Deep Unsupervised Convolutional Domain Adaptation

Junbao Zhuo[1,2], Shuhui Wang[1*], Weigang Zhang[3], Qingming Huang[1,2]
[1] Key Lab of Intell. Info. Process., Inst. of Comput. Tech., CAS, Beijing, 100190, China
[2] University of Chinese Academy of Sciences, Beijing, 100049, China
[3] Harbin Inst. of Tech., Weihai, 264200, China
junbao.zhuo@vipl.ict.ac.cn,wangshuhui@ict.ac.cn,wgzhang@hit.edu.cn,qmhuang@ucas.ac.cn

ABSTRACT

In multimedia analysis, the task of domain adaptation is to adapt the feature representation learned in the source domain with rich label information to the target domain with less or even no label information. Significant research endeavors have been devoted to aligning the feature distributions between the source and the target domains in the top fully connected layers based on unsupervised DNN-based models. However, the domain adaptation has been arbitrarily constrained near the output ends of the DNN models, which thus brings about inadequate knowledge transfer in DNN-based domain adaptation process, especially near the input end. We develop an attention transfer process for convolutional domain adaptation. The domain discrepancy, measured in correlation alignment loss, is minimized on the *second-order correlation statistics* of the attention maps for both source and target domains. Then we propose Deep Unsupervised Convolutional Domain Adaptation (DUCDA) method, which jointly minimizes the supervised classification loss of labeled source data and the unsupervised correlation alignment loss measured on both convolutional layers and fully connected layers. The multi-layer domain adaptation process collaborately reinforces each individual domain adaptation component, and significantly enhances the generalization ability of the CNN models. Extensive cross-domain object classification experiments show DUCDA outperforms other state-of-the-art approaches, and validate the promising power of DUCDA towards large scale real world application.

KEYWORDS

Unsupervised domain adaptation; Deep learning; Attention model; Correlation alignment

1 INTRODUCTION

As one of the fundamental technologies in multimedia research domain, technologies for visual recognition have been developed from various aspects such as feature learning [20], kernel learning [22] and classification hierarchies [34]. To successfully construct a visual recognition system, a sufficient number of manually annotated images for each specific target domain are required beforehand.

(a) Source domain **(b) Target domain**

Figure 1: Attention maps of samples from source and target domain. Discriminative parts like handlebar, pedal and saddle are allocated with higher attention in the source domain (a), while in the target domain (b), low-level textures like strings of the wheels are allocated with higher attention than those discriminative parts like pedal and saddle.

With a large amount of labeled training data and substantial computation resources, state-of-the-art performances have been achieved by Deep Neural Networks (DNN) recently [19, 41]. Nevertheless, in real situations, it is usually impractical to obtain sufficient manually labeled training data for every new scenario. To alleviate this problem, domain adaptation [1, 8, 9, 17, 28, 33], which aims to adapt the feature representation learned in the source domain with rich label information to the target domain with less or even no label information, has received much attention in recent years.

Recent studies have shown that deep neural networks can learn more transferable features for domain adaptation [7, 13, 48], and potential results have been achieved on some cross-domain learning tasks. For example, the visual representations learned by deep Convolutional Neural Networks (CNN) are known to be invariant to low-level cues to some degree [29, 30], hence leveraging the deep features pre-trained on a large generic domain (e.g., ImageNet [6]) is believed to have good generalization ability to new domains. However, the test error of supervised methods generally increases in proportion to the discrepancy between the distributions of training and test examples in both theoretical [2, 3] and practical results [28, 42].

To reduce the performance degradation in domain adaptation, domain-invariant models [8, 15, 33, 37–39] are established to encourage appropriate knowledge transfer, which bridge the source and target domains in an isomorphic latent feature space. In a similar research direction, a fruitful line of prior works have focused on learning shallow features by jointly minimizing a distance metric of domain discrepancy [1, 14, 25, 28, 45], measured by maximum mean discrepancy (MMD) [24, 44] or correlation [40].

*Corresponding author.

MM'17, October 23–27, 2017, Mountain View, CA, USA.
 ISBN 978-1-4503-4906-2/17/10…$15.00
DOI: https://doi.org/10.1145/3123266.3123292

However, based on the plausible common knowledge that features of the top layers, i.e., the fully connected (FC) layers, deliver richer domain independent information that is close to the human cognitive knowledge, significant research endeavors have been devoted to aligning the feature distributions between the source and the target domains on the top FC layers based on unsupervised DNN-based domain adaptation models. Despite of the promising results on various benchmark cross-domain visual recognition experimental evaluation, the domain adaptation has been arbitrarily constrained near the output ends of the DNN models, which thus brings about inadequate knowledge transfer in DNN-based domain adaptation process, especially near the input end. In this paper, to encourage a more sufficient knowledge transfer for deep unsupervised domain adaptation, we study CNN-based domain adaptation from a new perspective, i.e., to minimize the domain discrepancy measured on **the feature responses of the convolutional layers**. The reasons can be addressed in details as follows.

First, activations of convolutional layers are crucial for visual analysis tasks such as image categorization, object detection and semantic segmentation. For example, in Faster RCNN [31], the convolutional feature maps can be used by region-based detectors to generate region proposals. The state-of-the-art GoogLeNet [41] and ResNet [19] contain large number of well-designed fully convolutional layers except for the last task-specific FC layers. Convolutional feature retains spatial information which is very important to describe the semantic context of an image and reduce the ambiguity. Unfortunately, such spatial information and semantic context are neglected when we only perform domain adaptation in FC layers. Therefore, it is necessary to extend the domain adaptation mechanism to convolutional layers to capture the spatial context information.

