When Semi-Supervised Learning Meets Transfer Learning: Training Strategies, Models and Datasets

Hong-Yu Zhou[1] Avital Oliver[2] Jianxin Wu[3] Yefeng Zheng[1]
[1]YouTu Lab, Tencent [2]Google Brain [3]Nanjing University
{whuzhouhongyu, wujx2001, yefeng.zheng}@gmail.com, avitalo@google.com

Abstract

Semi-Supervised Learning (SSL) has been proved to be an effective way to leverage both labeled and unlabeled data at the same time. Recent semi-supervised approaches focus on deep neural networks and have achieved promising results on several benchmarks: CIFAR10, CIFAR100 and SVHN. However, most of their experiments are based on models trained from scratch instead of pre-trained models. On the other hand, transfer learning has demonstrated its value when the target domain has limited labeled data. Here comes the intuitive question: is it possible to incorporate SSL when fine-tuning a pre-trained model? We comprehensively study how SSL methods starting from pretrained models perform under varying conditions, including training strategies, architecture choice and datasets. From this study, we obtain several interesting and useful observations.

While practitioners have had an intuitive understanding of these observations, we do a comprehensive emperical analysis and demonstrate that: (1) the gains from SSL techniques over a fully-supervised baseline are smaller when trained from a pre-trained model than when trained from random initialization and (2) when the domain of the source data used to train the pre-trained model differs significantly from the domain of the target task, the gains from SSL are significantly higher.

We hope our studies can deepen the understanding of SSL research and facilitate the process of developing more effective SSL methods to utilize pre-trained models.

1. Introduction

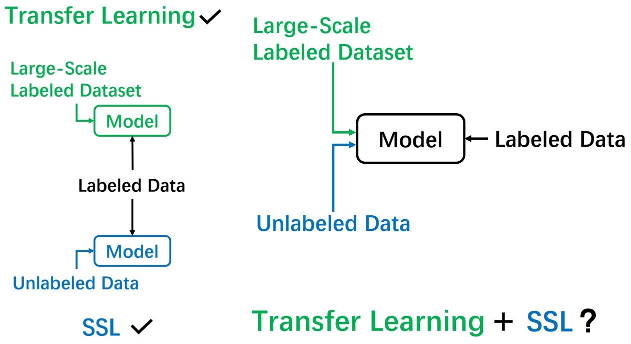
Deep Neural Networks have been found to be quite effective for solving problems in the domain of computer vision [6, 7, 29, 32, 33, 10]. One main reason is that deep models can "digest" large-scale labeled dataset, which was quite difficult using previous approaches. However, building a large image dataset can be very tedious and costly. Moreover, there are cases that image labels need expert experience and special devices. For example, medical images should be labeled by experienced doctors using specific medical instruments. Even then, some of the produced labels can be unreliable.

As people always want better performance for free, the community started to focus on how to make use of unlabeled images which are cheap and plentiful. Semi-Supervised Learning (SSL) is born with the ambition to learn from both labeled and unlabeled data simultaneously[1]. By introducing unlabeled data to the learning process, SSL is able to exploit the regularity hidden in the data. When combined with traditional supervised methods, SSL based approaches have shown its ability to enhance performance without importing noticeable supervision.

Another technique for improving on training only with labeled examples, transfer learning, has been widely employed in various settings, especially computer vision related tasks. Unlike SSL, transfer learning is good at tackling learning problems arised in different domains. Thanks to the popularity of vast image datasets, e.g., the ImageNet [6] and Places [43] dataset, the source domain is often large enough to share similar representations with the target domain in both low-level and high-level features. Thus models pre-trained on these datasets often have good initialization, and which enables them to surpass models trained from scratch in various open problems and competitions [18, 27, 38, 17, 9].

If we come to the topic of representation learning, it is not clear whether combining SSL and transfer learning would lead to learning better features. Then it is natural to wonder: would SSL benefit from transfer learning or would transfer learning from a large labeled dataset learn such a good classifier, such that SSL would offer no additional im-

[1] In this paper, we only focus on the applications of SSL in computer vision. Though SSL has shown its strength in other areas, we will leave the task of describing those to other papers.



(a) (b)

Figure 1: We present our core idea in this figure. From Figure 1a, we can see that the main difference between SSL and transfer learning is: SSL makes use of unlabeled images to facilitate the learning process in target domain, while transfer learning uses vast labeled dataset to learn generalizable representations in source domain. Since both two methods address learning better features for the target task, we want to check if they would have a conflict in real-world applications (as shown in Figure 1b).

provement? In fact, many recent works have demonstrated the effectiveness of SSL or transfer learning. But few studies tried to integrate both of them (as shown in Figure 1). In this paper, we are trying to understand the effectiveness of SSL on top of transfer learning. We fine-tune pre-trained models using existing SSL approaches under varying training strategies, models and datasets. We report some interesting observations appeared in our experiments.

Our contributions can be summarized as follows:

- We perform detailed experiments and discover that gains from modern SSL methods are smaller in settings where pre-training from a similar domain works well. Under many such experimental settings the gains entirely disappear.
- SSL-based algorithms work the best when the source domain is quite different from the target domain. Our experimental results show that SSL has great potential in processing medical images, where pre-training from ImageNet doesn't seem to lead to any significant improvements while SSL does.
- We find that Pseudo-Label with Adam as its optimizer, is surprisingly effective when you have enough labeled images. This fact gives the evidence that Pseudo-Label may be underestimated in previous literature. We caveat this point by pointing out that Oliver *et al.* [20] find that Pseudo-Label worked poorly when training models from scratch, so additional experiments may be warranted here to tease apart the differences.

2. Related Work

Both semi-supervised learning and transfer learning are hot topics in the computer vision and machine learning communities. There are many classical approaches in these two areas, such as graph-based approaches [45] and transductive SVM [12] in SSL, self-taught learning [24] and graph transfer method [4] in transfer learning. Due to the lack of space, we only focus attention on widely adopted approaches with deep neural networks: Π-model [14], Mean Teacher [35], Virtual Adversarial Training [19], Pseudo-Label [15], and fine-tuning.

