detection, we pre-train the backbone with MoCo-v2 on the full or filtered subset of ImageNet and then we finetune the network using VOC train2012. We evaluate on the VOC test2012 set with the following three metrics: (a) **mIOU**: standard segmentation metric, (b) **mAcc**: mean classwise pixel accuracy, (c) **allAcc**: total pixel accuracy. For filtering, we use the domain classifier with no modifications and for clustering we use MoCo-v2 on Pascal VOC to learn representations.

C. Additional Results

C.1. Active Learning

We utilize our Active Learning based methods using the same supervised pre-training and finetuning setup described previously. We present our results updated with Active Learning in Table 2.

Least vs Most Confident Samples We see that at 224×224 pixels resolution pre-training, standard active learning seems to be applicable to the transfer learning setting as selecting samples with high entropy generally does better than the inverse. However, at lower resolution (112×112 pixels) pre-training, active learning does worse than inverse active learning and random in most cases, suggesting a lack of robustness for the active learning method since filtering is performed at 224×224 pixels resolution.

Performance Comparison As alluded to, active learning methods perform noticeably worse in the lower resolution setting for all datasets, suggesting that filtering and pre-training conditions must be similar to maintain good performance, unlike domain classifier and clustering. In general, we see that for Cars and Birds, even at 224×224 pixels resolution pre-training, active learning performance lags behind our clustering and domain classifier methods and struggles to improve over the simple random baseline in several settings. In contrast, for an out of distribution dataset like fMoW, active learning does well in the 224×224 pixels resolution pre-training setting. Since active learning directly considers label distribution when filtering, it may be more prone to overfitting compared to the other methods. This can degrade its performance when relevant features are shared between the pre-training and target datasets and thus focusing only on features, not labels, when filtering may be more effective. However, in fMoW, there is very little overlap in relevant features with ImageNet, bridging the gap between active learning, and domain classifier and clustering.

Adaptability Comparison Active learning is noticeably less flexible than other methods as it relies on a notion of confidence that can be hard to construct and quantify for target tasks besides classification. As said, it is also much more sensitive to filtering and pre-training resolution, making it a less robust method. In general, we see that our clustering and domain classification methods can outperform a non-trivial baseline like active learning in flexibility, adaptability, and performance.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2

[2] Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6172–6180, 2018. 1

[3] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 2

[4] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 2

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[6] 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. 1, 2

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[8] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 1

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2

[10] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999. 2

[11] 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

[12] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 2

[13] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 267–283, 2018. 2

Supervised Pre-train.		**Target Dataset**			**Cost**
224 x 224		**Small Shift**		**Large Shift**	**(hrs)**
Pretrain. Sel. Method		**Cars**	**Birds**	**fMow**	
0%	**Random Init.**	52.89	42.17	43.35	0
100%	**Entire Dataset**	82.63	74.87	59.05	160-180
6%	**Random**	72.2	57.87	50.25	30-35
	Inv. Active Learning	72.19	58.17	49.7	40-45
	Active Learning	73.17	57.77	**50.91**	40-45
	Domain Cls.	**74.37**	**59.73**	**51.17**	35-40
	Clustering (Avg)	73.64	56.33	**51.14**	40-45
	Clustering (Min)	**74.23**	57.67	50.27	40-45
12%	**Random**	76.12	62.73	53.28	45-50
	Inv. Active Learning	76.1	62.7	**53.43**	55-60
	Active Learning	76.43	63.7	**53.63**	55-60
	Domain Cls.	76.18	**64**	**53.41**	50-55
	Clustering (Avg)	**77.12**	61.73	53.12	55-60
	Clustering (Min)	75.81	**64.07**	52.91	55-60

Supervised Pre-train.		**Target Dataset**			**Cost**
112 x 112		**Small Shift**		**Large Shift**	**(hrs)**
Pretrain. Sel. Method		**Cars**	**Birds**	**fMow**	
0%	**Random Init**	52.89	42.17	43.35	0
100%	**Entire Dataset**	83.78	73.47	57.39	90-110
6%	**Random**	72.76	57.4	49.73	15-20
	Inv. Active Learning	71.05	58.43	49.56	25-30
	Active Learning	72.95	56.3	48.94	25-30
	Domain Cls.	73.66	**58.73**	50.66	20-25
	Clustering (Avg)	**74.53**	56.97	**51.32**	25-30
	Clustering (Min)	71.72	**58.73**	49.06	25-30
12%	**Random**	75.4	62.63	52.59	30-35
	Inv. Active Learning	75.3	62.4	52.45	40-45
	Active Learning	76.26	61.9	52.04	40-45
	Domain Cls.	76.36	**63.5**	**53.37**	35-40
	Clustering (Avg)	**77.53**	61.23	52.67	40-45
	Clustering (Min)	76.36	63.13	51.6	40-45

Table 2. Results on supervised pre-training and classification tasks, including Active Learning.