Impact of ImageNet Model Selection on Domain Adaptation

Youshan Zhang   Brian D. Davison
Computer Science and Engineering, Lehigh University, Bethlehem, PA, USA
{yoz217, bdd3}@lehigh.edu

Abstract

Deep neural networks are widely used in image classification problems. However, little work addresses how features from different deep neural networks affect the domain adaptation problem. Existing methods often extract deep features from one ImageNet model, without exploring other neural networks. In this paper, we investigate how different ImageNet models affect transfer accuracy on domain adaptation problems. We extract features from sixteen distinct pre-trained ImageNet models and examine the performance of twelve benchmarking methods when using the features. Extensive experimental results show that a higher accuracy ImageNet model produces better features, and leads to higher accuracy on domain adaptation problems (with a correlation coefficient of up to 0.95). We also examine the architecture of each neural network to find the best layer for feature extraction. Together, performance from our features exceeds that of the state-of-the-art in three benchmark datasets.

1. Introduction

In recent years, we have witnessed the great success of deep neural networks in some standard benchmarks such as ImageNet [5] and CIFAR-10 [20]. However, in the real world, we often have a serious problem that lacks labeled data for training. It is known that training and updating of the machine learning model depends on data annotation. Although we can get a large amount of data, few data are correctly labeled. Data annotation is a time-consuming and expensive operation. This brings challenges to properly train and update machine learning models. As a result, some application areas have not been well developed due to insufficient labeled data for training. Therefore, it is necessary to reuse existing labeled data and models for labeling new data. However, we often encounter the problem of domain shift if we train on one dataset and test on another.

Existing work only addresses how ImageNet models affect the general classification problem [19]. However, no work considers how different ImageNet models affect domain adaptation.

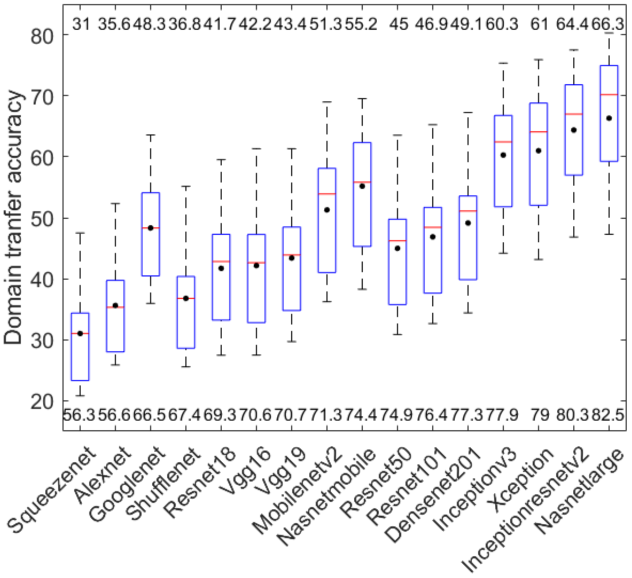


Figure 1: Boxplots of domain transfer accuracy across twelve methods on each of sixteen neural networks using the Office-Home dataset (bottom is the accuracy of the ImageNet models; top is the mean domain transfer accuracy across twelve methods; black dots are mean values and red line is the median value).

In this paper, we report the effect of different ImageNet models on three different domain adaption datasets using each of twelve methods. We want to find how the features from these deep neural networks affect the final domain transfer accuracy. Specifically, we conduct a large-scale study of transfer learning across 16 modern convolutional neural networks for image classification on office + caltech 10, office 31 and office-home image classification datasets using two basic classifiers and ten domain adaptation methods using extracted features. Fig. 1 presents boxplots of the performance of twelve methods across sixteen different neural networks using the Office-Home dataset.

This paper provides two specific contributions:

1. We are the first to examine how different ImageNet models affect domain transfer accuracy, using fea-

tures from sixteen distinct pre-trained neural networks on twelve methods across three benchmark datasets. The correlation of domain adaptation performance and ImageNet classification performance is high, ranging from 0.71 to 0.95, suggesting that features from a higher-performing ImageNet-trained model are more valuable than those from a lower-performing model.

2. We also find that all three benchmark datasets suggest that the layer prior to the last fully connected layer is the best source.

2. Background

2.1. Related work

Domain adaptation has emerged as a prominent method to solve the domain shift problem. There have been efforts for both traditional [11, 18, 48, 54] and deep learning-based [45, 43, 23, 8] methods in domain adaptation.

Traditional methods highly depend on the extracted features from raw images. Before the emergence of deep neural networks, lower-level SURF features have been widely used in domain adaptation [11]. However, with the development of deep neural networks, extracted features from pre-trained neural networks lead to higher performance than the use of lower-level features (Alexnet [21], Decaf [48], Resnet50 [29], Xception [54], etc.). Distribution alignment, feature selection, and subspace learning are three frequently used methods in traditional domain adaptation. There are also many methods that address the different kinds of distribution alignment, from marginal distribution alignment [32, 6, 27, 18], to conditional distribution alignment [12, 47] and finally joint alignment of these two distributions [46, 48]. Feature selection methods aim to find the shared features between source and target domain [1, 27]. Subspace learning includes transfer component analysis [32] in Euclidean space, and the Riemannian subspace space includes sampling geodesic flow [14], geodesic flow kernel (GKF) [11], and geodesic sampling on manifolds (GSM) [54]. However, the predicted accuracy of traditional methods is affected by the extracted features from deep neural networks. It is believed that in general, a better ImageNet model will produce better features than a lower accuracy model on ImageNet [19]. However, there is no such work to validate this hypothesis in domain adaptation.