Second, the FC representations are constructed upon the convolutional feature representations. Existing methods that perform domain adaptation in FC layers implicitly assume that all discriminative information is well captured in convolutional layers. However, this assumption is not always true. As shown in Figure 1, discriminative parts like handlebar, pedal and saddle are well captured in the source domain. While in the target domain, these discriminative parts are not highlighted. On the contrary, the wheel strings are captured because of their rich texture in appearance. This information incompleteness in convolutional layer will be propagated to FC layers which can not be recovered by any well-designed FC-layer-based knowledge transfer mechanism. In this situation, encouraging domain adaptation in convolutional layer will ensure a better FC representation and a more sufficient domain discrepancy minimization.

The main challenge of convolutional domain adaptation is the high dimensionality of the convolutional feature responses. For example, the dimension of conv4 activation of AlexNet is 64896 ($13 \times 13 \times 384$) when flattened into a vector. Characterizing the distributions of such high dimension convolutional features requires large number of training data, and thus may lead to ill-posed solutions.

Inspired by the activation-based attention model [49], we develop an **attention transfer process for convolutional domain adaptation**. For a specific convolutional layer, the activation maps for both source and target domains are first calculated by L_p-norm pooling on all the convolutional response channels. Then domain discrepancy minimization is performed on the *second-order correlation statistics* of the attention maps, which describes the correlation between discriminative parts of an image or a visual scene. Compared to existing FC-layer-based domain adaptation techniques [1, 14, 24, 25, 28, 45], the informative spatial context in the convolutional attention maps can be preserved and transferred from the source domain to target domain. Consequently, more discriminative object parts can be effectively discovered in the target domain rather than only those with rich textures, thus the discriminative representation power in the convolutional responses can be significantly enhanced to obtain more effective feature response for both source and target domains. Even without label information in the target domain, more perception-level knowledge can be transferred from the source to target domain in the lower convolutional layers of a CNN, providing appropriate guidance to enhance the generalization ability of the deep CNN models.

In this paper, based on the above carefully designed attention transfer mechanism, we propose Deep Unsupervised Convolutional Domain Adaptation (DUCDA), which jointly minimizes the supervised classification loss of labeled source data and the unsupervised correlation alignment loss. As shown in Figure 2, the domain adaptation is performed in both convolutional layers and FC layers. The multi-layer domain adaptation process provides reinforcement to each individual domain adaptation component [24]. On one hand, adaptation in convolutional layers enforces better convolutional representations and thus better supports the domain adaptation in FC layers. On the other hand, the high-level semantic information of the source domain encoded in FC layers may guide the domain adaptation in convolutional layers to capture more discriminative patterns of images and reduce the influence of useless background information. The proposed DUCDA can be efficiently trained in an end-to-end fashion, which facilitates large scale real-world applications.

In summary, the key contributions of this paper are summarized as follows.

- We develop an **attention transfer process for convolutional domain adaptation**, which performs domain discrepancy minimization on the *second-order correlation statistics* of the attention maps. To the best of our knowledge, our work is the first to consider the domain adaptation in the convolutional layers.
- We design an end-to-end DUCDA network to perform domain adaptation in both convolutional layers and FC layers. The multi-layer domain adaptation process provides reinforcement to each individual domain adaptation component.
- Extensive cross-domain object classification experiments show that DUCDA outperforms state-of-the-art methods. Both quantity and quality results verify the promising power of our approach towards large scale real-world applications.

2 RELATED WORK

In domain adaptation, we focus on deep unsupervised domain adaptation methods which are closely related to our study. Another related literature is attention model which plays a critical role in human visual cognition.

Most deep domain adaptation methods follow a Siamese architectures [5] with two streams, representing the models for source

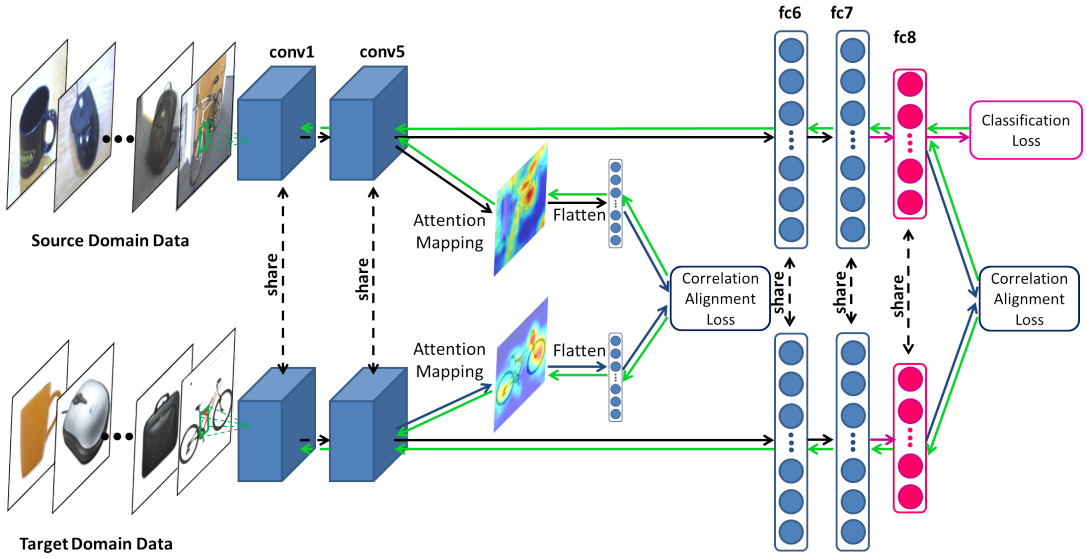


Figure 2: An illustration of DUCDA architecture. Weights of all layers are shared for both source and target domains. In this architecture, attention maps are extracted from conv5, while incorporating other convolutional layers is possible. We apply the correlation alignment loss to measure the domain discrepancy on the fc8 layer and vectorized attention maps extracted from conv5 and minimize it for aligning distributions between source and target domain.

and target domain, respectively. Apart from classification loss that depends on the labeled source data, deep domain adaptation models are trained in combination with an additional loss such as discrepancy loss [11, 24, 25, 40], adversarial loss [10, 32, 44] and reconstruction loss [51]. We roughly divide these methods into four categories according to the additional loss for incorporating with classification loss in domain adaptation.

