Borrowing Treasures from the Wealthy: Deep Transfer Learning through Selective Joint Fine-Tuning

Weifeng Ge   Yizhou Yu
Department of Computer Science, The University of Hong Kong

Abstract

Deep neural networks require a large amount of labeled training data during supervised learning. However, collecting and labeling so much data might be infeasible in many cases. In this paper, we introduce a deep transfer learning scheme, called selective joint fine-tuning, for improving the performance of deep learning tasks with insufficient training data. In this scheme, a target learning task with insufficient training data is carried out simultaneously with another source learning task with abundant training data. However, the source learning task does not use all existing training data. Our core idea is to identify and use a subset of training images from the original source learning task whose low-level characteristics are similar to those from the target learning task, and jointly fine-tune shared convolutional layers for both tasks. Specifically, we compute descriptors from linear or nonlinear filter bank responses on training images from both tasks, and use such descriptors to search for a desired subset of training samples for the source learning task.

Experiments demonstrate that our deep transfer learning scheme achieves state-of-the-art performance on multiple visual classification tasks with insufficient training data for deep learning. Such tasks include Caltech 256, MIT Indoor 67, and fine-grained classification problems (Oxford Flowers 102 and Stanford Dogs 120). In comparison to fine-tuning without a source domain, the proposed method can improve the classification accuracy by 2% - 10% using a single model. Codes and models are available at https://github.com/ZYYSzj/
Selective-Joint-Fine-tuning.

1. Introduction

Convolutional neural networks (CNNs) have become deeper and larger to pursue increasingly better performance on classification and recognition tasks [22, 18, 40, 12, 46]. Looking at the successes of deep learning in computer vison, we find that a large amount of training or pre-training data is essential in training deep neural networks.

Large-scale image datasets, such as the ImageNet ILSVRC dataset [36], Places [50], and MS COCO [23], have led to a series of breakthroughs in visual recognition, including image classification [24], object detection [10], and semantic segmentation [25]. Many other related visual tasks have benefited from these breakthroughs.

Nonetheless, researchers face a dilemma when using deep convolutional neural networks to perform visual tasks that do not have sufficient training data. Training a deep network with insufficient data might even give rise to inferior performance in comparison to traditional classifiers fed with handcrafted features. Fine-grained classification problems, such as Oxford Flowers 102 [29] and Stanford Dogs 120 [19], are such examples. The number of training samples in these datasets is far from being enough for training large-scale deep neural networks, and the networks would become overfit quickly.

Solving the overfitting problem for deep convolutional neural networks on learning tasks without sufficient training data is challenging [39]. Transfer learning techniques that apply knowledge learnt from one task to other related tasks have been proven helpful [30]. In the context of deep learning, fine-tuning a deep network pre-trained on the ImageNet or Places dataset is a common strategy to learn task-specific deep features.This strategy is considered a simple transfer learning technique for deep learning. However, since the ratio between the number of learnable parameters and the number of training samples still remains the same, fine-tuning needs to be terminated after a relatively small number of iterations; otherwise, overfitting still occurs.

In this paper, we attempt to tackle the problem of training deep neural networks for learning tasks that have insufficient training data. We adopt the source-target joint training methodology [45] when fine-tuning deep neural networks. The original learning task without sufficient training data is called the target learning task, $\mathcal{T}_t$. To boost its performance, the target learning task is teamed up with another learning task with rich training data. The latter is called the source learning task, $\mathcal{T}_s$. Suppose the source learning task has a large-scale training set $\mathcal{D}_s$, and the target learning task has a small-scale training set $\mathcal{D}_t$. Since the target

learning task is likely a specialized task, we envisage the image signals in its dataset possess certain unique low-level characteristics (e.g. fur textures in Stanford Dogs 120 [19]), and the learned kernels in the convolutional layers of a deep network need to grasp such characteristics in order to generate highly discriminative features. Thus supplying sufficient training images with similar low-level characteristics becomes the most important mission of the source learning task. Our core idea is to identify a subset of training images from $\mathcal{D}_s$ whose low-level characteristics are similar to those from $\mathcal{D}_t$, and then jointly fine-tune a shared set of convolutional layers for both source and target learning tasks. The source learning task is fine-tuned using the selected training images only. Hence, this process is called *selective joint fine-tuning*. The rationale behind this is that the unique low-level characteristics of the images from $\mathcal{D}_t$ might be overwhelmed if all images from $\mathcal{D}_s$ were taken as training samples for the source learning task.

How do we select images from $\mathcal{D}_s$ that share similar low-level characteristics as those from $\mathcal{D}_t$? Since kernels followed with nonlinear activation in a deep convolutional neural network (CNN) are actually nonlinear spatial filters, to find sufficient data for training high-quality kernels, we use the responses from existing linear or nonlinear filter banks to define similarity in low-level characteristics. Gabor filters [28] form an example of a linear filter bank, and the complete set of kernels from certain layers of a pre-trained CNN form an example of a nonlinear filter bank. We use histograms of filter bank responses as image descriptors to search for images with similar low-level characteristics.

The motivation behind selecting images according to their low-level characteristics is two fold. First, low-level characteristics are extracted by kernels in the lower convolutional layers of a deep network. These lower convolutional layers form the foundation of an entire network, and the quality of features extracted by these layers determines the quality of features at higher levels of the deep network. Sufficient training images sharing similar low-level characteristics could strength the kernels in these layers. Second, images with similar low-level characteristics could have very different high-level semantic contents. Therefore, searching for images using low-level characteristics has less restrictions and can return much more training images than using high-level semantic contents.