2.1. Semi-Supervised Learning

Many methods have been proposed to tackle semi-supervised learning, we only introduce the most related ones. For example, we will not provide a comprehensive review of graph-based methods as they have not been widely accepted on deep models. Recent studies in semi-supervised learning mainly lie in consistency-regularization and Generative Adversarial Networks (GANs). In this section, we also introduce entropy-based approaches and co-training.

Consistency-Based Method. Based on the network architecture presented in [36], Rasmus *et al.* [26] proposed a model which is trained to simultaneously minimize the sum of supervised and unsupervised reconstruction cost functions by back-propagation. Laine and Aila *et al.* [14] then simplified the ladder network to Π-model and introduced self-ensembling, a consensus prediction of the unknown labels using an exponential moving average of outputs of the network-in-training on different epochs. Tarvainen and Valpola [35] developed a method, named Mean Teacher, which keeps exponential moving averages of model weights instead of the self-ensembling mentioned above.

Recently, Chen *et al.* [3] proposed a method capable of exploiting the memory of model and introduced a memory mechanism into the network training process.

SSL using GANs and Adversarial Training. The objective of GAN is to generate visually realistic images, which seems to be a very suitable choice for SSL as these images can be taken as additional training data. Springenberg [30] presented a method for learning a discriminative classifier from unlabeled or partially labeled data, which can be interpreted as the first attempt trying to apply GANs to SSL. Salimans *et al.* [28] improved the techniques for training GANs and showed how a discriminator that also predicts classes can be used for SSL. Dai *et al.* [5] gave the definition of a preferred generator and derived a new formulation for improving previous feature matching GANs. Li *et al.* [16] pointed out that a single discriminator only estimates the data without considering the labels and proposed Tri-GANs to address this problem.

With respect to adversarial training, Miyato *et al.* [19]

modified the training course and proposed a new regularization method based on virtual adversarial loss. Park *et al.* [22] then introduced a minimal set of dropouts, called adversarial dropout, to further improve the performance of virtual adversarial training.

Entropy-Based SSL. Grandvalet and Bengio [8] considered entropy minimization as a regularizer to incorporate unlabeled data. Lee [15] proposed a simple method, called Pseudo-Label, which just picks up high-confidence predictions which are iteratively added to the labeled training set.

Co-Training. Co-Training, first proposed by Blum and Mitchell [1], utilizes the diversity between two classifiers and let them label unlabeled data for each other. Zhou and Li [44] presented Tri-Trainig to use bootstrap sampling to get three different training sets and generates three classifiers from these three training sets respectively. For deep models, Chen [2] tried to build Tri-Net to combine tri-training with deep models.

In this paper, we choose to evaluate SSL methods based on consistency-regularization, adversarial training and entropy-based methods. The reason why we ignore GANs series is that pre-trained models are *not* widely used in such methodologies. To keep pace with [20], we mainly perform experiments on Π model, Mean Teacher, VAT and Pseudo-Label.

2.2. Transfer Learning

According to [21] and [34], research on transfer learning can be divided into different categories using different rules. In this paper, we mainly focus on "Network-based deep transfer learning", a concept proposed in Tan *et al.* [34], where they make such classification based on the techniques in transfer learning.

In our experiments (see Section 5), we use fine-tuning as our main method of transfer learning. This is because fine-tuning is the most commonly used technique under various datasets and network architectures. Yosinski *et al.* [41] provided a thorough study about the fine-tuning performance across different network layers and varying image classes. As for the reason why we choose to fine-tune all layers, we argue that this fits the setting of SSL: training numerous parameters with the help from unlabeled data.

In the rest of this paper, we adhere to a similar idea proposed in [41] except that we incorporate SSL into the fine-tuning process. Our work shares some similarities with a recent evaluation paper on SSL [20], in which the authors made a comprehensive study about the performance of SSL on real-world applications. But their experiments were mostly based on models trained from scratch and reported few results about fine-tuning a pre-trained model under various conditions (e.g., different datasets and model architectures). In this paper, we expand their analysis to the combination of fine-tuning and SSL.

3. Revisiting SSL Methods

Since there are so many SSL algorithms, it is necessary to select several representatives so as to perform further experiments. Oliver *et al.* [20] chose two consistency-regularization (or so called smooth regularization in [5] and [19]) method: Π-model [14] and Mean Teacher [35], one adversarial training approach: Virtual Adversarial Training (VAT) [19] and the Pseudo-Label [15] as experimental subjects. We think this algorithm pool sounds reasonable and adopt this setting in our studies.

In this section, we show that both VAT and Pseudo-Label may also be categorized as consistency-regularization SSL methods. From a view of smooth regularization, no matter labeled images or unlabeled ones are able to facilitate the process of learning better representations.

3.1. Preliminary

We use D_{L} to denote a set of labeled images, and D_{UL} to denote a set of unlabeled images. The original input image is x_i. To clarify the difference between labeled and unlabeled image, we use the following convention: The first elements in D are the labeled images $\{x_{1,2,\ldots,\text{n}}\} \in D_{\text{L}}$ and the later elements in D are the unlabeled images $\{x_{\text{n+1,n+2,...,n+m}}\} \in D_{\text{UL}}$. The number of all images is N, which equals $n + m$.

In consistency-based SSL methods, we also have some transformed (or corrupted) images. We denote the transformed image as $\tilde{x}_i$. Note that $\tilde{x}_i$ comes from x_i and hence $\tilde{x}_i$ can be written as $g(x_i, d_i)$, where $g(\cdot,\cdot)$ is a perturbation function and d_i represents a probability distribution for x_i.

When d_i is independent of x_i, $g(x_i, d_i)$ can be written as $g(d_i)$. For example, if d_i obeys uniform distribution, $x_i * g(d_i)$ can be interpreted as horizontal flip. When $g(\cdot,\cdot)$ is a clip function and d_i submits to Bernoulli distribution, $x_i * g(d_i)$ represents random translation. Since dropout [31] is also popular in SSL approaches, we can add a superscript to d_i which then becomes $d_i^{(l)}$, where l is the number of a specific layer and $d_i^{(l)}$ follows Bernoulli distribution. So the dropout can be formalized as $x_i^{(l)} * g(d_i^{(l)})$ where $x_i^{(l)}$ is the input of layer l. However, for simplicity, we ignore dropout in following equations.