Discrepancy-based methods explicitly measure the discrepancy between corresponding activation layers of the two streams of the Siamese architecture, i.e., the discrepancy between source and target domain. A single linear kernel was applied to only one layer to minimize Maximum Mean Discrepancy (MMD) in DDC [44] while the sum of MMDs defined between several FC layers, including the soft prediction layer, is considered in Deep Adaptation Network (DAN) [24]. Furthermore, multiple kernels for adapting these deep representations is used in DAN to substantially enhance adaptation effectiveness compared to a single kernel method used in [11] and [44]. In Joint Adaptation Networks [25], the joint distribution discrepancies of the multi-layer activations are considered instead of separate adaptations of marginal and conditional distributions which often require strong independence and/or smoothness assumptions on the factorized distributions. Instead of MMD, domain discrepancy is measured by the difference of covariance matrices between the corresponding activation layers of the two streams of the Siamese architecture in DCORAL [40]. In contrast to the above methods that adopt sharing weights of the two streams of the Siamese architecture, Rozantsev et al. [32] relaxed the sharing weight constraint but assumed that the weights of corresponding layers in the two models remain linearly related.

Adversarial discriminative models aim at encouraging domain confusion via an adversarial objective with respect to a domain discriminator. The Domain-Adversarial Neural Networks (DANN) [10] integrates a gradient reversal layer into the standard architecture in order to push the learnt features to maximize the loss of the domain classifier. The Adversarial Discriminative Domain Adaptation [43] uses an inverted label GAN loss rather than directly using the minimax loss to split the optimization process into two independent objectives for generator and discriminator, respectively.

Adversarial generative models combine the discriminative model with a generative component in general based on GANs [16]. The Coupled Generative Adversarial Networks (CoGAN) [23] consists of a tuple of GANs, each corresponding to one domain. Utilizing a weight sharing constraint, CoGAN can learn a joint distribution of multi-domain images without existence of corresponding images in different domains. Moreover, by enforcing the layers that decode high-level semantics in all GANs to share the weights, it enforces all GANs to decode the high-level semantics in the same way. The model proposed in [4] also exploits GANs that adapt source-domain images to appear as if they are drawn from the target domain. To penalize large differences between source and generated images for foreground pixels only, Bousmalis et al. [4] proposed to minimize a masked Pairwise Mean Squared Error which only calculates the masked pixels (foreground) of the source and the generated images. Furthermore, it is able to generalize to object classes unseen during the training phase as the model decouples the process of domain adaptation from the task-specific architecture.

Data reconstruction based methods incorporate a reconstruction loss that minimizes the difference between input and reconstructed input. The Deep Reconstruction Classification Network [12] combines the standard convolutional network for source label prediction with a de-convolutional network [51] for target data reconstruction which can be viewed as an auxiliary task to support the adaptation of the label prediction.

As another related literature mentioned above, attention is proved to be useful in computer-vision-related tasks such as image captioning [46], visual question answering [47], as well as in weakly-supervised object localization [27] and classification [26]. Gradient-based attention model computes a Jacobian of network output w.r.t. the input [35], and guided backpropagation [36] is proposed to improve gradient-based attention. In [50], Zeiler used "deconvnet" which shares weights with the original network to project certain features onto the image plane. In [49], Zagoruyko et al. proposed activation-based attention model to guide the training of a weak CNN model by forcing it to mimic the attention maps of a powerful teacher network. Regarding the activation-based attention maps of convolutional layers as middle level features, adaptation strategies on FC layers can also be applied to convolutional layers as shown in this paper.

3 METHOD

We consider the unsupervised domain adaptation scenario where labels for target domain data are not available. Despite of the large domain discrepancy between source and target domain and the absence of labels of target domain data, we want to learn a single deep CNN that performs well on both source and target domains. We focus on domain adaptation in convolutional layers which is equally important as the domain adaptation in FC layers. Based on activation-based attention model, activations of convolutional layers can be distilled into lower-dimensional features which enables effective domain adaptation in convolutional layers. We adopt correlation alignment loss (CAL) in our method as the semantic context of attention maps can be effectively modeled via correlation. Intuitively, adaptation in convolutional layers will boost the adaptation in FC layers as better convolutional representations are learnt, and thus lead to better representations of FC layers. The CAL loss is also applied to FC layers to construct a multi-layer domain adaptation process in both convolutional layers and FC layers. The DUCDA architecture is shown in Figure 2.

3.1 Common Notations

We first introduce some common notations used in this paper. Suppose we are given N_S source-domain training examples $D_S = \{z_i^s\}_{i=1}^{N_S}$ with labels $L_S = \{y_i\}, y_i \in \{1, \ldots, L\}$, and N_T unlabeled target examples $\{z_j^t\}_{j=1}^{N_T}$, where z^s and z^t are the raw images from source and target domain respectively and L is the number of categories. As our network is extended from AlexNet, we use the symbols in AlexNet here for simplicity. Let $\phi_{conv5}(\cdot;\theta_{conv5})$ be the subnetwork composed of conv1~conv5 parameterized by θ_{conv5} and let $A_S = \{a_i^s\}_{i=1}^{N_S}$ and $A_T = \{a_j^t\}_{j=1}^{N_T}$ be the sets of activations of conv5 layer where $a_i^s = \phi(z_i^s;\theta_{conv5}), z_i^s \in D_S$ and $a_j^t = \phi(z_j^t;\theta_{conv5}), z_t^t \in D_T$. Denote by $\psi(\cdot;\theta_{cls})$ the subnetwork composed of fc6~fc8 parameterized by θ_{cls} that maps conv5 activations to a class-conditional distribution. Let $vec(\cdot)$ be the flatten operation that transforms the attention map into vectorized form and let $X_S = \{x_i^s\}_{i=1}^{N_S}, x_i \in R^d$ and $X_T = \{x_j^t\}_{j=1}^{N_T}, x_j \in R^d$ be the sets of attention maps in vectorized form. Let $F_S = \{f_i^s\}_{i=1}^{N_S}$ and $F_T = \{f_j^t\}_{j=1}^{N_T}$ be the sets of fc8 activations where $f_i^s = \psi(a_i^s;\theta_{cls}), a_i^s \in A_S$ and $f_j^t = \psi(a_j^t;\theta_{cls}), a_j^t \in A_T$. Let B_S be a batch randomly selected from X_S (or F_S) with n_S examples, and B_T be a batch randomly selected from X_T (or F_T) with n_T examples.