The above source-target selective joint fine-tuning scheme is expected to benefit the target learning task in two different ways. First, since convolutional layers are shared between the two learning tasks, the selected training samples for the source learning task prevent the deep network from overfitting quickly. Second, since the selected training samples for the source learning task share similar low-level characteristics as those from the target learning task, kernels in their shared convolutional layers can be trained

more robustly to generate highly discriminative features for the target learning task.

The proposed source-target selective joint fine-tuning scheme is easy to implement. Experimental results demonstrate state-of-the-art performance on multiple visual classification tasks with many fewer training samples than what is required by recent deep learning architectures. These visual classification tasks include fine-grained classification on Stanford Dogs 120 [19] and Oxford Flowers 102 [29], image classification on Caltech 256 [11], and scene classification on MIT Indoor 67 [33].

In summary, this paper has the following contributions:

- We introduce a new deep transfer learning scheme, called selective joint fine-tuning, for improving the performance of deep learning tasks with insufficient training data. It is an important step forward in the context of the widely adopted strategy of fine-tuning a pre-trained deep neural network.

- We develop a novel pipeline for implementing this deep transfer learning scheme. Specifically, we compute descriptors from linear or nonlinear filter bank responses on training images from both tasks, and use such descriptors to search for a desired subset of training samples for the source learning task.

- Experiments demonstrate that our deep transfer learning scheme achieves state-of-the-art performance on multiple visual classification tasks with insufficient training data for deep learning.

2. Related Work

Multi-Task Learning. Multi-task learning (MTL) obtains shared feature representations or classifiers for related tasks [3]. In comparison to learning individual tasks independently, features and classifiers learned with MTL often have better generalization capability. In deep learning, faster RCNN [34] jointly learns object locations and labels using shared convolutional layers but different loss functions for these two tasks. In [7], the same multi-scale convolutional architecture was used to predict depth, surface normals and semantic labels. This indicates that convolutional neural networks can be adapted to different tasks easily. While previous work [8, 34] attempts to find a shared feature space that benefits multiple learning tasks, the proposed joint training scheme in this paper focuses on learning a shared feature space that improves the performance of the target learning task only.

Feature Extraction and Fine-tuning. Off-the-shelf CNN features [37, 5] have been proven to be powerful in various computer vision problems. Pre-training convolutional neural networks on ImageNet [36] or Places [50] has been the standard practice for other vision problems. However, features learnt in pre-trained models are not tailored for the