Recently, deep learning models have been treated as a better mechanism for feature representation in domain adaptation. There are four major types of methods in deep domain adaptation: discrepancy-based methods, adversarial discriminative models, adversarial generative models, and data reconstruction-based models. Among all of these, maximum mean discrepancy (MMD) is one of the most efficient ways to minimize the discrepancy between source and target domain [43, 23, 9]. Adversarial discriminative based models aim to define a domain confusion objective to identify the domains via a domain discriminator. The Domain-Adversarial Neural Network (DANN) considers a minimax loss to integrate a gradient reversal layer to promote the discrimination of source and target domain [8]. The Adversarial Discriminative Domain Adaptation (ADDA) method uses an inverted label GAN loss to split the source and target domain, and features can be learned separately [42]. The adversarial generative models combine the discriminative model with generative components based on Generative Adversarial Networks (GANs) [13]. Coupled Generative Adversarial Networks [22] consist of a series of GANs, and each of them can represent one of the domains. Data reconstruction-based methods jointly learn source label predictions and unsupervised target data reconstruction [2].

Both traditional and deep learning-based methods more or less rely on the extracted feature from deep neural networks. However, with so many different deep neural networks, we do not know which one is the best. Therefore, it is necessary to explore how different pre-trained models affect domain transfer accuracy. We focus exclusively on models trained on ImageNet because it is a large benchmark dataset and pre-trained models are widely available.

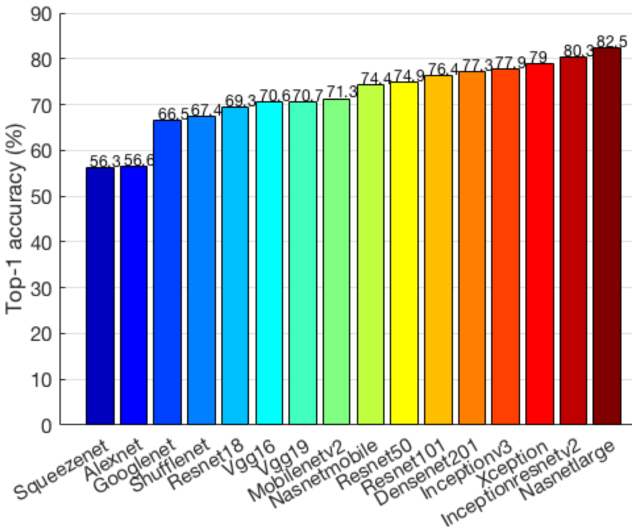


Figure 2: Top-1 accuracy of the sixteen neural networks on the ImageNet task.

2.2. ImageNet models

There are many deep neural networks well-trained on ImageNet with differing accuracy. Kornblith et al. [19] explored the effects of sixteen variations of five models (Inception, Resnet, Densenet, Mobilenet, Nasnet) trained on ImageNet on general transfer learning (there is no domain shift in datasets). In contrast, here we explore sixteen different neural network architectures from light-weight but low performing networks to expensive, but high performing networks that have been proposed in the last decade. Specifically, these sixteen neural networks are Squeezenet [17], Alexnet [21], Googlenet [40], Shufflenet [51], Resnet18 [15], Vgg16 [36], Vgg19 [36], Mobilenetv2 [35], Nasnetmobile [55], Resnet50 [15], Resnet101 [15], Densenet201 [16], Inceptionv3 [41], Xception [4], Inceptionresnetv2 [39], Nasnetlarge [55].

Fig. 2 shows the top-1 classification accuracy of the sixteen neural networks on ImageNet task. Performance ranges from 56.3% (Squeezenet) to 82.5% (Nasnetlarge). In this paper, we examine the domain adaptation feature sources according to their ImageNet accuracy. Fig. 3 shows top-1 accuracy and number of parameters and network size in each network; the gray circles show the size of memory (megabyte) and other colors represent the different models.

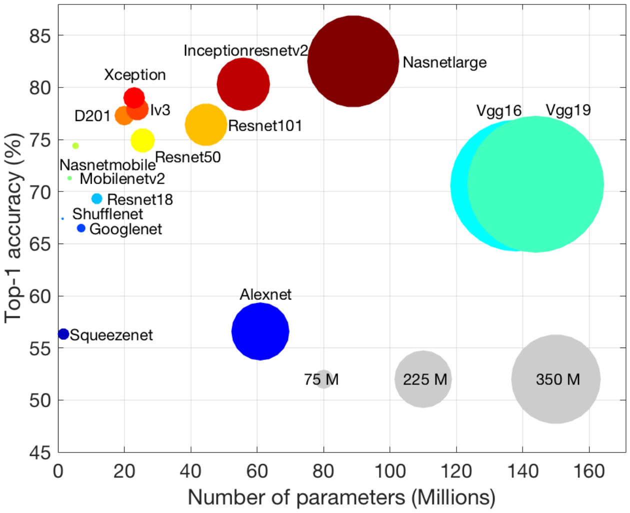


Figure 3: Top-1 accuracy versus network size and parameters. (D201: Densenet201; Iv3: Inceptionv3)

2.3. Domain transfer problem

Most previous domain adaptation methods focus only on extracted features from one neural network without exploring other networks. Resnet50 is one of the most frequently used models in this field. Kornblith et al. [19] pointed out that Resnet is the best source for extracting features for transfer learning. However, there is no reason to suggest this source is the best for domain adaptation, but many more methods are developed based on this source without justification. In our experience, extracting features from a better ImageNet model will lead to better domain adaptation performance [54]. However, it is unclear whether this conclusion applies across the domain adaptation field. To address this question, we perform extensive experiments to show the effects of features from different ImageNet models on domain transfer accuracy for domain adaptation.

3. Methods

3.1. Problem and notation

For unsupervised domain adaptation, given the source domain $\mathcal{D}_\mathcal{S}$ data: $\mathcal{X}_\mathcal{S}$ with its labels $\mathcal{Y}_\mathcal{S}$ in C categories and target domain $\mathcal{D}_\mathcal{T}$ data $\mathcal{X}_\mathcal{T}$ without any labels ($\mathcal{Y}_\mathcal{T}$ for evaluation only). Our ultimate goal is to predict the label in the target domain with a high accuracy using the trained model from source domain.