3.2 Activation-based Attention Model

As revealed by Yosinski et al. [48], feature transferability gets worse on conv4~conv5 of AlexNet. Hence, adaptation in convolutional layers is as important as adaptation in FC layers. Adaptation in convolutional layers is essential since FC layers activations are computed on the basis of convolutional layers activations. If useful information can not be captured in convolutional layers, the representation incompleteness will be propagated to FC layers and can not be recovered. In this circumstance, adaptation merely in FC layers will not work. However, adaptation in convolutional layers is rarely considered in previous study. One reason may be attributed to the high dimension of convolutional layer activations. One way to solve this problem is to distill the convolutional layer activations into low dimensional representations via activations-based attention mapping [49]. Specifically, given an activation tensor

Algorithm 1 The DUCDA$_{conv5}$ learning algorithm.

Input: Labeled source data: $D_S = \{z_i^s, y_i\}_{i=1}^{N_S}$; Unlabeled target data: $D_T = \{z_j^t\}_{j=1}^{N_T}$;

Output: DUCDA$_{conv5}$ learnt parameters:$\hat{\theta}_{conv5}$ and $\hat{\theta}_{cls}$;

Initialize parameters θ_{conv5} and θ_{cls} with pretrained AlexNet. Re-initialize the weight of fc8 layer with N(0, 0.005).
repeat
  for each source batch B_{S_img} and target batch B_{T_img} **do**
    Calculate $B'_S = \phi(B_{S\_img};\theta_{conv5})$
    Calculate $B'_T = \phi(B_{T\_img};\theta_{conv5})$
    Calculate $B_S = Att(B'_S)$
    Calculate $B_T = Att(B'_T)$
    Calculate L_{CLS}
    Calculate $\frac{\partial L_{CLS}}{\partial \theta_{cls}}$
    Update θ_{cls} using SGD
    Calculate $\frac{\partial L_{CLS}}{\partial B'_S}\frac{\partial B'_S}{\partial \theta_{conv5}}$
    Calculate $L_{CAL_{conv5}}$
    Calculate $\frac{\partial L_{CAL_{conv5}}}{\partial B_S}\frac{\partial B_S}{\partial B'_S}\frac{\partial B'_S}{\partial \theta_{conv5}}$
    Calculate $\frac{\partial L_{CAL_{conv5}}}{\partial B_T}\frac{\partial B_T}{\partial B'_T}\frac{\partial B'_T}{\partial \theta_{conv5}}$
    Update θ_{conv5} using SGD
  end for
until *Convergence Or Reach Maximum Iterations*

$A \in R^{C\times H\times W}$ which consists of C channels with spatial dimensions $H \times W$, a mapping function F_{att} that takes the above convolutional layer activations A (3D tensor) as input and outputs a spatial attention map is defined as

$$F_{att} : R^{C\times H\times W} \rightarrow R^{H\times W} \tag{1}$$

As the absolute value of a hidden neuron activation indicates the importance of that neuron w.r.t. the specific input, we can construct spatial attention map by computing statistics of these absolute values across the channel dimension. Specifically, we consider the

following spatial attention mappings:

$$(F_{att}(A))_{i,j} = \sum_{ch=1}^{C} |A_{ch,i,j}|^p \quad (2)$$

where $i \in \{1, 2, ..., H\}$ and $j \in \{1, 2, ..., W\}$ are spatial indexes.

To effectively estimate the covariance matrices described in next subsection, we apply a logarithmic function $Log(\cdot)$ over the above attention maps. Therefore, the element x_i^s and x_j^t in X_S and X_T are actually computed as $x_i^s = Log(vec(F_{att}(a_i^s))), a_i^s \in A_S$ and $x_j^t = Log(vec(F_{att}(a_j^t))), a_j^t \in A_T$. We replace $Log(vec(F_{att}(\cdot)))$ with $Att(\cdot)$ for simplicity. Note that, $Att(\cdot)$ is differentiable. Now, the knowledge of convolutional layer activations is distilled into a d-dimensional feature with appropriate values and can be well adapted similar to methods that adapt in FC layers like [24] and [40].

3.3 Correlation Alignment

Aligning the second-order statistics-correlation of the source and target distributions has shown superior performance to MMD-based domain adaptation methods like [44] and [24]. The goal is to transfer such correlations from source domain to the target domain. As some discriminative parts of objects are usually effectively captured in attention maps and these parts are usually positively correlated, CAL is more appropriate for adapting attention maps of convolutional layers. Hence, we also adopt CAL in our method.