(a) Source and Target Domain Training Data

(b) Search *k* Nearest Neighbors in Shallow Feature Space

(c) Deep Convolutional Neural Networks

(d) Joint Optimization in Different Label Spaces

Source Training Samples in Shallow Feature Spaces

Source Domain Data

Target Domain Data

Target Training Samples in Shallow Feature Spaces

Convolutional layers shared by the source and target learning tasks

Linear classifier

Sea Snake
Black Grouse
Chameleon
⋮
Dog
Barn Spider

Source Domain Loss Minimization

Linear classifier

Fire Lily
Clematis
Rose
⋮
Artichoke
Bee Balm

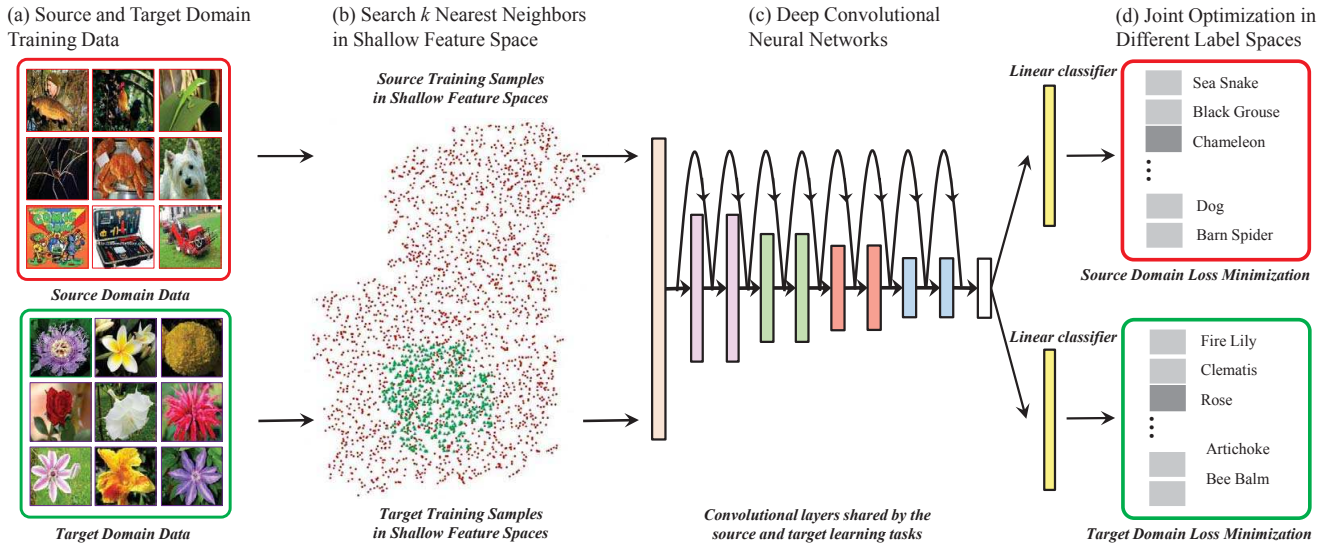*Target Domain Loss Minimization*

Figure 1. Pipeline of the proposed selective joint fine-tuning. From left to right: (a) Datasets in the source domain and the target domain. (b) Select nearest neighbors of each target domain training sample in the source domain via a low-level feature space. (c) Deep convolutional neural network initialized with weights pre-trained on ImageNet or Places. (d) Jointly optimize the source and target cost functions in their own label spaces.

target learning task. Fine-tuning pre-trained models [10] has become a commonly used method to learn task-specific features. The transfer ability of different convolutional layers in CNNs has been investigated in [48]. However, for tasks that do not have sufficient training data, overfitting occurs quickly during fine-tuning. The proposed pipeline in this paper not only alleviates overfitting, but also attempts to find a more discriminative feature space for the target learning task.

Transfer Learning. Different from MTL, transfer learning (or domain adaptation) [30] applies knowledge learnt in one domain to other related tasks. Domain adaptation algorithms can be divided into three categories, including instance adaption [16, 1], feature adaption [26, 41], and model adaption [6]. Hong *et al.* [15] transferred rich semantic information from source categories to target categories via the attention model. Tzeng *et al.* [41] performed feature adaptation using a shared convolutional neural network by transferring the class relationship in the source domain to the target domain. To make our pipeline more flexible, this paper does not assume the source and target label spaces are the same as in [41]. Different from the work in [1] which randomly resamples training classes or images in the source domain, this paper conducts a special type of transfer learning by selecting source training samples that are nearest neighbors of samples in the target domain in the space of certain low-level image descriptor.

Krause *et al.* [21] directly performed Google image search using keywords associated with categories from the target domain, and download a noisy collection of images

to form a training set. In our method, we search for nearest neighbors in a large-scale labeled dataset using low-level features instead of high-level semantic information. It has been shown in [27] that low-level features computed in the bottom layers of a CNN encode very rich information, which can completely reconstruct the original image. Our experimental results show that nearest neighbor search using low-level features can outperform that using high-level semantic information as in [21].

3. Selective Joint Fine-tuning

3.1. Overview

Fig. 1 shows the overall pipeline for our proposed source-target selective joint fine-tuning scheme. Given a target learning task $\mathcal{T}_t$ that has insufficient training data, we perform selective joint fine-tuning as follows. The entire training dataset associated with the target learning task is called the target domain. The source domain is defined similarly.

Source Domain : The minimum requirement is that the number of images in the source domain, $\mathcal{D}_s = \left\{ \left(\boldsymbol{x}_i^s, y_i^s \right) \right\}_{i=1}^{n_s}$, should be large enough to train a deep convolutional neural network from scratch. Ideally, these training images should present diversified low-level characteristics. That is, running a filter bank on them give rise to as diversified responses as possible. There exist a few large-scale visual recognition datasets that can serve as the source domain, including ImageNet ILSVRC dataset [36], Places [50], and MS COCO [23].

Source Domain Training Images : In our selective joint fine-tuning, we do not use all images in the source domain as training images. Instead, for each image from the target domain, we search a certain number of images with similar low-level characteristics from the source domain. Only images returned from these searches are used as training images for the source learning task in selective joint fine-tuning. We apply a filter bank to all images in both source domain and target domain. Histograms of filter bank responses are used as image descriptors during search. We associate an adaptive number of source domain images with each target domain image. Hard training samples in the target domain might be associated with a larger number of source domain images. Two filter banks are used in our experiments. One is the Gabor filter bank, and the other consists of kernels in the convolutional layers of AlexNet pre-trained on ImageNet [22].

CNN Architecture : Almost any existing deep convolutional neural network, such as AlexNet [22], VGGNet [18], and ResidualNet [12], can be used in our selective joint fine-tuning. We use the 152-layer residual network with identity mappings [13] as the CNN architecture in our experiments. The entire residual network is shared by the source and target learning tasks. An extra output layer is added on top of the residual network for each of the two learning tasks. This output layer is not shared because the two learning tasks may not share the same label space. The residual network is pre-trained either on ImageNet or Places.

Source-Target Joint Fine-tuning : Each task uses its own cost function during selective joint fine-tuning, and every training image only contributes to the cost function corresponding to the domain it comes from. The source domain images selected by the aforementioned searches are used as training images for the source learning task only while the entire target domain is used as the training set for the target learning task only. Since the residual network (with all its convolutional layers) is shared by these two learning tasks, it is fine-tuned by both training sets. And the output layers on top of the residual network are fine-tuned by its corresponding training set only. Thus we conduct end-to-end joint fine-tuning to minimize the original loss functions of the source learning task and the target learning task simultaneously.

3.2. Similar Image Search

There is a unique step in our pipeline. For each image from the target domain, we search a certain number of images with similar low-level characteristics from the source domain. Only images returned from these searches are used as training images for the source learning task in selective joint fine-tuning. We elaborate this image search step below.