The whole network can be described as producing an output class probability distribution $f_\theta(\cdot)$, where θ represents the network parameters, and we denote the number of classes by k.

We also use $L_{\text{CE}}[\cdot,\cdot]$ to represent cross-entropy loss, while $L_{\text{MSE}}[\cdot,\cdot]$ stands for the mean square error loss.

3.2. Π Models

Π model can be seen as a simpler version of Γ model [26]. It removes the layer-wise noise and replaces the noise with input perturbation, which mainly includes random translation and horizontal flip. The cost function of Π model is:

$$\begin{aligned} L_{\Pi}(D_{\mathrm{L}}, D_{\mathrm{UL}}) = & \sum_{i=1}^{N} L_{\mathrm{MSE}}[f_{\theta}(x_i * g(d_i)), f_{\theta}(x_i * g(\hat{d_i}))] \\ & + \quad \alpha \sum_{i=1}^{n} L_{\mathrm{CE}}[f_{\theta}(x_i * g(d_i)), y_i] \end{aligned} \tag{1}$$

In the formula above, we use $\hat{d_i}$ to denote that we generate a similar distribution to construct corrupted input $\tilde{x_i}$.

3.3. Mean Teacher Model

The outputs of Γ and Π models are noisy during the initial training stage. Laine and Aila [14] presented "Temporal Ensembling" to split Π model into two different networks $f_{\theta}(\cdot)$ and $f_{\theta'}(\cdot)$. This method accumulates outputs of each checkpoint (every epoch) and update predictions from $f_{\theta'}(\cdot)$ by keeping exponential moving averages of the model parameter values. Tarvainen and Valpola [35] then made some modifications by simply substituting moving-average model weights for ensembled predictions. To incorporate such Mean Teacher method into formula (1), we import a subscript t into θ, so the loss function becomes:

$$\begin{aligned} L_{\mathrm{MT}}(D_{\mathrm{L}}, D_{\mathrm{UL}}) = & \sum_{i=1}^{N} L_{\mathrm{MSE}}[f_{\theta_t}(x_i * g(d_i)), f_{\theta_t^{'}}(x_i * g(\hat{d_i}))] \\ & + \quad \alpha \sum_{i=1}^{n} L_{\mathrm{CE}}[f_{\theta_t}(x_i * g(d_i)), y_i] \end{aligned} \tag{2}$$

where $\theta_t^{'} = \beta\theta_{t-1}^{'} + (1-\beta)\theta_t$, t stands for the training step and β is a decay factor.

3.4. VAT and Pseudo-Label

Different from aforementioned techniques which improve SSL from a perspective of model construction, these two approaches mainly operate on $g(x_i, d_i)$. VAT [19] proposed to choose adversarial perturbations that maximally change the predictions made by the model. In this setting, d_i depends on x_i. Hence, we rewrite $g(d_i)$ to $g(d_i|x_i)$, where $d_i|x_i$ means the distribution depends on the input image. And $L_{\mathrm{VAT}}(\cdot,\cdot)$ is summarized as:

$$\begin{aligned} L_{\mathrm{VAT}}(D_{\mathrm{L}}, D_{\mathrm{UL}}) = & \sum_{i=1}^{N} L_{\mathrm{MSE}}[f_{\theta}(x_i), f_{\theta}(x_i * g(d_i|x_i))] \\ & + \quad \alpha \sum_{i=1}^{n} L_{\mathrm{CE}}[f_{\theta}(x_i), y_i] \end{aligned} \tag{3}$$

Pseudo-Label tries to select high-confidence predictions as the groundtruth label where "high-confidence" can be *considered as a type of perturbation operation* on the output of $f_{\theta}(\cdot)$. While other perturbation make use of every unlabeled image, "high-confidence" only utilize a portion of unlabeled data. The loss criterion is as follows:

$$\begin{aligned} L_{\mathrm{PL}}(D_{\mathrm{L}}, D_{\mathrm{UL}}) = & \sum_{i=(n+1)}^{N} L_{\mathrm{CE}}[f_{\theta_t}(x_i), f_{\theta_{t-1}}(x_i) * g(d_i)] \\ & + \quad \alpha \sum_{i=1}^{n} L_{\mathrm{CE}}[f_{\theta_t}(x_i), y_i] \end{aligned} \tag{4}$$

where d_i can be regarded as a simple binary mask with a specific threshold. The process of using pseudo-label as supervision can be considered as a $L_{\mathrm{CE}}(\cdot,\cdot)$ regularization between different time steps. Note that for clarify, we don't follow the mini-batch manner. Instead, we update the network using all samples at each time step. By now, we are able to gather all these four methods into a consistency-regularization manner.

4. Training Strategies, Models and Datasets

Oliver *et al.* [20] introduced a shared implementation of SSL algorithms and studied the performance of SSL under different data conditions, like distribution mismatch between labeled and unlabeled data, amount of labeled and unlabeled data, influence of having ImageNet for pre-training, and more. However, in this section, we extend the analysis to reveal more details about the combination of SSL and fine tuning from pre-trained models.

Our experiments mainly focus on three aspects: training strategies, models and datasets. These factor are not only the foundation of machine learning but may also lead to different conclusions from SSL experiments. This section will briefly outline these points and detailed analysis will be offered in next section.

4.1. Training Strategies

We find that changes in training strategy may lead to huge differences in the insights gleaned from looking at the results of SSL training. In this section, we describe the factors that impacted the intepretation of our results. More details will be reported in the following section.

(a) **Optimizer**. We tested each setting with both Adam [13] and SGD with Momentum.

(b) **Amount of labeled data**. The number of labeled images is quite influential as reported in [20]. We studied different numbers of labeled images across different datasets to check the influence those settings had on the final experimental results.

(c) **Input Perturbation**. As mentioned in Section 3, manually designed perturbations are the basis of some of the consistency regularization SSL techniques we study, namely Π-model and Mean Teacher. We tested these SSL approaches under different perturbation methods to test how the experimental results depend on the perturbations used.