The CAL is defined as the distance between the covariances of the vectorized attention maps of conv5 (or fc8 features) of source and target samples:

$$L_{CAL} = \frac{1}{4d^2}||C_S - C_T||_F^2 \quad (3)$$

where $||\cdot||_F^2$ denotes the squared matrix Frobenius norm and C_S and C_T are covariance matrices of the source and target samples denoted by:

$$C_S = \frac{1}{n_S - 1}(B_S^T B_S - \frac{1}{n_S}(\mathbf{1}^T B_S)^T(\mathbf{1}^T B_S)) \quad (4)$$

$$C_T = \frac{1}{n_T - 1}(B_T^T B_T - \frac{1}{n_T}(\mathbf{1}^T B_T)^T(\mathbf{1}^T B_T)) \quad (5)$$

where $\mathbf{1}$ is a column vector with all elements equal to 1. Applying chain rule, the gradient w.r.t. the input features can be calculated as:

$$\frac{\partial L_{CAL}}{\partial B_S^{ij}} = \frac{1}{d^2(n_S - 1)}((B_S^T - \frac{1}{n_S}(\mathbf{1}^T B_S)^T \mathbf{1}^T)^T (C_S - C_T))^{ij} \quad (6)$$

$$\frac{\partial L_{CAL}}{\partial B_T^{ij}} = \frac{-1}{d^2(n_T - 1)}((B_T^T - \frac{1}{n_T}(\mathbf{1}^T B_T)^T \mathbf{1}^T)^T (C_S - C_T))^{ij} \quad (7)$$

The gradients $\frac{\partial L_{CAL}}{\partial B_S}$ and $\frac{\partial L_{CAL}}{\partial B_T}$ can be easily propagated back to conv5 (or fc8) which enables end-to-end training. Note that, CAL can be applied to other convolutional layers and FC layers.

3.4 End-to-end Domain Adaptation in Convolutional layer

As mentioned above, adaptation in convolutional layers is essentianl. Adaptation in FC layers will not help if effective information can not be captured in convolutional layers. The learnt representations are required to be both discriminative and domain invariant. Joint training with the classification loss combined with CAL on on convolutional attention maps is likely to learn representation that work well on both the source and target domain. On one hand, overfitting to the source domain often occurs by merely minimizing the classification loss, which limits the performance on the target domain. On the other hand, minimizing the CAL alone might lead to degenerated representations. To well understand the effectiveness of adaptation in convolutional layers, we design a deep CNN model that adapts only in conv5 denoted by DUCDA$_{conv5}$. Our objective for DUCDA$_{conv5}$ for a mini-batch is:

$$\begin{aligned} L_{conv5} &= L_{CLS} + \lambda_1 L_{CAL_{conv5}} \\ &= l_c(\psi(B'_S; \theta_{cls}), L_{B'_S}) + \lambda_1 L_{CAL_{conv5}} \end{aligned} \quad (8)$$

where $L_{CAL_{conv5}}$ is calculated according to equations (3)~(5) with B_S and B_T randomly selected from X_S and X_T respectively and B'_S is the batch selected from A_S corresponding to B_S. Let B'_T is the batch selected from A_T corresponding to B_T. That is, $B_S = Att(B'_S)$ and $B_T = Att(B'_T)$. λ_1 is a weight that can be tuned to achieve better trade-off between the adaptation and classification accuracy on the source domain and l_c is the cross-entropy loss. $\frac{\partial L_{CLS}}{\partial \theta_{cls}}$, the gradient of θ_{cls} can be used to train fc6~fc8 and $\frac{\partial L_{CLS}}{\partial B'_S}$, and the gradient of B'_S can be propagated back to conv1~conv5 to update θ_{conv5} incorporated with $\frac{\partial L_{CAL_{conv5}}}{\partial B_S}\frac{\partial B_S}{\partial B'_S}$ and $\frac{\partial L_{CAL_{conv5}}}{\partial B_T}\frac{\partial B_T}{\partial B'_T}$ using chain rule. That means DUCDA$_{conv5}$ can be trained in an end-to-end fashion. The whole training algorithm of DUCDA$_{conv5}$ is shown in Algorithm 1.

3.5 End-to-end Domain Adaptation both in conv5 and fc8

Adaptation in convolutional layers enforces the network to learn effective filters and results in more domain-invariant and discriminative convolutional features which helps FC layers to learn better representations. Therefore, in addition to applying adaptation in conv5, we also add CAL to fc8 to adapt fc8 and we call this model as DUCDA. Moreover, adaptation in fc8 layer may penalize the activations of some irrelevant neurons which in turn enforces the conv5 layer to pay less attention to irrelevant patterns of input images. Hence, adaptation in convolutional layers and FC layers will mutually reinforce each other and finally boost the performance. Our objective for DUCDA for a mini-batch is:

$$L = L_{conv5} + \lambda_2 L_{CAL_{fc8}} \quad (9)$$

where $L_{CAL_{fc8}}$ is calculated according to equations (3)~(5) with B_S and B_T selected from F_S and F_T respectively. The gradient of $L_{CAL_{fc8}}$ w.r.t. fc8 features can be calculated and propagated to fc8 layer incorporated with the gradient brought by classification loss L_{CLS}. λ_2 is the weight together with λ_1 that can be tuned to achieve better trade-off between the adaptation and classification accuracy on the source domain.

3.6 Discussions

Compare to methods that adapt merely in FC layers such as DDC [44], DAN [24] and DCORAL [40], DUCDA performs adaptation in convolutional layers that contain spatial information which is lost in FC layers. Correlated semantic context can be well transferred from source domain to target domain via adaptation in convolutional

layers in DUCDA. Such semantic context is very useful for object recognition in target domain.

By utilizing adaptation both in convolutional layers and FC layers, DUCDA leverages the power of multi-layer adaptation where adaptation in convolutional layers and FC layers mutually reinforce each other. On one hand, adaptation in convolutional layers enforces better convolutional representations to be learnt which leads to better representations in FC layers. On the other hand, high-level semantic information of FC layers will guide the convolutional layers to capture more discriminative patterns of images and ignore irrelevant patterns.

Moreover, DUCDA is computationally efficient and simpler to optimize compare to DAN which needs to investigate optimal kernel parameter via quadratic programming.

4 EXPERIMENT

Extensive experiments on two domain adaptation benchmarks were conducted to evaluate our methods. Analysis is focusing on the effectiveness of adaptation in convolutional layers and the efficacy of adaptation both in convolutional and FC layers.