Filter Bank We use the responses to a filter bank to describe the low-level characteristics of an image. The first filter bank we use is the Gabor filter bank. Gabor filters are commonly used for feature description, especially texture description [28]. Gabor filter responses are powerful low-level features for image and pattern analysis. We use the parameter setting in [28] as a reference. For each of the real and imaginary parts, we use 24 convolutional kernels with 4 scales and 6 orientations. Thus there are 48 Gabor filters in total.

Kernels in a deep convolutional neural network are actually spatial filters. When there is nonlinear activation following a kernel, the combination of the kernel and nonlinear activation is essentially a nonlinear filter. A deep CNN can extract low/middle/high level features at different convolutional layers [48]. Convolutional layers close to the input data focus on extract low-level features while those further away from the input extract middle- and high-level features. In fact, a subset of the kernels in the first convolutional layer of AlexNet trained on ImageNet exhibit oriented stripes, similar to Gabor filters [22]. When trained on a large-scale diverse dataset, such as ImageNet, such kernels can be used for describing generic low-level image characteristics. In practice, we use all kernels (and their following nonlinear activation) from the first and second convolutional layers of AlexNet pre-trained on ImageNet as our second choice of a filter bank.

Image Descriptor Let $\mathbb{C}_i(m, n)$ denote the response map to the i-th convolutional kernel or Gabor filter in our filter bank, and ϕ_i its histogram. To obtain more discriminative histogram features, we first obtain the upper bound h_i^u and lower bound h_i^l of the i-th response map by scanning the entire target domain $\mathcal{D}_t$. Then the interval h_i^l, h_i^u is divided into a set of small bins. We adaptively set the width of every histogram bin so that each of them contains a roughly equal percentage of pixels. In this manner, we can avoid a large percentage of pixels falling into the same bin. We concatenate the histograms of all filter response maps to form a feature vector, $\phi^k = \{\phi_1, \phi_2, , \phi_D\}$, for image x_k.

Nearest Neighbor Ranking Given the histogram-based descriptor of a training image x_i^t in the target domain, we search for its nearest neighbors in the source domain $\mathcal{D}_s$. Note that the number of kernels in different convolutional layers of AlexNet might be different. To ensure equal weighting among different convolutional layers during nearest neighbor search, each histogram of kernel responses is normalized by the total number of kernels in the corresponding layer. Thus the distance between the descriptor of a source image x_j^s and that of a target image x_i^t is computed as follows.

$$\mathcal{H}(x_i^t, x_j^s) = \sum_{h=1}^{D} w_h[\kappa(\phi_h^{i,t}, \phi_h^{j,s}) + \kappa(\phi_h^{j,s}, \phi_h^{i,t})], \quad (1)$$

where $w_h = 1/N_h$, N_h is the number of convolutional kernels in the corresponding layer, $\phi_h^{i,t}$ and $\phi_h^{j,s}$ are the h-th histogram for images x_i^t and x_j^s, and $\kappa(\cdot, \cdot)$ is the KL-divergence.

Hard Samples in the Target Domain   The labels of training samples in the target domain have varying degrees of difficulty to satisfy. Intuitively, we would like to seek extra help for those hard training samples in the target domain by searching for more and more nearest neighbors in the source domain. We propose an iterative scheme for this purpose. We calculate the information entropy to measure the classification uncertainty of training samples in the target domain after the m-th iteration as follows.

$$\mathcal{H}_i^m = -\sum_{c=1}^{C} p_{i,c}^m \log(p_{i,c}^m), \qquad (2)$$

where C is the number of classes, $p_{i,c}^m$ is the probability that the i-th training sample belongs to the c-th class after a softmax layer in the m-th iteration.

Training samples that have high classification uncertainty are considered hard training samples. In the next iteration, we increase the number of nearest neighbors of the hard training samples as in Eq. (3.2), and continue fine-tuning the model trained in the current iteration. For a training sample x_i^t in the target domain, the number of its nearest neighbors in the next iteration is defined as follows.

$$\mathcal{K}_i^{m+1} = \begin{cases} \mathcal{K}_i^m + \sigma_0, & \widehat{y}_i^t \neq y_i^t \\ \mathcal{K}_i^m + \sigma_1, & \widehat{y}_i^t = y_i^t \quad and \quad \mathcal{H}_i^m \geq \delta \\ \mathcal{K}_i^m, & \widehat{y}_i^t = y_i^t \quad and \quad \mathcal{H}_i^m < \delta \end{cases} \quad (3)$$

where σ_0, σ_1 and δ are constants, $\widehat{y}_i^t$ is predicted label of x_i^t, and $\mathcal{K}_i^m$ is the number of nearest neighbors in the m-th iteration. By changing the number of nearest neighbors for samples in the target domain, the subset of the source domain used as training data evolves over iterations, which in turn gradually changes the feature representation learned in the deep network. In the above equation, we typically set $\delta = 0.1$, $\sigma_0 = 4\mathcal{K}_0$ and $\sigma_1 = 2\mathcal{K}_0$, where $\mathcal{K}_0$ is the initial number of nearest neighbors for all samples in the target domain. In our experiments, we stop after five iterations.

In Table 1, we compare the effectiveness of Gabor filters and various combinations of kernels from AlexNet in our selective joint fine-tuning. In this experiment, we use the 50-layer residual network [12] with half the number of kernels in each convolutional layer of the original architecture.