3.2. Feature extraction

We extract the features from raw images using the above sixteen pre-trained neural networks. To get consistent numbers of features, we extract the features from the activations of the last fully connected layer [52, 53]; thus the final output of one image becomes one vector 1×1000. Feature extraction is implemented via two steps: (1) rescale the image into different input sizes of pre-trained neural networks; (2) extract the feature from the last fully connected layer. Fig. 4 presents the t-SNE view of extracted features from sixteen neural networks of the Amazon domain in Office31 dataset.

3.3. Different classifiers

To avoid the bias that might come from a particular classification or domain adaptation method, we evaluate the performance of extracted features across twelve methods.

1. **Support vector machines (SVM)** and **1-nearest neighbor (1NN)** are two basic classifiers that do not perform domain adaptation. Thus, these classifiers reveal the fundamental accuracy from raw features, and serve as baselines for comparison with domain adaptation methods.
2. **Geodesic Flow Kernel (GFK)** [11], which learns the "geodesic" features from a manifold.
3. **Geodesic sampling on manifolds (GSM)** [54], which performs generalized subspace learning on manifolds.
4. **Balanced distribution adaptation (BDA)** [46], which tackles the imbalanced data and leverages marginal and conditional distribution discrepancies.
5. **Joint distribution alignment (JDA)** [26], which changes both marginal and conditional distribution.
6. **CORrelation Alignment (CORAL)** [54], which performs second-order subspace alignment.
7. **Transfer Joint Matching (TJM)** [27], which changes the marginal distribution by source sample selection.
8. **Joint Geometrical and Statistical Alignment (JGSA)** [49], which aligns marginal and conditional distributions with label propagation.
9. **Adaptation Regularization (ARTL)** [25] learns an adaptive classifier via optimizing the structural risk function and the distribution matching between domains, and the manifold marginal distribution.

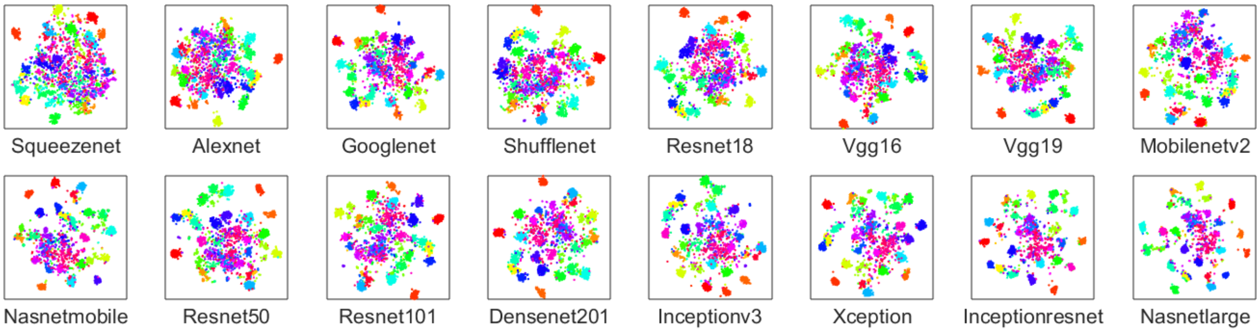


Figure 4: t-SNE view of extracted features from the last fully connected layer of sixteen neural networks. Different colors represent different classes. The more separation of the classes in the dataset, the better the features are (Amazon domain in the Office31 dataset).

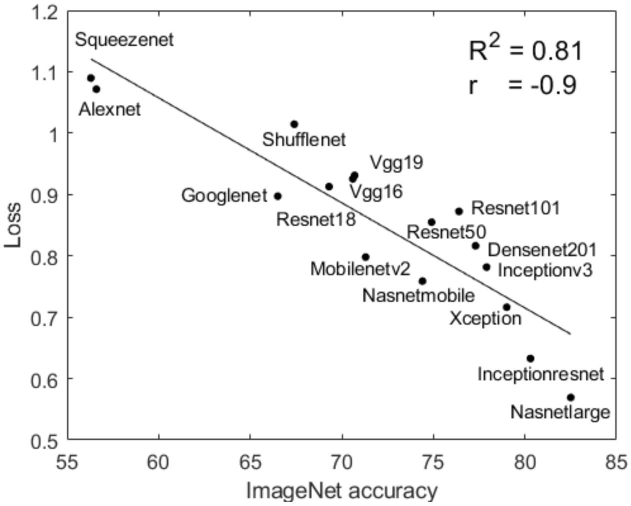


Figure 5: t-SNE loss of sixteen neural networks on domain Amazon in the office31 dataset. With increase of ImageNet accuracy, the loss is reduced, representing better features.

10. **Manifold Embedded Distribution Alignment (MEDA)** [48] addresses degenerated features transformation and unevaluated distribution alignment.
11. **Modified Distribution Alignment (MDA)** [53] is based on the MEDA model, but it removed the GFK model and replaced it with well-represented features.

3.4. Statistical methods

Analysis of the domain transfer accuracy and ImageNet accuracy requires consideration of the relationship between them. We hence report the correlation coefficient score and coefficient of determination scores.

Correlation coefficient. We examine the strength of the correlation between ImageNet accuracy and the accuracy of the domain adaptation accuracy.

$$r(A,B) = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n A_{mn} - \bar{A})^2 (\sum_m \sum_n B_{mn} - \bar{B})^2}}, \tag{1}$$

where $\bar{A}$ and $\bar{B}$ is average of vector elements. The range of the correlation is from -1 (strong negative) to 1 (strong positive), while 0 indicates there is no correlation between sub-source data and sub-target data.