4.1 Setup

We conducted our experiments on Office31 and Office-10 + Caltech-10 datasets.

Office-31 [33] is a standard benchmark for visual domain adaptation, which contains 4,652 images in total within 31 categories collected from office environment in three image domains: Amazon (A), comprising images downloaded from amazon.com, Webcam (W) and DSLR (D), comprising images taken by web camera and digital SLR camera with different photographical settings, respectively. Following the standard protocol [7, 9, 14, 24, 44], all the source data with labels and all the target data without labels were used. All six domain adaptation tasks A→W, A→D, W→D, W→A, D→A, and D→W were evaluated for unbiased evaluation.

Office-10 + Caltech-10 [14] that comprises of images selected from the 10 common categories shared by the Office-31 and Caltech-256 (C) [18] is also widely adopted to evaluate domain adaptation methods. Another six domain adaptation tasks A→C, W→C, D→C, C→A, C→W, and C→D can be built for unbiased evaluation.

The original top FC layer (fc8) of the pre-trained AlexNet was removed and a new FC layer with 31/10 hidden neurons (the number of categories for Office-31/Office-10 + Caltech-10) was added. The weights of the added fc8 layer ware randomly initialized with N(0, 0.005). The learning rate of the added fc8 layer was set to 10 times that of the lower layers. We used stochastic gradient descent (SGD) with 0.9 momentum and the weight decay was set to 0.0005. The batch size and base learning rate were set to 256 and 0.001 respectively. Note that the activations of conv5 (behind relu5 and in front of pool5) were used to calculate attention maps whose dimension is 169. The batch size should be sufficiently large to effectively estimate the covariance matrices C_S and C_T as the dimension of attention map is large. As suggested in [40], the weight λ_2 of the $L_{CAL_{fc8}}$ was set in such way that at the end of training the classification loss is the same order of magnitude as L_{CLS}. However, there is no heuristic way to set the weight λ_1 of $L_{CAL_{conv5}}$ applied in conv5 layer. Therefore, we extensively investigated appropriate λ_1 via grid search. All of our experiments were implemented with Caffe [20].

We compared DUCDA to a variety of published methods: GFK [15], TCA [28], CNN [21] (no adaptation), DDC [44], DAN [24] and DCORAL [40]. Specifically, GFK and TCA are not end-to-end deep methods. TCA aims at learning some transferable components across domains in a reproducing kernel Hilbert space regularized with MMD. GFK is a widely-adopted method for our datasets which bridges the source and target domain by interpolating them across intermediate subspaces along a geodesic path. DDC is a cross-domain variant of CNN with single-layer adaptation via single-kernel MMD. DAN utilizes a multi-kernel selection method for better mean embedding matching and adapts in multiple layers to undo the dataset bias as feature transferability significantly drops on fc6~fc8. Domain distributions is aligned via second order statistic-correlation in DCORAL. For fair comparison, we also reported the performances of DAN_{fc7} and DAN_{fc8}, variants of DAN that adapt in single FC layer as DUCDA_{conv5} just adapts in single convolutional layer as well.

As revealed in [37], fc7 feature fine-tuned on the source domain (FT7) achieved better performance than generic pre-trained features. Therefore, for Office-31, FT7 was used to train a linear SVM [8, 37] in GFK and TCA. For Office-10 + Caltech-10, accuracies for TCA, GFK, DCC and DAN reported in [24] are directly reported here. In [24], instead of FT7, SURF features were used to a linear SVM in GFK and TCA.

4.2 Result and Analysis

The results on the first six Office-31 unsupervised domain adaptation tasks are shown in Table 1, and the results on the other six Office-10 + Caltech-10 adaptation tasks are shown in Table 2.

From Table 1 and Table 2 we can see that DUCDA_{conv5} is comparable to DDC, DAN_{fc7}, DAN_{fc8} and DCORAL. Note that these methods apply adaptation in fc7 or fc8 layers, whose representations are more discriminative than conv5 layer activations. This validates the effectiveness of adaptation in convolutional layers.

From Table 1, we can see that DUCDA shows better average accuracy than DCORAL and the other six baseline methods, proving that adapting in convolutional layer significantly boosts the adaptation in FC layers. In three out of six tasks, DUCDA achieves the highest accuracies. For the other three tasks, the margin between DUCDA and the best baseline method is small. However, as shown in Table 2, DUCDA does not outperform DAN in average. We attribute this to the small size of Office-10 + Caltech-10 dataset as we need sufficient samples to effectively estimate the covariance matrices in DUCDA. In this setting, DUCDA outperforms DCORAL similar to Office-31 setting which validates that adaptation in convolutional layer boosts the adaptation in FC layers.

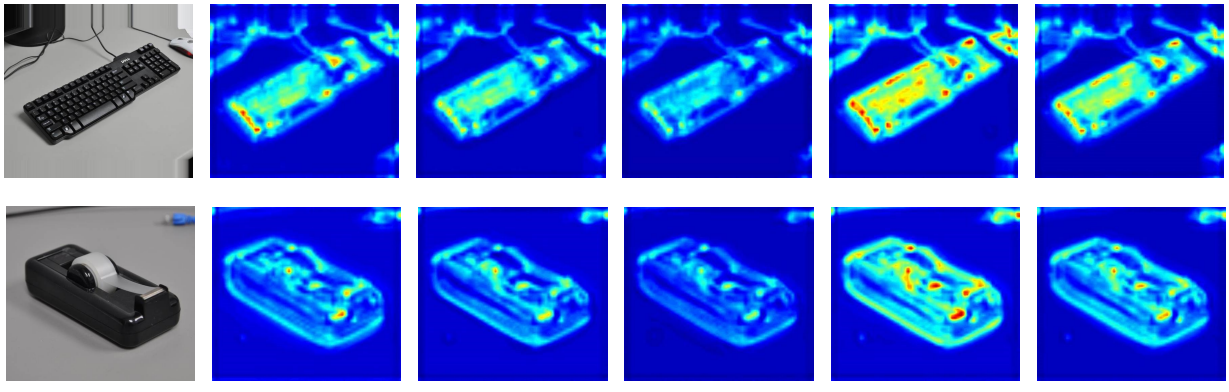
To get a better understanding of DUCDA, in Figure 3, we visualized the attention maps extracted with several networks, including the pre-trained AlexNet, AlexNet fine-tuned (AlexNet_{FT}) on source domain (Amazon), DCORAL, DUCDA_{conv5} and DUCDA to see what effects that adaptation in convolutional layer brings. DCORAL, DUCDA_{conv5} and DUCDA were learnt specifically to A→D task and the two input samples were drawn from DLSR dataset. Filters general to 1000 classes were learnt in the original AlexNet