4. Experiments

4.1. Implementation

In all experiments, we use the 152-layer residual network with identity mappings [13] as the deep convolutional architecture, and conventional fine-tuning performed on a pre-trained network with the same architecture without using

Filter Bank	over all Accuracy(%)
Conv1-Conv2 in AlexNet	**89.59**
Conv1-Conv5 in AlexNet	88.82
Conv4-Conv5 in AlexNet	88.48
Gabor Filters	88.90
Fine-tuning w/o source domain	88.12

Table 1. A comparison of classification performance on Oxford Flowers 102 using various choices for the filter bank in selective joint fine-tuning.

any source datasets as our *baseline*. Note that the network architecture we use is different from those used in most published methods for the datasets we run experiments on, and many existing methods adopt sophisticated parts models and feature encodings. The performance of such methods are still included in this paper to indicate that our simple holistic method without incorporating parts models and feature encodings is capable of achieving state-of-the-art performance.

We use the pre-trained model released in [12] to initialize the residual network. During selective joint fine-turning, source and target samples are mixed together in each mini-batch. Once the data has passed the average pooling layer in the residual network, we split the source and target samples, and send them to their corresponding softmax classifier layer respectively. Both the source and target classifiers are initialized randomly.

We run all our experiments on a TITAN X GPU with 12GB memory. All training data is augmented as in [31] first, and we follow the training and testing settings in [12]. Every mini-batch can include 20 224×224 images using a modified implementation of the residual network. We include randomly chosen samples from the target domain in a mini-batch. Then for each of the chosen target sample, we further include one of its retrieved nearest neighbors from the source domain in the same mini-batch. We set the iter size to 10 for each iteration in Caffe [17]. The momentum parameter is set to 0.9 and the weight decay is 0.0001 in SGD. During selective joint fine-tuning, the learning rate starts from 0.01 and is divided by 10 after every $2400-5000$ iterations in all the experiments. Most of the experiments can finish in 16000 iterations.

4.2. Source Image Retrieval

We use the ImageNet ILSVRC 2012 training set [36] as the source domain for Stanford Dogs [19], Oxford Flowers [29], and Caltech 256 [11], and the combination of the ImageNet and Places 205 [50] training sets as the source domain for MIT Indoor 67 [33]. Fig. 2 shows the retrieved 1-st, 10-th, 20-th, 30-th, and 40-th nearest neighbors from ImageNet [36] or Places [50]. It can be observed that corresponding source and target images share similar colors,
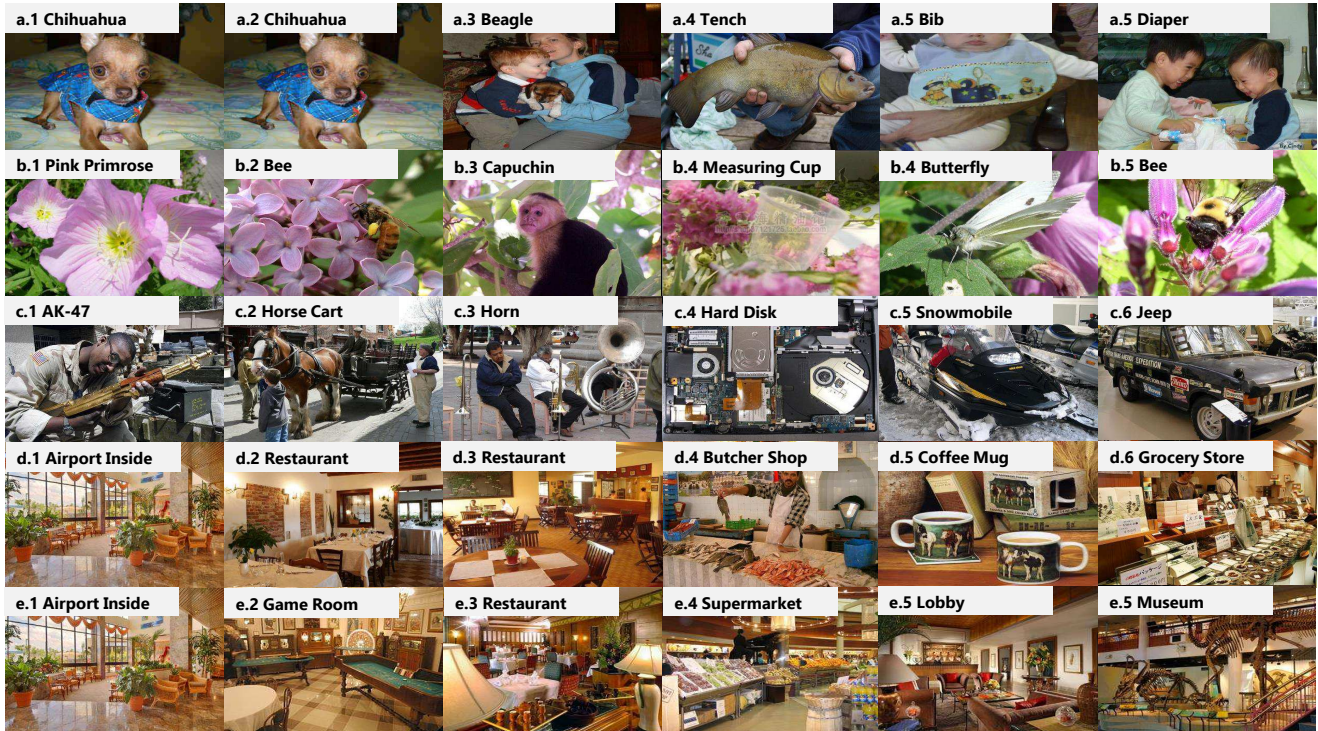
Figure 2. Images in the source domain that have similar low-level characteristics with the target images. The first column shows target images from Stanford Dogs 120 [19], Oxford Flowers 102 [29], Caltech 256 [11], and MIT Indoor 67 [33]. The following columns in rows (a)-(d) are the corresponding 1st, 10-th, 20-th, 30-th and 40-th nearest images in ImageNet (source domain). The following columns in row (e) are images retrieved from Places (source domain for MIT Indoor 67).