Coefficient of determination $\mathbf{R^2}$. The R^2 statistic has proven to be a useful metric to indicate the significance of linear regression models [7]. The range of the R^2 statistic is between $[0, 1]$; the higher the R^2 value, the more variation is explained by the model, and the better the model fits data.

$$R^2 = 1 - \frac{\text{Unexplained variation}}{\text{Total variation}} = 1 - \frac{\sum_{i=1}^{N} S_{residual}}{\sum_{i=1}^{N} S_{total}}, \tag{2}$$

where $S_{residual} = (y_i - y_i')^2$, $S_{total} = (y_i - \bar{y})^2$, N is the number of samples, y_i' is the estimate from the regression model, y_i is the actual value, $\bar{y}$ is the mean value of y.

By evaluating methods using the above metrics, we can determine the relationship between features from ImageNet models and domain transfer accuracy.

4. Results

4.1. Datasets

We calculate the accuracy of each method in the image recognition problem of three datasets[1].

Office + Caltech 10 [11] is a standard benchmark for domain adaptation, which consists of Office 10 and Caltech 10 datasets. It contains 2,533 images in four domains: Amazon (A), Webcam (W), DSLR (D) and Caltech (C). Amazon images are mostly from online merchants, DSLR and Webcam images are mostly from offices. In the experiments, C

[1] Source code is available at https://github.com/heaventian93/ImageNet-Models-on-Domain-Adaptation.

→ A means learning knowledge from domain C and applied to domain A. We evaluate all methods and networks across twelve transfer tasks.

Office-31 [34] is another benchmark dataset for domain adaptation, and it consists of 4,110 images in 31 classes from three domains: Amazon (A), which contains images from amazon.com, Webcam (W), and DSLR (D), both containing images that are taken by a web camera or a digital SLR camera with different settings, respectively. We evaluate all methods on all six transfer tasks A→W, D→W, W→D, A→D, D→A, and W→A.

Office-Home [44] contains 15,588 images from four domains, and it has 65 categories. Specifically, Art (Ar) denotes artistic depictions for object images, Clipart (Cl) describes picture collection of clipart, Product (Pr) shows object images with a clear background and is similar to Amazon category in Office-31, and Real-World (Rw) represents object images collected with a regular camera. We also have twelve tasks in this dataset.

To get a consistent number of features, we extract features from the last fully connected layer. Fig. 4 represents the t-SNE view of extracted features from the sixteen neural networks. With the increasing of ImageNet classification accuracy, the separation of features is also improved, which indicates the features are better (from the mixed colors of Squeezenet to the more clearly separated Nasnetlarge model). Also, t-SNE projection loss illustrates the goodness of features. The loss function of the t-SNE method is Kullback-Leibler divergence, which measures the difference between similarities of points in the original space and those in the projected distribution [31]. Thus if the features are well separated in the original higher dimensional space, then in a successful mapping to a low-dimensional space they will also be well separated; the more similar the two distributions, the lower the loss. Typically, lower losses correspond to better features. As shown in Fig. 5, the correlation between the loss and different neural networks is -0.9, and R^2 is .81, which suggests a strong relationship.

Table 1 shows the size and number of parameters of different neural networks and feature extraction time[2]. We find an interesting phenomenon that some networks with larger size and more parameters use less time (e.g., Resnet101), which implies that the size of neural networks are not the only factor affecting feature extraction time. The correlation between extraction time and the network size and the number of parameters are 0.38 and 0.35, respectively, which further reflects the limits of the effects of network size and number of parameters on extraction time.

Table 1: Feature extraction time (Seconds), number of parameters (Millions), and network size (Megabytes) for each source on Office + Caltech-10 datasets.

Task	Net size	Parameters	Time
Squeezenet [17]	46	1.24	13.3
Alexnet [21]	227	61	13.9
Googlenet [40]	27	7	15.9
Shufflenet [51]	6.3	1.4	17.0
Resnet18 [15]	44	11.7	14.8
Vgg16 [36]	515	138	33.6
Vgg19 [36]	535	144	37.1
Mobilenetv2 [35]	13	3.5	21.4
Nasnetmobile [55]	20	5.3	39.3
Resnet50 [15]	96	25.6	22.7
Resnet101 [15]	167	44.6	26.7
Densenet201 [16]	77	20	61.8
Inceptionv3 [41]	89	23.9	28.2
Xception [4]	85	22.9	48.1
Inceptionresnetv2 [39]	209	55.9	54.1
Nasnetlarge [55]	360	88.9	141.2