Table 1: We report Top-1 and Top-5 Accuracy on ImageNet, floating point operations per second (FLOPs) and number of parameters in this table. Convolutional operations account for FLOPs. Inception-v3 has the highest Top-1 record while VGG-16 has the most parameters.

Network	Top-1	Top-5	FLOPs	Params.
ResNet-50	75.2	92.2	8B	26M
VGG-16	71.5	89.8	32B	138M
Inception-v3	78.0	93.9	12B	23M

(d) **Training Iterations**. We find that SSL methods differ in how they perform when evaluated at different steps through the their training process. Some methods converge faster than others.

We perform ablative experiments on these factors to show their influence on SSL. Note that we tune the baseline models and report results with appropriate learning rate and weight decay. More details can be found in Section 5.

4.2. Models

Model architecture plays a significant role in the performance of SSL algorithms. For fairness, [20] used an identical 28-layer wide-resnet [42] to perform evaluation of different SSL algorithms. Nevertheless, we'd like to take a step forward: assessing different SSL algorithms under different network architectures.

We give a brief summary of our candidates in Table 1. ResNet-50 [10] is a residual network where skip connects facilitate the training of deep models. VGG-16 [29] is a popular architecture with fully-connected layers. Though it does not have high accuracy (at present), it is still active in many technique reports, which merits our attention. Inception-v3 [33] achieves the best performance over ResNet-50 and VGG-16 on ImageNet and was designed to be light-weight. We select these three models because they can be regarded as three important branches of the development of model architecture design. By testing on various model architectures, we better understand which techniques work well in particular settings, and which can be expected to improve results across the board.

4.3. Datasets

As is commonly known, the gap between transfer learning and training from random initialization is highest when the target domain has a small number of labels. This overlaps with the setting where SSL is most helpful. Previous evaluation of SSL is done mostly on the MNIST, CIFAR and SVHN datasets. However, images in these datasets are usually small (less than 50×50 in resolution), making them unsuitable for ImageNet-based models. In this paper, we use pre-trained ImageNet models and fine-tune on three modern datasets: Indoor67 [23], CUB200 [37] and MURA [25]. Note that Indoor67 and CUB200 have some class overlaps with ImageNet, while MURA is a medical image dataset which lies in a completely different domain.

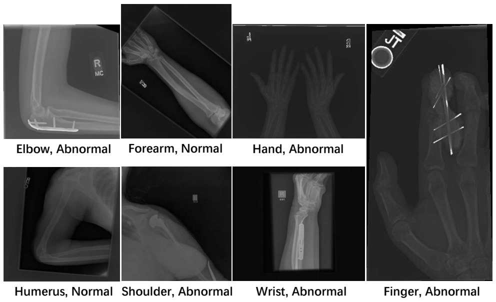


Figure 2: Samples from MURA dataset. We show X-Rays of seven different parts from human body.

Indoor67 is a scene-oriented dataset which is different from object-centered ImageNet. It contains 67 scene categories. There are 6,700 images in total, where each class has 80 images for training and 20 images for testing.

CUB200 is an object-oriented dataset with 200 species of natural birds. CUB200 includes 11,788 images and the official split for train/test is 5,994 vs. 5,794. Our experiments followed this convention.

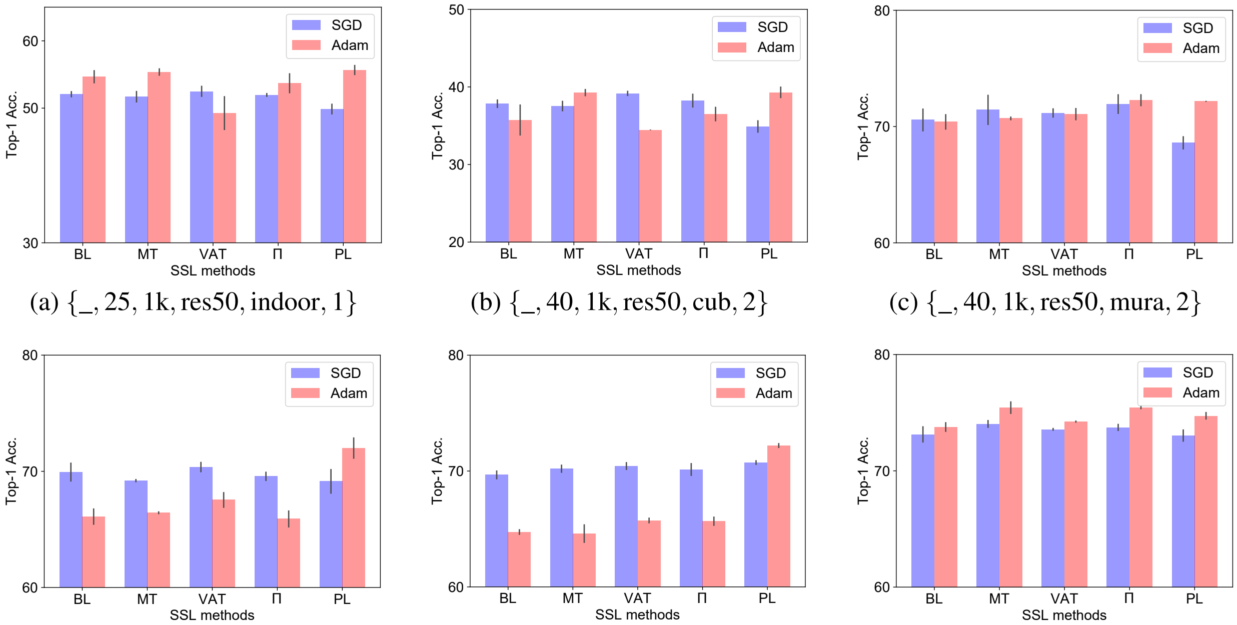
MURA is a dataset of musculoskeletal radiographs, which contains 40,561 images from 14,863 patient studies. X-Ray images are collected from seven parts of human body: elbow, finger, forearm, hand, humerus, shoulder and wrist (cf. Figure 2). The goal of this dataset is to distinguish normal musculoskeletal *studies* from abnormal ones (a study often contains more than one image). In this paper, we make a modification to this goal: to simply tell the difference between normal and abnormal *radiographs* (one image). The reason why we'd like to evaluate SSL on MURA is that we hope to check the value of SSL when ImageNet pre-trained models are not that useful.