local patterns and global structures. Since low-level filter bank responses do not encode strong semantic information, the 50 nearest neighbors from a target domain include images from various and sometimes completely unrelated categories.

We determined experimentally that there should be at least 200,000 retrieved images from the source domain. Too few source images give rise to overfitting quickly. Therefore, the initial number of retrieved nearest neighbors ($\mathcal{K}_0$) for each target training sample is set to meet this requirement. On the other hand, a surprising result is that setting $\mathcal{K}_0$ too large would make the performance of the target learning task drop significantly. In our experiments, we set $\mathcal{K}_0$ to different values for Stanford Dogs ($\mathcal{K}_0 = 100$), Oxford Flowers ($\mathcal{K}_0 = 300$), Caltech 256 ($\mathcal{K}_0 = 50 - 100$), and MIT Indoor 67 ($\mathcal{K}_0 = 100$). Since there exists much overlap among the nearest neighbors of different target samples, the retrieved images typically do not cover the entire ImageNet or Places datasets.

4.3. Fine-grained Object Recognition

Stanford Dogs 120. Stanford Dogs 120 [19] contains 120 categories of dogs. There are 12000 images for training, and 8580 images for testing. We do not use the parts in-

formation during selective joint fine-tuning, and use the commonly used mean class accuracy to evaluate the performance as in [11].

As shown in Table 2, the mean class accuracy achieved by fine-tuning the residual network using the training samples of this dataset only and without a source domain is 80.4%. It shows that the 152-layer residual network [12, 13] pre-trained on the ImageNet dataset [36] has a strong generalization capability on this fine-grained classification task. Using the entire ImageNet dataset during regular joint fine-tuning can improve the performance by 5.1%. When we finally perform our proposed selective joint fine-tuning using a subset of source domain images retrieved using histograms of low-level convolutional features, the performance is further improved to 90.2%, which is 9.8% higher than the performance of conventional fine-tuning without a source domain and 4.3% higher than the result reported in [21], which expands the original target training set using Google image search. This comparison demonstrates that selective joint fine-tuning can significantly outperform conventional fine-tuning.

Oxford Flowers 102. Oxford Flowers 102 [29] consists of 102 flower categories. 1020 images are used for training, 1020 for validation, and 6149 images are used for testing.

Method	mean Acc(%)
HAR-CNN [44]	49.4
Local Alignment [9]	57.0
Multi scale metric learning [32]	70.3
MagNet [35]	75.1
Web Data + Original Data [21]	85.9
Training from scratch using target domain only	53.8
Selective joint training from scratch	83.4
Fine-tuning w/o source domain	80.4
Joint fine-tuning with all source samples	85.6
Selective joint FT with random source samples	85.5
Selective joint FT w/o iterative NN retrieval	88.3
Selective joint FT with Gabor filter bank	87.5
Selective joint fine-tuning	90.2
Selective joint FT with Model Fusion	**90.3**

Table 2. Classification results on Stanford Dogs 120.

There are only 10 training images in each category.

As shown in Table 3, the mean class accuracy achieved by conventional fine-tuning using the training samples of this dataset only and without a source domain is 92.3%. Selective joint fine-tuning further improves the performance to 94.7%, 3.3% higher than previous best result from a single network [35]. To compare with previous state-of-the-art results obtained using an ensemble of different networks, we also average the performance of multiple models obtained during iterative source image retrieval for hard training samples in the target domain. Experiments show that the performance of our ensemble model is 95.8%, 1.3% higher than previous best ensemble performance reported in [20]. Note

Method	mean Acc(%)
MPP [47]	91.3
Multi-model Feature Concat [1]	91.3
MagNet [35]	91.4
VGG-19 + GoogleNet + AlexNet [20]	94.5
Training from scratch using target domain only	58.2
Selective joint training from scratch	80.6
Fine-tuning w/o source domain	92.3
Joint fine-tuning with all source samples	93.4
Selective joint FT with random source samples	93.2
Selective joint FT w/o iterative NN retrieval	94.2
Selective joint FT with Gabor filter bank	93.8
Selective joint fine-tuning	94.7
Selective joint FT with model fusion	**95.8**
VGG-19 + Part Constellation Model [38]	95.3
Selective joint FT with val set	**97.0**

Table 3. Classification results on Oxford Flowers 102. The last two rows compare performance using the validation set as additional training data.

that Simon *et al.* [38] used the validation set in this dataset as additional training data. To verify the effectiveness of our joint fine-tuning strategy, we have also conducted experiments using this training setting and our result from a single network outperforms that of [38] by 1.7%.