4.2. ImageNet and domain transfer accuracy

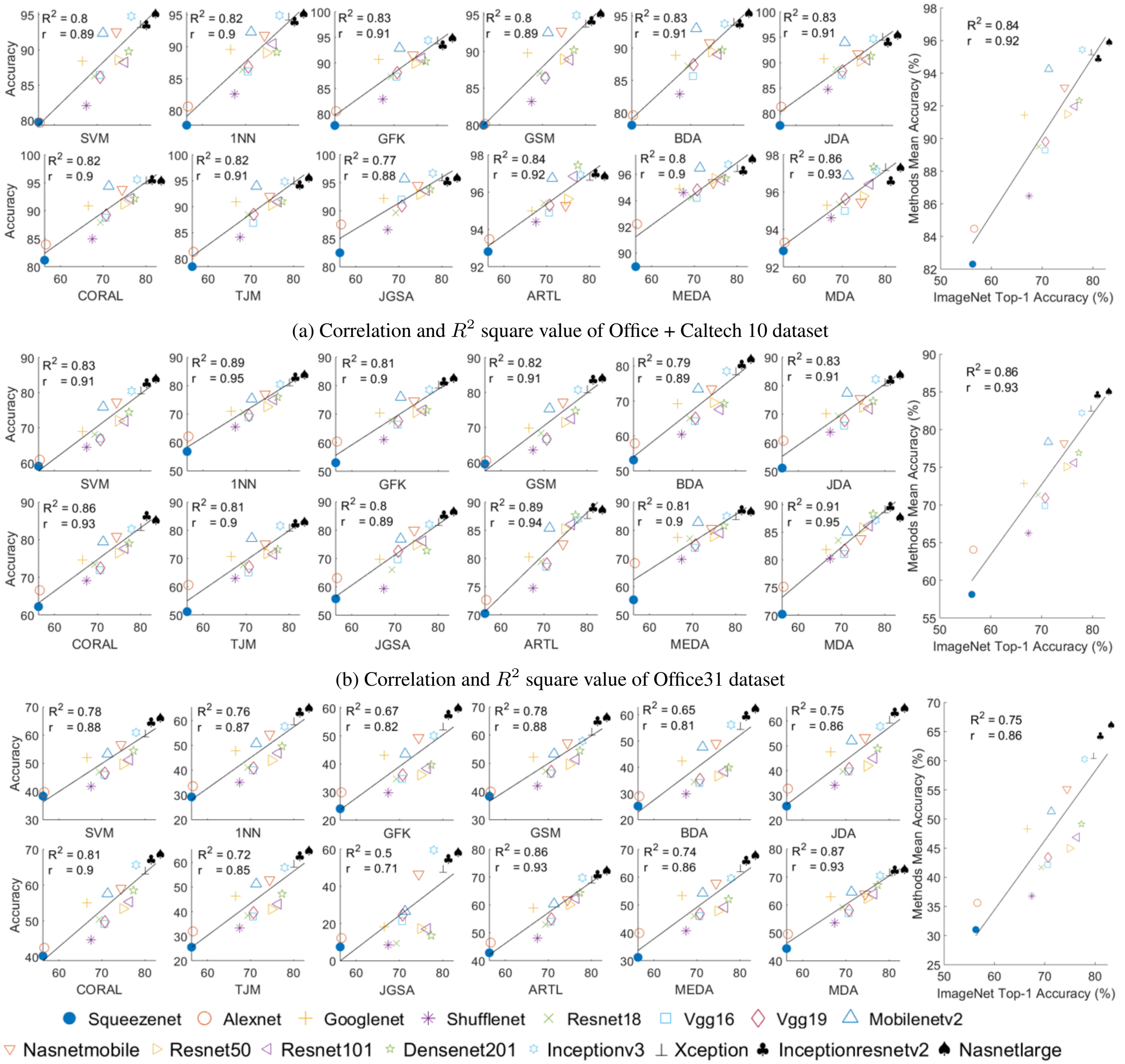
In this setting, different neural networks are only used to extract features; we do not re-train the neural network since we want to explore purely how the different deep ImageNet models affect domain transfer accuracy.

We examine the sixteen deep neural networks in ImageNet, and top-1 target domain accuracy ranges from 56.3% to 82.5%. We measure the trend of domain adaptation performance across three datasets and twelve methods using correlation and R^2 statistics. Fig. 6 presents the correlations and R^2 statistics between top-1 accuracy on ImageNet and the performance of the domain adaptation accuracy. We can make several observations: first of all, the overall domain transfer performance from three datasets is linearly correlated with the increase of the ImageNet model performance.

Among the three datasets, office-home is most challenging because the average performance of all twelve methods are lower than 70% and there is more domain shift in this dataset; the overall accuracy of the other two datasets are higher than 85%, and this leads to correlation score and R^2 value of the office-home dataset to be lower than the other two datasets. Secondly, in the office + caltech 10 dataset, the result from each method presents a similar trend such that with the increasing of ImageNet accuracy, the transfer accuracy is also improved. Notably, we get a different conclusion from previous work [19], which stated that Resnet and Densenet usually give the highest performance.

Thirdly, we see that Nasnetlarge currently has the highest top-1 accuracy in ImageNet. We therefore expect that the features from Nasnetlarge would have the highest performance across three datasets, and it is true that most methods follow this observation. However, we notice that the JGSA model in Office + caltech 10 and MDA model has a lower accuracy than the Inceptionresnetv2 model, which is caused by an error in the model (invalid update of the conditional

[2] Features are extracted using a Geforce 1080 Ti.



(a) Correlation and R^2 square value of Office + Caltech 10 dataset

(b) Correlation and R^2 square value of Office31 dataset

(c) Correlation and R^2 square value of Office-Home dataset

Figure 6: The relationship between ImageNet models and three bookmarking domain adaptation datasets with 12 methods. In each subfigure, the left is the relationship between ImageNet and the domain transfer accuracy across twelve methods, and the right is the average performance of twelve methods.

and marginal distributions). Fourth, the ARTL model has a strange relative performance in the Office-Home dataset; the transfer accuracy from Squeezenet to the Densenet201 is significantly lower than Inceptionv3, Xception, Inceptionresnetv2, and Nasnetlarge. The reason is that the JGSA model does not perform well if there is a significant difference between the source and target domains.

4.3. Comparison with state-of-the-art results

Due to space limitations, we only list the highest accuracy across three datasets using the twelve representative methods along with a few other state-of-the-art methods in Tables 2-4 (Office + Caltech-10 and Office-Home use features from Nasnetlarge and Office 31 uses features from Inceptionresnetv2). The overall performance across all twelve methods is higher than state-of-the-art methods,