5. Experiments

In this section, we mainly report the performance of SSL methods on pre-trained models. Before we start experiments, a few notations have to be introduced. For the abbreviation of SSL algorithms, we use Π to stand for Π model, `MT` for Mean Teacher, `VAT` for Virtual Adversarial Training, `PL` for Pseudo-Label and `BL` for fully-supervised method. In following experiments, we will make use of a set to denote hyperparameters during the training stage,

$$\{\text{Adam}, 20, 1\text{k}, \text{res50}, \text{indoor}, 3\}$$

- `Adam` means we employed Adam for optimization while `SGD` represents SGD optimizer.



(a) {_, 25, 1k, res50, indoor, 1}

(b) {_, 40, 1k, res50, cub, 2}

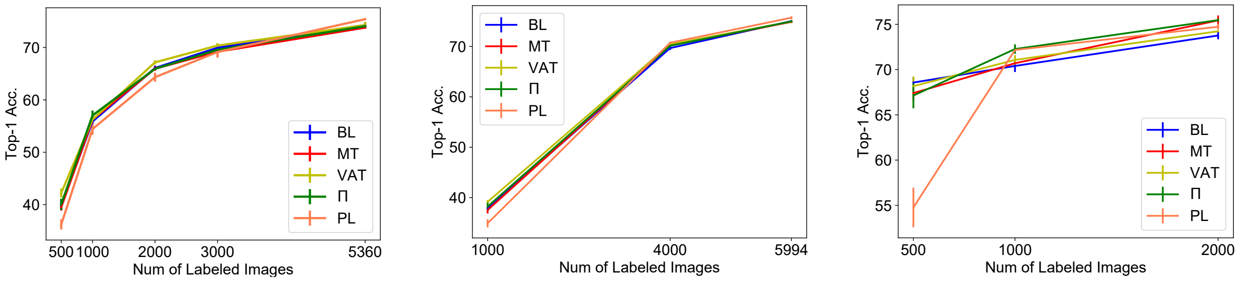
(c) {_, 40, 1k, res50, mura, 2}

(d) {_, 40, 3k, res50, indoor, 2}

(e) {_, 40, 4k, res50, cub, 2}

(f) {_, 40, 2k, res50, mura, 2}

Figure 3: We performed experiments on different optimizers. Two conclusions are straightforward: 1. Pseudo-Label with Adam optimizer achieves the best results. 2. SGD surpasses Adam when increasing the number of labeled images and training iterations.



(a) {SGD, 40, _, res50, indoor, 2}

(b) {SGD, 40, _, res50, cub, 2}

(c) {SGD, 40, _, res50, mura, 2}

Figure 4: Comparison between fully-supervised model and SSL algorithms with varying amount of labeled data.

- `20` denotes we trained both SSL methods and fully-supervised model for 20 epochs (similar strategy as in [20]).
- `1k` means we have one thousand labeled images while the rest are unlabeled.
- `indoor` stands for the fine-tuned dataset. The other two databases are `cub` and `mura`, respectively.
- The last `3` tells us how many augmentation methods we have used. `1` means we only used horizontal flip, `2` appends random translation, while `3` includes gaussian noise.

In summary, the above set means we trained all models (SSL and fully-supervised methods) based on Adam for 20 epochs using 1k labeled images. The pre-trained model is resnet-50 and we fine-tuned it on Indoor67 with 3 types of perturbation. At times, we refer to a class of experiments where only one setting varies – we denote the varying setting with an underscore. We set the Mean Teacher exponential moving average decay rate to 0.99. The learning rates of Adam and SGD are 1e-4 and 1e-3, respectively – we explain below how we standardizes on these. Learning rates are decayed by a factor of 10 at three quarters of the total training steps. The size of input image is 224 on ResNet-50 and VGG-16, and 299 on Inception-v3. Other hyperparameters follow the settings in [20].

The learning rates for Adam and SGD were chosen this way: We chose 1e-4 out of {5e-3, 1e-3, 5e-4, 1e-4, 1e-5} for

Adam, and 1e-3 from {5e-2, 1e-2, 5e-3, 1e-3, 1e-4}. Out of those learning rates, we chose the one that performed best on {_, 25, 1k, res50, indoor, 1} (see Figure 3a). Then we fixed those learning rates for all other experiments. While we acknowledge that more careful tuning for each task may yield different results, we aimed to simulate the experience of practicioners that may not have the resources needed to tune for every new dataset. This methodology is also the one used in [20], where they fixed hyperparameters for a fixed number of labeled examples and evaluated with those same hyperparameters on other SSL settings.

The Adam hyperparameters are kept as in [13]: we set β_1 to 0.9 and β_2 to 0.999 in Adam. The SGD momentum is kept at 0.9. Note that we don't tune the Momentum carefully as Wilson *et al.* [39] showed that initial step size and the step decay scheme heavily influence the final performance.

5.1. Adam versus SGD

We evaluated SSL approaches using different optimizers and display results in Figure 3. From this figure, we can see that without re-tuning for each setting, different SSL methods seem to work better with different optimizers. When comparing Figure 3a with 3d and 3b with 3e, it is easy tell that *simply increasing training iterations and amount of labeled data* may help SGD surpass Adam in all SSL approaches except Pseudo-Label.

In fact, if we look carefully at the six figures, we see that *Pseudo-Label with Adam* usually achieves the best results (sometimes quite close to the best) in all settings. This phenomenon is strange as Pseudo-Label got the worst performance on models trained from random initialization [20]. We guess the reason might be that Pseudo-Label acts as a regularizer by adding a certain form of noise into the training process. This characteristic makes Pseudo-Label unable to achieve satisfying performance on models trained from scratch but helps prevent overfitting in pre-trained models. More details can be found in the appendix.

Another fact is that if we temporarily ignore Pseudo-Label, *the other three SSL methods are comparable with simple baselines*. The "best" SSL candidate barely improves over the fully-supervised baseline. This phenomenon may complement the conclusion from [20] which tells us good pre-trained model could beat models trained from random initialization with SSL. However, our experiments demonstrate evidence towards the claim that **under domain gap, you should not expect too much improvement from SSL on pre-trained models even with lots of unlabeled images**.