4.4. General Object Recognition

Caltech 256. Caltech 256 [11] has 256 object categories and 1 background cluster class. In every category, there are at least 80 images used for training, validation and testing. Researchers typically report results with the number of training samples per class falling between 5 and 60. We follow the testing procedure in [42] to compare with state-of-the-art results.

We conduct four experiments with the number of training samples per class set to 15, 30, 45 and 60, respectively. According to Table 4, in comparison to conventional fine-tuning without using a source domain, selective joint fine-tuning improves classification accuracy in all four experiments, and the degree of improvement varies between 2.6% and 4.1%. Performance improvement due to selective joint fine-tuning is more obvious when a smaller number of target training image per class are used. This is because limited diversity in the target training data imposes a greater need to seek help from the source domain. In most of these experiments, the classification performance of our selective joint fine-tuning is also significantly better than previous state-of-the-art results.

Method	mean Acc(%)
MetaObject-CNN [43]	78.9
MPP + DFSL [47]	80.8
VGG-19 + FV [4]	81.0
VGG-19 + GoogleNet [20]	84.7
Multi scale + multi model ensemble [14]	86.0
Fine-tuning w/o source domain	81.7
Selective joint FT with ImageNet[(i)]	82.8
Selective joint FT with Places[(ii)]	85.8
Selective joint FT with hybrid data[(iii)]	85.5
Average the output of (ii) and (iii)	**86.9**

Table 5. Classification results on MIT Indoor 67.

4.5. Scene Classification

MIT Indoor 67. MIT Indoor 67 [33] has 67 scene categories. In each category, there are 80 images for training and 20 images for testing. Since MIT Indoor 67 is a scene dataset, in addition to the ImageNet ILSVRC 2012 training set [36], the Places-205 training set [50] is also a potential source domain. We compare three settings during selective joint fine-tuning: ImageNet as the source domain, Places as

Method	mean Acc(%) 15/class	mean Acc(%) 30/class	mean Acc(%) 45/class	mean Acc(%) 60/class
M-HMP [2]	40.5±0.4	48.0±0.2	51.9±0.2	55.2±0.3
Z. & F. Net [49]	65.7±0.2	70.6±0.2	72.7±0.4	74.2±0.3
VGG-19 [18]	-	-	-	85.1±0.3
VGG-19 + GoogleNet +AlexNet [20]	-	-	-	86.1
VGG-19 + VGG-16 [18]	-	-	-	86.2±0.3
Fine-tuning w/o source domain	76.4±0.1	81.2±0.2	83.5±0.2	86.4±0.3
Selective joint fine-tuning	**80.5±0.3**	**83.8±0.5**	**87.0±0.1**	**89.1±0.2**

Table 4. Classification results on Caltech 256.

the source domain, and the combination of both ImageNet and Places as the source domain.

As shown in Table 5, the mean class accuracy of selective joint fine-tuning with ImageNet as the source domain is 82.8%, 1.1% higher than that of conventional fine-tuning without using a source domain. Since ImageNet is an object-centric dataset while MIT Indoor 67 is a scene dataset, it is hard for training images in the target domain to retrieve source domain images with similar low-level characteristics. But source images retrieved from ImageNet still prevent the network from overfitting too heavily and help achieve a performance gain. When the Places dataset serves as the source domain, the mean class accuracy reaches 85.8%, which is 4.1% higher than the performance of fine-tuning without a source domain and 4.8% higher than previous best result from a single network [4]. And the hybrid source domain based on both ImageNet and Places does not further improve the performance. Once averaging the output from the networks jointly fine-tuned with Places and the hybrid source domain, we obtain a classification accuracy 0.9% higher than previous best result from an ensemble model [14].

4.6. Ablation Study

We perform an ablation study on both Stanford Dogs 120 [19] and Oxford Flowers 102 [29] by replacing or removing a single component from our pipeline. First, instead of fine-tuning, we perform training from scratch in two settings, one using the target domain only and the other using selective joint training. Tables 2 and 3 show that while selective joint training obviously improves the performance, it is still inferior than fine-tuning pre-trained networks. This is because we only subsample a relatively small percentage (20-30%) of the source data, which is still insufficient to train deep networks from scratch. Second, instead of using a subset of retrieved training images from the source domain, we simply use all training images in the source domain. Joint fine-tuning with the entire source domain decrease the performance by 4.6% and 1.3% respectively. This demonstrates that using more training data from the source domain is not always better. On the contrary, using less but

more relevant data from the source domain is actually more helpful. Third, instead of using a subset of retrieved training images, we use the same number of randomly chosen training images from the source domain. Again, the performance drops by 4.7% and 1.5% respectively. Fourth, to validate the effectiveness of iteratively increasing the number of retrieved images for hard training samples in the target domain, we turn off this feature and only use the same number ($\mathcal{K}_0$) of retrieved images for all training samples in the target domain. The performance drops by 1.9% and 0.5% respectively. This indicates that our adaptive scheme for hard samples is useful in improving the performance. Fifth, we use convolutional kernels in the two bottom layers of a pre-trained AlexNet as our filter bank. If we replace this filter bank with the Gabor filter bank, the overall performance drops by 2.7% and 0.9% respectively, which indicates a filter bank learned from a diverse dataset could be more powerful than an analytically defined one. Finally, if we perform conventional fine-tuning without using a source domain, the performance drop becomes quite significant and reaches 9.8% and 2.4% respectively.