Table 2: Accuracy (%) on Office + Caltech-10 datasets

Task	C → A	C → W	C → D	A → C	A → W	A → D	W → C	W → A	W → D	D → C	D → A	D → W	**Average**
SVM	94.7	97.3	99.4	93.3	90.5	92.4	93.9	95.4	**100**	94.2	94.4	99.0	95.4
1NN	95.7	96.3	95.5	93.6	91.5	95.5	93.7	95.7	**100**	93.5	94.8	98.3	95.3
GFK [11]	94.8	96.6	94.9	92.4	92.5	94.9	93.6	95.2	**100**	94.2	94.4	98.3	95.2
GSM [54]	95.6	96.3	98.1	93.9	90.2	93.0	93.9	95.5	**100**	94.4	94.4	99.0	95.4
BDA [46]	95.7	95.6	96.8	92.8	96.6	94.9	93.5	95.8	**100**	93.3	95.8	96.3	95.6
JDA [26]	95.3	96.3	96.8	93.9	95.9	95.5	93.5	95.7	**100**	93.3	95.5	96.9	95.7
CORAL [37]	95.6	96.3	98.1	95.2	89.8	94.3	93.9	95.7	**100**	94.0	96.2	98.6	95.6
TJM [27]	95.7	96.6	95.5	93.2	95.9	97.5	93.4	95.7	**100**	93.5	95.6	96.9	95.8
JGSA [49]	95.2	97.6	96.8	95.2	93.2	95.5	94.6	95.2	**100**	94.9	96.1	99.3	96.1
ARTL [25]	95.7	97.6	97.5	94.6	98.6	**100**	94.6	96.1	**100**	93.5	95.8	99.3	96.9
MEDA [48]	**96.0**	**99.3**	98.1	94.2	99.0	100	94.6	96.5	**100**	94.1	96.1	99.3	97.3
MDA [53]	**96.0**	**99.3**	**99.4**	**94.2**	99.0	**100**	**94.6**	**96.5**	**100**	**94.2**	**96.1**	99.3	**97.4**
DAN [23]	92.0	90.6	89.3	84.1	91.8	91.7	81.2	92.1	**100**	80.3	90.0	98.5	90.1
DDC [43]	91.9	85.4	88.8	85.0	86.1	89.0	78.0	83.8	**100**	79.0	87.1	97.7	86.1
DCORAL [38]	89.8	97.3	91.0	91.9	**100**	90.5	83.7	81.5	90.1	88.6	80.1	92.3	89.7
RTN [28]	93.7	96.9	94.2	88.1	95.2	95.5	86.6	92.5	**100**	84.6	93.8	99.2	93.4
MDDA [33]	93.6	95.2	93.4	89.1	95.7	96.6	86.5	94.8	**100**	84.7	94.7	**99.4**	93.6

Table 3: Accuracy (%) on Office-Home datasets

Task	Ar → Cl	Ar → Pr	Ar → Rw	Cl → Ar	Cl → Pr	Cl → Rw	Pr → Ar	Pr → Cl	Pr → Rw	Rw → Ar	Rw → Cl	Rw → Pr	**Average**
SVM	47.8	76.1	79.2	61.7	70.2	69.5	64.4	48.7	79.5	70.6	49.1	82.1	66.6
1NN	46.4	71.7	77	63.9	69.6	70.4	65.5	46.8	76.0	71.4	48.5	78.7	65.5
GFK [11]	39.6	66.0	72.5	55.7	66.4	64.0	58.4	42.5	73.3	66.0	44.1	76.1	60.4
GSM [54]	47.6	76.4	79.5	62.2	69.7	69.2	65.1	49.5	79.8	71.0	49.6	82.1	66.8
BDA [46]	43.3	69.8	74.1	58.7	66.3	67.7	60.6	46.3	75.3	67.3	48.7	77.0	62.9
JDA [26]	47.4	72.8	76.1	60.7	68.6	70.5	66.0	49.1	76.4	69.6	52.5	79.7	65.8
CORAL [37]	48.0	78.7	80.9	65.7	74.7	75.5	68.4	49.8	80.7	73.0	50.1	82.4	69.0
TJM [27]	47.6	72.3	76.1	60.7	68.6	71.1	64.0	49.0	75.9	68.6	51.2	79.2	65.4
JGSA [49]	42.9	69.5	71.2	50.1	63.0	63.3	55.6	42.6	71.8	60.8	42.1	74.6	59.0
ARTL [25]	53.5	80.2	81.6	71.5	79.9	78.3	**73.1**	56.1	**82.9**	**75.9**	**57.1**	83.7	72.8
MEDA [48]	48.5	74.5	78.8	64.8	76.1	75.2	67.4	49.1	79.7	72.2	51.7	81.5	68.3
MDA [53]	**54.8**	**81.2**	**82.3**	**71.9**	**82.9**	**81.4**	71.1	**53.8**	82.8	75.5	55.3	**86.2**	**73.3**
DCORAL [38]	32.2	40.5	54.5	31.5	45.8	47.3	30.0	32.3	55.3	44.7	42.8	59.4	42.8
RTN [28]	31.3	40.2	54.6	32.5	46.6	48.3	28.2	32.9	56.4	45.5	44.8	61.3	43.5
DAH [44]	31.6	40.8	51.7	34.7	51.9	52.8	29.9	39.6	60.7	45.0	45.1	62.5	45.5
MDDA [33]	35.2	44.4	57.2	36.8	52.5	53.7	34.8	37.2	62.2	50.0	46.3	66.1	48.0
DAN [23]	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN [10]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN [29]	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN-RM [24]	49.2	64.8	72.9	53.8	62.4	62.9	49.8	48.8	71.5	65.8	56.4	79.2	61.5
CDAN-M [24]	50.6	65.9	73.4	55.7	62.7	64.2	51.8	49.1	74.5	68.2	56.9	80.7	62.8

which demonstrates the superiority of the extracted features. However, the classification results of twelve methods are compromised in some tasks (e.g., W → A and D → A in Office 31 datasets), which is likely caused by the differences in tasks, and we cannot guarantee top features are best in all tasks but overall performance is significantly better.

4.4. Which is the best layer for feature extraction?

We extract the features from the last fully connected layer which corresponds to a feature size of 1×1000. However, we do not know which layer is the best one for feature extraction in the domain adaptation problem. In this section, we give an experimental suggestion to choose the best layer for feature extraction. In Tab. 5, we show the results of the last four layers (as other layers often have an extremely large number of features). The output and softmax layers have the same accuracy since the output just changes the probability of the softmax layer to a real class. In addition, we find that the last fully connected layer (LFC) is not the best layer for feature extraction. Instead, the layer prior to the last fully connected layer (P_LFC) has the highest performance. The average improvement of P_LFC layer over the LFC layer across sixteen neural networks for each of the datasets are 0.2%,1.1%, and 1.5%, respectively.