When we turn to Figure 3f, we see that SSL offer more improvement when the target domain (radiograph) is different from the source domain (natural image). Nearly all SSL methods surpass the strong baseline where Mean Teacher and Π-model perform the best.

5.2. The Number of Labeled Images

In this part, we will use SGD as the default optimizer because SGD usually gets better accuracy when having more iterations and more labeled images as reported in Section 5.1. Figure 4 shows cross-dataset performance when increasing the number of labeled images.

Still, the difference between the fully-supervised baseline and SSL is small. In Figure 4a, when the number of labeled images increases, *Pseudo-Label gradually surpasses other methods including the baseline*. A similar phenomenon also appears in Figure 4b. We argue that Pseudo-Label may not be good at making use of large amount of unlabeled data but succeed over other SSL methods when there are enough labeled images. For example, in Figure 4a and 4b (see the rightmost datapoints in the graph, that correspond to training on the entire dataset), Pseudo-Label even surpasses fully-supervised models with no labeled images in all three datasets. Since we borrowed the code of Pseudo-Label from [20], we argue the reason of this phenomenon is that they used pseudo-label in a way that adds two loss terms for all labeled examples, one based on the true label and one based on the confident predictions when they exist, while traditional Pseudo-Label separates the labeled and unlabeled data entirely. This implementation may provide a regularization effect on fully-supervised models.

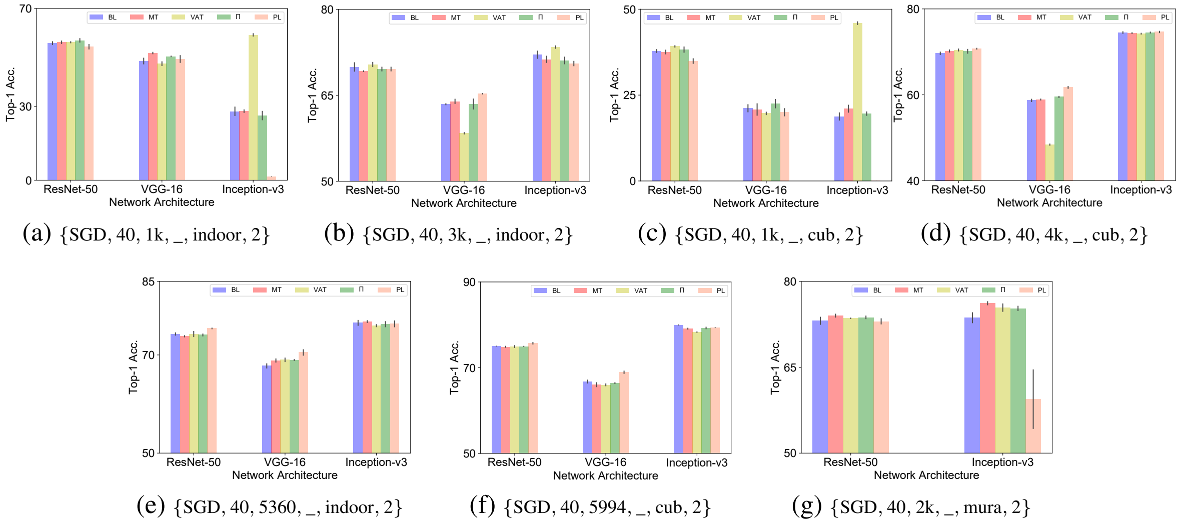
In Figure 4c, properties of Pseudo-Label become more clear. And we we can find that SSL methods surpass the fully-supervised baseline with different amounts of labeled data. This phenomenon shows us again that SSL improves the performance of pre-trained models when there is a domain gap.

5.3. Models

Figure 5 shows our experiments on different network architectures across three datasets. We choose ResNet-50, VGG-16 and Inception-v3 pre-trained on ImageNet as our basis. As before, for each of these architectures, we fine-tune on Indoor67, CUB200 and MURA, with and without SSL methods. For consistency with those experiments above, we also studied their performance over varying amounts of labeled data.

Figure 5a and 5c show that *fine-tuning from Inception-v3 trained on ImageNet to other SSL datasets with little labeled data using VAT works very well*. Though Inception-v3 has inferior transfer ability when there is no unlabeled data present, or with other SSL techniques, VAT still helps it get the best performance, defeating ResNet-50 and VGG-16. This points to using Inception series models as a good starting point for fine-tuning with unlabeled data and VAT.

When we increase the amount of labeled data, the superiority of Inception-v3 becomes apparent. If the ratio of



(a) {SGD, 40, 1k, _, indoor, 2} (b) {SGD, 40, 3k, _, indoor, 2} (c) {SGD, 40, 1k, _, cub, 2} (d) {SGD, 40, 4k, _, cub, 2}

(e) {SGD, 40, 5360, _, indoor, 2} (f) {SGD, 40, 5994, _, cub, 2} (g) {SGD, 40, 2k, _, mura, 2}

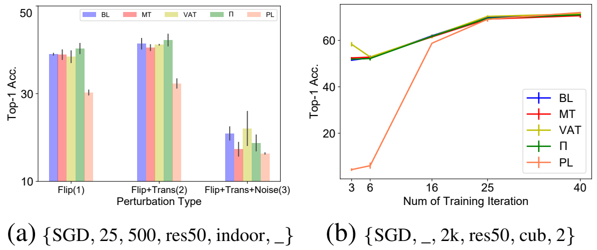
Figure 5: SSL algorithms are evaluated on different pre-trained models and datasets. We also increase the amount of labeled data to ensure the reliability of these experiments.

labeled images to unlabeled images is roughly two-thirds (cf. Figure 5b and 5d) or even 100% (cf. Figure 5e and 5f), Inception-v3 win the first place among three models. An important fact is that *when you have enough labeled images, better pre-trained models seem to counteract the influence of SSL methods*. When we talk about VGG-16, it is easy to summarize that SSL methods help a lot. In Figure 5b-5f, SSL achieves better results than fully-supervised baseline. If we come to ResNet-50, this improvement shrinks but still exists. Finally, when we look at Inception-v3 in Figure 5b, 5d, 5e and 5f, we improvement is gone. Furthermore, fully-supervised method nearly beats all SSL approaches. We are not certain if this means SSL is useless on a good pre-trained model (Inception-v3 scores the best on ImageNet), but this phenomenon can be a good clue to evaluate SSL methods on other well-trained models, such as ResNeXt [40] and SE-Net [11]. However, it is undeniable that SSL performs well when there is a small number of labeled images, or when the domain gap between source and target training is high.