5. Conclusions

In this paper, we address deep learning tasks with insufficient training data by introducing a new deep transfer learning scheme called selective joint fine-tuning, which performs a target learning task with insufficient training data simultaneously with another source learning task with abundant training data. Different from previous work which directly adds extra training data to the target learning task, our scheme borrows samples from a large-scale labeled dataset for the source learning task, and do not require additional labeling effort beyond the existing datasets. Experiments show that our deep transfer learning scheme achieves state-of-the-art performance on multiple visual classification tasks with insufficient training data for deep networks. Nevertheless, how to find the most suitable source domain for a specific target learning task remains an open problem for future investigation.

References

[1] H. Azizpour, A. Sharif Razavian, J. Sullivan, A. Maki, and S. Carlsson. From generic to specific deep representations for visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 36–45, 2015. 3, 7

[2] L. Bo, X. Ren, and D. Fox. Multipath sparse coding using hierarchical matching pursuit. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 660–667, 2013. 8

[3] R. Caruana. Multitask learning. In *Learning to learn*, pages 95–133. Springer, 1998. 2

[4] M. Cimpoi, S. Maji, I. Kokkinos, and A. Vedaldi. Deep filter banks for texture recognition, description, and segmentation. *International Journal of Computer Vision*, 118(1):65–94, 2016. 7, 8

[5] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655, 2014. 2

[6] L. Duan, D. Xu, I. W.-H. Tsang, and J. Luo. Visual event recognition in videos by learning from web data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1667–1680, 2012. 3

[7] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015. 2

[8] A. Evgeniou and M. Pontil. Multi-task feature learning. *Advances in neural information processing systems*, 19:41, 2007. 2

[9] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders, and T. Tuytelaars. Local alignments for fine-grained categorization. *International Journal of Computer Vision*, 111(2):191–212, 2015. 7

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1, 3

[11] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007. 2, 5, 6, 7

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 1, 4, 5, 6

[13] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. *arXiv preprint arXiv:1603.05027*, 2016. 4, 5, 6

[14] L. Herranz, S. Jiang, and X. Li. Scene recognition with cnns: objects, scales and dataset bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 571–579, 2016. 7, 8

[15] S. Hong, J. Oh, B. Han, and H. Lee. Learning transferrable knowledge for semantic segmentation with deep convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3

[16] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601–608, 2006. 3

[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014. 5

[18] S. Karen and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2015. 1, 4, 8

[19] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, volume 2, 2011. 1, 2, 5, 6, 8

[20] Y.-D. Kim, T. Jang, B. Han, and S. Choi. Learning to select pre-trained deep representations with bayesian evidence framework. *arXiv preprint arXiv:1506.02565*, 2015. 7, 8

[21] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, and L. Fei-Fei. The unreasonable effectiveness of noisy data for fine-grained recognition. In *European Conference on Computer Vision*, 2015. 3, 6, 7

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 2012. 1, 4

[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 1, 3

[24] 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

[25] 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

[26] M. Long and J. Wang. Learning transferable features with deep adaptation networks. *CoRR, abs/1502.02791*, 1:2, 2015. 3

[27] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *2015 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 5188–5196. IEEE, 2015. 3

[28] B. S. Manjunath and W.-Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on pattern analysis and machine intelligence*, 18(8):837–842, 1996. 2, 4

[29] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, pages 722–729. IEEE, 2008. 1, 2, 5, 6, 8

[30] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010. 1, 3

[31] M. Paulin, J. Revaud, Z. Harchaoui, F. Perronnin, and C. Schmid. Transformation pursuit for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3646–3653, 2014. 5

[32] Q. Qian, R. Jin, S. Zhu, and Y. Lin. Fine-grained visual categorization via multi-stage metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3716–3724, 2015. 7

[33] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 413–420. IEEE, 2009. 2, 5, 6, 7

[34] 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. 2

[35] O. Rippel, M. Paluri, P. Dollar, and L. Bourdev. Metric learning with adaptive density discrimination. *stat*, 1050:2, 2016. 7

[36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1, 2, 3, 5, 6, 7

[37] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014. 2

[38] M. Simon and E. Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1143–1151, 2015. 7

[39] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 1

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1

[41] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, 2015. 3

[42] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367. IEEE, 2010. 7

[43] R. Wu, B. Wang, W. Wang, and Y. Yu. Harvesting discriminative meta objects with deep cnn features for scene classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1287–1295, 2015. 7

[44] S. Xie, T. Yang, X. Wang, and Y. Lin. Hyper-class augmented and regularized deep learning for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2645–2654, 2015. 7

[45] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8(Jan):35–63, 2007. 1

[46] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, and Y. Yu. Hd-cnn: Hierarchical deep convolutional neural networks for large scale visual recognition. In *International Conference on Computer Vision (ICCV)*, 2015. 1

[47] D. Yoo, S. Park, J.-Y. Lee, and I. So Kweon. Multi-scale pyramid pooling for deep convolutional representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 71–80, 2015. 7

[48] 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, 4

[49] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014. 8

[50] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014. 1, 2, 3, 5, 7