4.5. How to choose the neural network to improve the domain transfer accuracy?

Based on the above results, we suggest extracting features from the layer which is right before the last fully connected layer. The features in this layer are not only well represented but also use less memory. Moreover, although the Nasnetlarge feature has higher accuracy among most

Table 4: Accuracy (%) on Office 31 datasets

Task	A → W	A → D	W → A	W → D	D → A	D → W	**Average**
SVM	81.5	80.9	73.4	96.6	70.6	95.1	83.0
1NN	80.3	81.1	71.8	99.0	71.3	96.4	83.3
GFK [11]	78.1	78.5	71.7	98.0	68.9	95.2	81.7
GSM [54]	84.8	82.7	73.5	96.6	70.9	95.0	83.9
BDA [46]	77.0	79.3	70.3	97.0	68.0	93.2	80.8
JDA [26]	79.1	79.7	72.9	97.4	71.0	94.2	82.4
CORAL [37]	88.9	87.6	74.7	99.2	73.0	96.7	86.7
TJM [27]	79.1	81.1	72.9	96.6	71.2	94.6	82.6
JGSA [49]	81.1	84.3	76.5	99.0	75.8	97.2	85.7
ARTL [25]	92.5	91.8	76.9	99.6	77.1	97.5	89.2
MEDA [48]	90.8	91.4	74.6	97.2	75.4	96.0	87.6
MDA [53]	**94.0**	**92.6**	77.6	99.2	78.7	96.9	**89.8**
DAN [23]	80.5	78.6	62.8	99.6	63.6	97.1	80.4
RTN [28]	84.5	77.5	64.8	99.4	66.2	96.8	81.6
DANN [10]	82.0	79.7	67.4	99.1	68.2	96.8	81.6
ADDA [42]	86.2	77.8	68.9	98.4	69.5	96.2	82.9
CAN [50]	81.5	65.9	**98.2**	85.5	**99.7**	63.4	82.4
JDDA [3]	82.6	79.8	66.7	99.7	57.4	95.2	80.2
JAN [29]	85.4	84.7	70.0	**99.8**	68.6	**97.4**	84.3
GCAN [30]	82.7	76.4	62.6	**99.8**	64.9	97.1	80.6

tasks, the Inceptionresnetv2 features can sometimes achieve a better or similar result compared to Nasnetlarge, e.g., the P_LFC performance on Tab. 5, and the Inceptionresnetv2 model is substantially smaller and runs significantly faster than the Nasnetlarge model. Therefore, we recommend choosing one of these two models for feature selection.

Table 5: Average accuracy (%) of layer selection on Office-Home datasets with MDA method [53]

Task	Output	Softmax	LFC	P_LFC
Squeezenet [17]	42.0	42.0	**44.4**	-
Alexnet [21]	43.0	43.0	49.6	**50.4**
Googlenet [40]	53.0	53.0	62.9	**64.2**
Shufflenet [51]	45.9	45.9	53.5	**54.7**
Resnet18 [15]	49.5	49.5	59.2	**62.0**
Vgg16 [36]	47.8	47.8	57.1	**58.3**
Vgg19 [36]	48.4	48.4	58.0	**59.4**
Mobilenetv2 [35]	52.4	52.4	52.4	**64.7**
Nasnetmobile [55]	52.8	52.8	63.8	**64.6**
Resnet50 [15]	50.0	50.0	62.4	**62.5**
Resnet101 [15]	51.2	51.2	63.9	**64.7**
Densenet201 [16]	54.3	54.3	67.1	**69.5**
Inceptionv3 [41]	57.4	57.4	69.7	**70.4**
Xception [4]	59.4	59.4	72.0	**72.3**
Inceptionresnetv2 [39]	60.1	60.1	72.8	**73.8**
Nasnetlarge [55]	60.6	60.6	73.3	**73.6**

5. Discussion

Before the rise of the convolution neural network, handcrafted features were well used (e.g., SURF), and since deep features can substantially improve the performance of domain adaption, they are now widely used. However, most research has stuck with one pre-trained neural network, and researchers did not know which one will give the highest performance. In this paper, we are the first to present how different well-trained ImageNet models affect domain adapted classification. We have several novel observations. By exploring how different ImageNet models affect domain transfer accuracy, we find a roughly linear relationship between them, which suggests that Inceptionresnetv2 and Nasnetlarge are better sources for feature extraction. This differs from the conclusion in Kornblith et al. [19], which says these two neural networks do not transfer well in general classification problems. We see improved performance because of the better alignment of the source and target distribution using features from these two networks, while the domain shift issue does not exist in the general classification problem considered by Kornblith. We also find, perhaps surprisingly, that the best layer for feature extraction is the layer before the last fully-connected layer.

Table 6: Improvement of SVM and MDA model based on lowest and highest ImageNet model (Squeez.: Squeezenet, NAST.: Nasnetlarge and Impro.: Improvement)

Task	SVM_Squeez.	SVM_NAST.	Impro.
Office+Caltech-10	79.8	95.4	19.6%
Office-31	59.1	84.3	42.6%
Office-Home	38.4	66.6	73.4%

Task	MDA_Squeez.	MDA_NAST.	Impro.
Office+Caltech-10	92.9	97.4	4.8%
Office-31	70.2	88.0	25.4%
Office-Home	44.4	73.3	65.1%

Tab. 6 lists the improvement of SVM and MDA models using the Squeenezenet and Nasnetlarge neural networks for feature extraction. The improvement is nontrivial across all three datasets. Especially in the most difficult dataset, office-home, performance is notably improved 73.4% in SVM and 65.1% in MDA model. We hence can conclude that Nasnetlarge will be particularly useful in the case in which there is larger discrepancy between source and target domains. In addition, overall improvement suggests that better neural network features will be important for domain transfer accuracy.