From Figure 5g, we once again find that SSL algorithms work the best under a domain gap. Even on Inception-v3, Mean Teacher still achieves a much better result than fully-supervised method which hints that SSL can be expected in the area of medical image processing.

5.4. Perturbation and Increased Training Iteration

As shown in the left figure in Figure 6, combining horizontal flip and random translation achieves the highest accuracy among three types of perturbation. On the other hand, simply adding augmentation like noise degenerates the performance. These are the reason why flip+trans (2) is the first choice in most of our experiments. We also discover that all SSL methods *agree on which perturbation leads to better performance*.



(a) {SGD, 25, 500, res50, indoor, _} (b) {SGD, _, 2k, res50, cub, 2}

Figure 6: Experiments on different types of perturbation and varying numbers of training iteration.

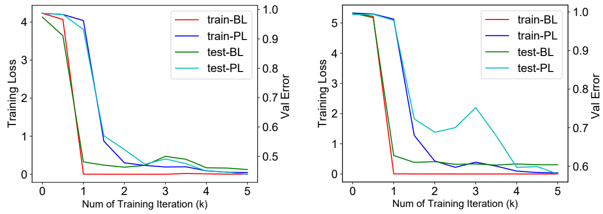
Figure 6b suggests *increasing the number of training iteration will close the gap* among different SSL methods. When models are trained for less than 10 epochs, both VAT and Pseudo-Label have unstable performance. However, if we increase the number of training steps, their performances gradually converge.

6. Conclusion

In this work, we identify the relationship between SSL and transfer learning. We provide a view that most SSL algorithms can be regarded as a type of smooth regularization, which hints that recent SSL methods may help improve fully-supervised models. To verify the effectiveness of SSL on pre-trained models, we perform detailed experiments under various conditions. The experiments provide several meaningful observations including the effectiveness of SSL on fully supervised methods. Different SSL algorithms usually display different performance under various conditions. However, we find that SSL algorithms are quite useful on medical images. This phenomenon suggests that domain gap might be the place where SSL and transfer learning combined can yield the most improvements.

7. Supplementary

7.1. The regularization effect of Pseudo-Label



(a) {adam, 40, 1k, res50, indoor, 2} (b) {adam, 40, 1k, res50, cub, 2}

Figure 7: We report the training and validation details of BL and PL. The horizontal axis represents the number of training iteration and the unit is k (thousand). The training loss means the loss on labeled images.

In order to understand the regularization effect of Pseudo-Label, we show the performance of both training and validation process in Figure 7. The experiments are conducted on Indoor67 and CUB200. We split the original training set (5360 for Indoor67, 5994 for CUB200) into two parts: train and valid and report the training loss and valid errors. The valid set of each dataset contains 1k images. From Figure 7a, we can see that fully-supervised model (BL) achieves lower training loss when compared with Pseudo-Label (PL). In contrast, PL scores lower validation error after $3k$ iterations. If we focus on CUB200 (Figure 7b), it is obvious that BL converges faster and better on the training set while PL gets a lower validation error on the valid set. These phenomenons hint that *PL may serve as a regularizer and helps prevent overfitting when there is limited labeled data.*

7.2. Hyperparameters of VAT

Since VAT has two hyperparameters: ϵ and ξ, we try to study the influence of them on SSL performance. We report the validation performance of VAT on the same valid set as mentioned in Section 7.1. The default values of ϵ and ξ are 6 and 1e-6, respectively. From Figure 8, we find that simply decreasing ϵ (red bar) or increasing ξ (green bar) would have positive effects on Indoor67. However, when we come to CUB200, the default values achieve comparable results. To keep pace with previous studies, we decide to maintain the default choices of ϵ (6) and ξ (1e-6) in the paper.

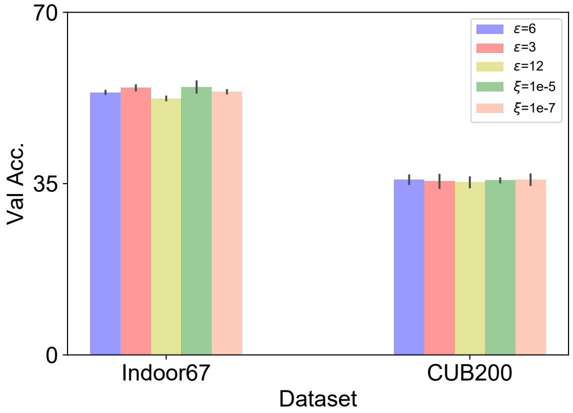


Figure 8: We compare the performance of VAT using different hyperparameters.

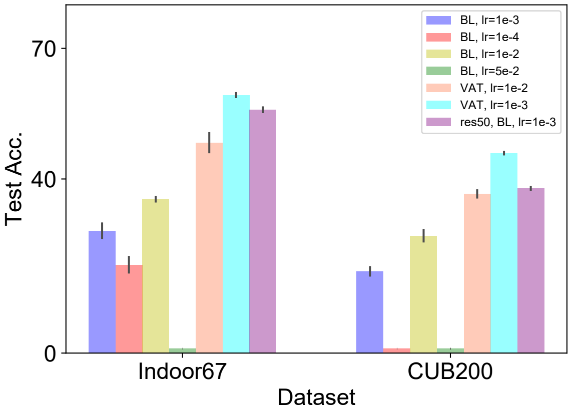


Figure 9: Influence of different learning rates on Inception-v3.