Table 1: Accuracy on Office-31 dataset with standard unsupervised adaptation protocol [40].

Algorithms	A→W	A→D	D→A	D→W	W→A	W→D	Avg
GFK	54.7±0.0	52.4±0.0	43.2±0.0	92.1±0.0	41.8±0.0	96.2±0.0	63.4
TCA	45.5±0.0	46.8±0.0	36.4±0.0	81.1±0.0	39.5±0.0	92.2±0.0	56.9
CNN	61.6±0.5	63.8±0.5	51.1±0.6	95.4±0.3	49.8±0.4	99.0±0.2	70.1
DDC	61.8±0.4	64.4±0.3	52.1±0.8	95.0±0.5	52.2±0.4	98.5±0.4	70.6
DAN_{fc7}	63.2±0.2	65.2±0.4	52.3±0.4	94.8±0.4	52.1±0.4	98.9±0.3	71.1
DAN_{fc8}	63.8±0.4	65.8±0.4	52.8±0.4	94.6±0.5	51.9±0.5	98.8±0.6	71.3
DCORAL	66.8±0.6	66.4±0.4	52.8±0.2	95.7±0.3	51.5±0.3	99.2±0.1	72.1
DAN	**68.5**±0.4	67.0±0.4	**54.0**±0.4	96.0±0.3	**53.1**±0.3	99.0±0.2	72.8
$DUCDA_{conv5}$	62.6±0.6	64.7±0.6	52.4±0.5	96.0±0.5	49.6±0.6	99.4±0.2	70.8
DUCDA	68.3±0.4	**68.3**±0.6	53.6±0.4	**96.2**±0.2	51.6±0.6	**99.7**±0.2	**73.0**

Table 2: Accuracy on Office-10 + Caltech-10 dataset with standard unsupervised adaptation protocol [14].

Algorithms	A→C	W→C	D→C	C→A	C→W	C→D	Avg
GFK	41.4±0.0	26.4±0.0	36.4±0.0	56.2±0.0	43.7±0.0	42.0±0.0	41.0
TCA	42.7±0.0	34.1±0.0	35.4±0.0	54.7±0.0	50.5±0.0	50.3±0.0	44.6
CNN	83.8±0.3	76.1±0.5	80.8±0.4	91.1±0.2	83.1±0.3	89.0±0.3	84.0
DDC	84.3±0.5	76.9±0.4	80.5±0.2	91.3±0.3	85.5±0.3	89.1±0.3	84.6
DAN_{fc7}	84.7±0.3	78.2±0.5	81.8±0.3	91.6±0.4	87.4±0.3	88.9±0.5	85.4
DAN_{fc8}	84.4±0.3	80.8±0.4	81.7±0.2	91.7±0.3	90.5±0.4	89.1±0.4	86.4
DAN	**86.0**±0.5	**81.5**±0.3	82.0±0.4	92.0±0.3	**92.0**±0.4	90.5±0.2	**87.3**
DCORAL	84.7±0.3	79.3±0.6	82.8±0.5	92.4±0.2	91.1±0.6	91.4±0.6	87.0
$DUCDA_{conv5}$	84.7±0.5	79.4±0.5	**83.0**±0.6	92.7±0.6	85.6±0.6	90.0±0.4	85.9
DUCDA	84.8±0.5	80.2±0.1	82.5±0.6	**92.8**±0.6	91.6±0.6	**91.7**±0.4	**87.3**



(a) Input **(b) AlexNet** **(c) $AlexNet_{FT}$** **(d) DCORAL** **(e) $DUCDA_{conv5}$** **(f) DUCDA**

Figure 3: Attention maps of two samples of target domain DSLR. These attention maps are extracted from conv5 layers in several networks, including the original AlexNet, AlexNet fine-tuning on source domain (Amazon), DCORAL, $DUCDA_{conv5}$ and DUCDA. The values of attention maps on objects in (e) and (f) are generally larger than those extracted from other three methods. Moreover, attention maps on objects in (e) and (f) tends to cover the whole object.

while we want the filters adapted to our task (31 categories). Intuitively, applying adaptation in convolutional layer will enforce the filters to learn to capture more effective patterns which means the filters will have high response to some specific patterns. In Figure 2, we can see that with adaptation in conv5 layer ((e) and (f)), the filters have higher level attention on objects in general compare to other networks without adaptation in conv5 layer. This phenomenon indicates that adapting in conv5 layer enforces the filters to capture discriminative parts of objects. As mentioned above, if useful information cannot be captured in convolutional layers, the representation incompleteness will be propagated to FC layers and cannot be recovered. We showed that there are some samples that can be correctly classified by $DUCDA_{conv5}$ but misclassified by DCORAL in Figure 4. DCORAL and $DUCDA_{conv5}$ were also learnt specifically to A→D task and all samples were drawn from DLSR dataset in Figure 4.

Figure 4: Samples misclassified by DCORAL (red) while correctly classified by DUCDA$_{conv5}$ (green) for $A \rightarrow D$.