Although we explore how to choose the best neural network source and layer for feature extraction, we notice that features from a lower performance ImageNet-trained network can produce a higher transfer accuracy in some tasks. Therefore, more work is needed to consider the combination of extracted features to produce even higher performance.

6. Conclusion

In this paper, we are the first to examine how features from many different ImageNet models affect domain adaptation. Extensive experiments demonstrate that a better ImageNet model will give a higher performance in transfer learning. We also find that the layer prior to the last fully connected layer is the best layer for extracting features.

References

[1] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Proc. Conf. on Empirical Methods in Natural Language Processing*, pages 120–128, 2006. 2

[2] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems*, pages 343–351, 2016. 2

[3] C. Chen, Z. Chen, B. Jiang, and X. Jin. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. *arXiv preprint arXiv:1808.09347*, 2018. 8

[4] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017. 3, 5, 8

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[6] F. Dorri and A. Ghodsi. Adapting component analysis. In *2012 IEEE 12th International Conference on Data Mining*, pages 846–851. IEEE, 2012. 2

[7] L. J. Edwards, K. E. Muller, R. D. Wolfinger, B. F. Qaqish, and O. Schabenberger. An R^2 statistic for fixed effects in the linear mixed model. *Statistics in medicine*, 27(29):6137–6157, 2008. 4

[8] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 2

[9] M. Ghifary, W. Bastiaan Kleijn, M. Zhang, and D. Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pages 2551–2559, 2015. 2

[10] M. Ghifary, W. B. Kleijn, and M. Zhang. Domain adaptive neural networks for object recognition. In *Pacific Rim international conference on artificial intelligence*, pages 898–904. Springer, 2014. 7, 8

[11] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2066–2073. IEEE, 2012. 2, 3, 4, 7, 8

[12] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf. Domain adaptation with conditional transferable components. In *International conference on machine learning*, pages 2839–2848, 2016. 2

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[14] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *IEEE International Conference on Computer Vision (ICCV)*, pages 999–1006. IEEE, 2011. 2

[15] 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 (CVPR)*, pages 770–778, 2016. 3, 5, 8

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 3, 5, 8

[17] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 3, 5, 8

[18] M. Jiang, W. Huang, Z. Huang, and G. G. Yen. Integration of global and local metrics for domain adaptation learning via dimensionality reduction. *IEEE Transactions on Cybernetics*, 47(1):38–51, 2017. 2

[19] S. Kornblith, J. Shlens, and Q. V. Le. Do better imagenet models transfer better? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2661–2671, 2019. 1, 2, 3, 5, 8

[20] A. Krizhevsky and G. Hinton. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40(7):1–9, 2010. 1

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 2, 3, 5, 8

[22] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016. 2

[23] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015. 2, 7, 8

[24] M. Long, Z. Cao, J. Wang, and M. I. Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1647–1657, 2018. 7

[25] M. Long, J. Wang, G. Ding, S. J. Pan, and S. Y. Philip. Adaptation regularization: A general framework for transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 26(5):1076–1089, 2013. 3, 7, 8

[26] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2200–2207, 2013. 3, 7, 8

[27] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer joint matching for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1410–1417, 2014. 2, 3, 7, 8

[28] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016. 7, 8

[29] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2208–2217. JMLR.org, 2017. 2, 7, 8

[30] X. Ma, T. Zhang, and C. Xu. Gcan: Graph convolutional adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8266–8276, 2019. 8

[31] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 5

[32] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Trans. on Neural Networks*, 22(2):199–210, 2011. 2

[33] M. M. Rahman, C. Fookes, M. Baktashmotlagh, and S. Sridharan. On minimum discrepancy estimation for deep domain adaptation. *arXiv preprint arXiv:1901.00282*, 2019. 7

[34] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010. 5

[35] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 3, 5, 8

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3, 5, 8

[37] B. Sun, J. Feng, and K. Saenko. Correlation alignment for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision Applications*, pages 153–171. Springer, 2017. 7, 8

[38] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, pages 443–450. Springer, 2016. 7

[39] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 3, 5, 8

[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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 3, 5, 8

[41] 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. 3, 5, 8

[42] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017. 2, 8

[43] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 2, 7

[44] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017. 5, 7

[45] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008. 2

[46] J. Wang, Y. Chen, S. Hao, W. Feng, and Z. Shen. Balanced distribution adaptation for transfer learning. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 1129–1134, 2017. 2, 3, 7, 8

[47] J. Wang, Y. Chen, L. Hu, X. Peng, and S. Y. Philip. Stratified transfer learning for cross-domain activity recognition. In *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–10. IEEE, 2018. 2

[48] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, and P. S. Yu. Visual domain adaptation with manifold embedded distribution alignment. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, pages 402–410, 2018. 2, 4, 7, 8

[49] J. Zhang, W. Li, and P. Ogunbona. Joint geometrical and statistical alignment for visual domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1859–1867, 2017. 3, 7, 8

[50] W. Zhang, W. Ouyang, W. Li, and D. Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3801–3809, 2018. 8

[51] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018. 3, 5, 8

[52] Y. Zhang, J.-P. Allem, J. B. Unger, and T. B. Cruz. Automated identification of hookahs (waterpipes) on instagram: an application in feature extraction using convolutional neural network and support vector machine classification. *Journal of medical Internet research*, 20(11):e10513, 2018. 3

[53] Y. Zhang and B. D. Davison. Modified distribution alignment for domain adaptation with pre-trained inception resnet. *arXiv preprint arXiv:1904.02322*, 2019. 3, 4, 7, 8

[54] Y. Zhang, S. Xie, and B. D. Davison. Transductive learning via improved geodesic sampling. In *Proceedings of the 30th British Machine Vision Conference*, 2019. 2, 3, 7, 8

[55] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018. 3, 5, 8

View publication stats