7.3. Learning rates on Inception-v3

It is easy to find that Inception-v3 has inferior performance when having limited labeled data (figure 5 (a) and (c) in the paper). We also conducted complementary ablative studies on this issue and find that learning rate plays an important role. Note that models are evaluated on the test set which is different from the valid set in Section 7.1 and 7.2. We report the experimental results in Figure 9. All training details follow the settings in figure 5 (a) and (c) (please refer to the paper for more details). From Figure 9, we can see that inappropriate choices of learning rates may heavily influence the performance of Inception-v3, such as lr=5e-2 (red bar) has bad results on both Indoor67 and CUB200. Also, increasing the learning rate to 1e-2 will enhance the model accuracy (comparing blue with yellow) but deteriorate VAT (comparing coral with aqua).

It is worth noting that there are several reasons that make us to report the results of training Inception-v3 using lr=1e-3 in figure 5 (cf. our paper). Firstly, fully-supervised

Inception-v3 with lr=1e-2 (yellow bar) still cannot defeat res50 (bar in purple). Next, Inception-v3 achieves satisfying results using lr=1e-3 with increased labeled images (refer to figure 5 (b) and (d-g)). Considering the experimental consistency with ResNet-50 and VGG-16, we decide to perform experiments using lr=1e-3.

References

[1] A. Blum and T. Mitchell. Combining Labeled and Unlabeled Data with Co-Training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100. ACM, 1998. 3

[2] D.-D. Chen, W. Wang, G. Wei, and Z.-H. Zhou. Tri-net for Semi-Supervised Deep Learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018. 3

[3] Y. Chen, X. Zhu, and S. Gong. Semi-supervised deep learning with memory. In *The European Conference on Computer Vision*, pages 3546–3554, 2015. 2

[4] W. Dai, O. Jin, G.-R. Xue, Q. Yang, and Y. Yu. Eigen-Transfer: A Unified Framework for Transfer Learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 193–200. ACM, 2009. 2

[5] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. R. Salakhutdinov. Good Semi-Supervised Learning that Requires a Bad Gan. In *Advances in Neural Information Processing Systems*, pages 6510–6520, 2017. 2, 3

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: a Large-Scale Hierarchical Image Database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009. 1

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 1

[8] Y. Grandvalet and Y. Bengio. Semi-Supervised Learning by Entropy Minimization. In *Advances in Neural Information Processing Systems*, pages 529–536, 2005. 3

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision*, pages 2980–2988. IEEE, 2017. 1

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 5

[11] J. Hu, L. Shen, and G. Sun. Squeeze-and-Excitation Networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. 8

[12] T. Joachims. Transductive Inference for Text Classification using Support Vector Machines. In *International Conference on Machine Learning*, volume 99, pages 200–209, 1999. 2

[13] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4, 7

[14] S. Laine and T. Aila. Temporal Ensembling for Semi-Supervised Learning. *arXiv preprint arXiv:1610.02242*, 2016. 2, 3, 4

[15] D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, 2013. 2, 3

[16] C. Li, K. Xu, J. Zhu, and B. Zhang. Triple Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, pages 4088–4098, 2017. 2

[17] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear CNN Models for Fine-Grained Visual Recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1449–1457, 2015. 1

[18] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 1

[19] T. Miyato, S.-i. Maeda, S. Ishii, and M. Koyama. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 2, 3, 4

[20] A. Oliver, A. Odena, C. Raffel, E. D. Cubuk, and I. J. Goodfellow. Realistic Evaluation of Deep Semi-Supervised Learning Algorithms. *Advances in Neural Information Processing Systems*, 2018. 2, 3, 4, 5, 6, 7

[21] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. 3

[22] S. Park, J.-K. Park, S.-J. Shin, and I.-C. Moon. Adversarial Dropout for Supervised and Semi-supervised Learning. In *The 32th AAAI Conference on Artificial Intelligence*, 2018. 3

[23] A. Quattoni and A. Torralba. Recognizing Indoor Scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420. IEEE, 2009. 5

[24] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-Taught Learning: Transfer Learning from Unlabeled Data. In *Proceedings of the 24th International Conference on Machine Learning*, pages 759–766. ACM, 2007. 2

[25] P. Rajpurkar, J. Irvin, A. Bagul, D. Ding, T. Duan, H. Mehta, B. Yang, K. Zhu, D. Laird, R. L. Ball, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng. MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs. *arXiv preprint arXiv:1712.06957*, 2017. 5

[26] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-Supervised Learning with Ladder Networks. In *Advances in Neural Information Processing Systems*, pages 268–283, 2018. 2, 3

[27] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks . In *Advances in Neural Information Processing Systems*, pages 91–99, 2015. 1

[28] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved Techniques for Training Gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016. 2

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *arXiv preprint arXiv:1409.1556*, 2014. 1, 5

[30] J. T. Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*, 2015. 2

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 3

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 1

[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 1, 5

[34] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu. A Survey on Deep Transfer Learning. In *International Conference on Artificial Neural Networks*, pages 270–279. Springer, 2018. 3

[35] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, pages 1195–1204, 2017. 2, 3, 4

[36] H. Valpola. From Neural PCA to Deep Unsupervised Learning. In *Advances in Independent Component Analysis and Learning Machines*, pages 143–171. Elsevier, 2015. 2

[37] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 5

[38] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In *The European Conference on Computer Vision*, pages 20–36. Springer, 2016. 1

[39] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht. The Marginal Value of Adaptive Gradient Methods in Machine Learning. In *Advances in Neural Information Processing Systems*, pages 4148–4158, 2017. 7

[40] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated Residual Transformations for Deep Neural Networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 5987–5995. IEEE, 2017. 8

[41] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How Transferable are Features in Deep Neural Networks? In *Advances in Neural Information Processing Systems*, pages 3320–3328, 2014. 3

[42] S. Zagoruyko and N. Komodakis. Wide Residual Networks. In *The 27th British Machine Vision Conference (BMVC)*, 2016. 5

[43] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2017. 1

[44] Z.-H. Zhou and M. Li. Tri-training: Exploiting Unlabeled Data Using Three Classifiers. *IEEE Transactions on knowledge and Data Engineering*, 17(11):1529–1541, 2005. 3

[45] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-Supervised Learning using Gaussian Fields and Harmonic Functions. In *Proceedings of the 20th International Conference on Machine Learning*, pages 912–919, 2003. 2