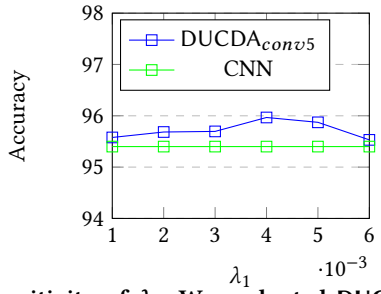


Figure 5: Sensitivity of λ_1. We evaluated DUCDA$_{conv5}$ in A→D tasks. Green line shows results of AlexNet fine-tuned on Amazon.

The most important thing is that comparing to methods that do not adopt adaptation in conv5 layer, DUCDA$_{conv5}$ and DUCDA push the attention maps to cover the entire objects. There are more isolated clusters in the attention maps extracted with AlexNet, AlexNet$_{FT}$ and DCORAL. On the contrary, the isolated clusters are grouped together to nearly cover the entire object in the attention maps extracted with DUCDA$_{conv5}$ and DUCDA. Utilizing these semantic context, DUCDA significantly outperforms DCORAL.

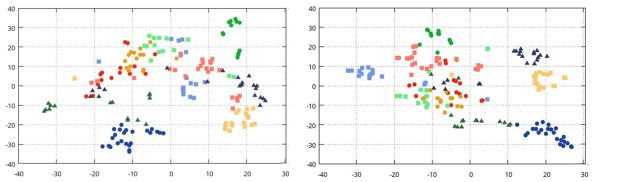
Moreover, compared to DUCDA$_{conv5}$, attention maps extracted with DUCDA are smoother, which indicates that adaptation in FC layers also boosts the adaptation in convolutional layers. High-level semantic information in FC layers prevents convolutional layers to paying too much attention to some specific parts.

4.3 Parameter Sensitivity

We investigated the effects of the hyperparameter λ_1. Figure 5 gives an illustration of the variation of transfer classification performance as $\lambda_1 \in \{0.001, 0.002, 0.003, 0.004, 0.005, 0.006\}$ on tasks A→D. We can observe that the DUCDA$_{conv5}$ accuracy first increases and then decreases as λ_1 varies. This confirms the validity of jointly learning deep features and adapting distribution discrepancy, since a good trade-off between them can enhance feature transferability.

4.4 Feature Visualization

To demonstrate the effectiveness of the learned features of DUCDA, similar to DAN [24], we plotted in Figures 6 the t-SNE embeddings of the images in task A→D with DCORAL features and DUCDA features, respectively. We visualized only 10 categories for clarity.



(a) DCORAL (b) DUCDA

Figure 6: t-SNE visualization of DCORAL features and DUCDA features of target domain DSLR (Best viewed in color). Blue squares, green triangles and yellow squares in (b) are better separated than those in (a).

We can observe that with DCORAL features, the target points are not discriminated very well compared to DUCDA features. For example, classes represented with blue square, green triangle and yellow square in (b) are better separated than those in (a). Apparently, DUCDA learns better representations than DCORAL which validates the effectiveness of adaptation in convolutional layers.

5 CONCLUSION

We study unsupervised domain adaptation from the convolutional perspective and develop an attention transfer process for convolutional domain adaptation. The domain discrepancy is minimized on *second-order correlation statistics* of the attention maps for both source and target domains. Then we propose DUCDA, which jointly minimizes the supervised classification loss of labeled source data and the unsupervised correlation alignment loss measured on both convolutional layers and FC layers. Extensive experiments show that DUCDA outperforms state-of-the-art approaches, and validate the promising power of DUCDA towards large scale real world application. In future research, it is interesting to extend DUCDA to ResNet or GoogLeNet. Another promising direction is to apply dimension reduction to produce more compact attention maps on multiple convolutional layers.

6 ACKNOWLEDGE

This work was supported in part by National Natural Science Foundation of China: 61672497, 61332016, 61620106009, 61650202 and U1636214, in part by National Basic Research Program of China (973 Program): 2015CB351802 and in part by Key Research Program of Frontier Sciences of CAS: QYZDJ-SSW-SYS013.

REFERENCES

[1] Mahsa Baktashmotlagh, Mehrtash T. Harandi, Brian C. Lovell, and Mathieu Salzmann. 2013. Unsupervised Domain Adaptation by Domain Invariant Projection. In *ICCV*. 769–776.

[2] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2006. Analysis of representations for domain adaptation. In *NIPS*. 137–144.

[3] John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, Boomboxes And Blenders: Domain Adaptation For Sentiment Classification. *Acl* 31, 2 (2007), 187–205.

[4] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. 2016. Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks. *arXiv preprint arXiv:1612.05424* (2016).

[5] Jane Bromley, Isabelle Guyon, Yann Lecun, Eduard SÃďckinger, and Roopak Shah. 1993. Signature Verification Using a Siamese Time Delay Neural Network. In *NIPS*. 737–744.

[6] Jia Deng, Wei Dong, R. Socher, Li Jia Li, Kai Li, and Fei Fei Li. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*. 248–255.

[7] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2013. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *Computer Science* 50, 1 (2013), 815–830.

[8] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. 2013. Unsupervised Visual Domain Adaptation Using Subspace Alignment. In *ICCV*. 2960–2967.

[9] Yaroslav Ganin and Victor Lempitsky. 2014. Unsupervised Domain Adaptation by Backpropagation. *arXiv preprint arXiv:1409.7495v1* (2014).

[10] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, Fran Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research* 17, 1 (2016), 2096–2030.

[11] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. 2015. Domain Generalization for Object Recognition with Multi-task Autoencoders. In *ICCV*.

[12] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. 2016. *Deep Reconstruction-Classification Networks for Unsupervised Domain Adaptation.* Springer International Publishing.

[13] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. In *ICML*.

[14] Boqing Gong, Kristen Grauman, and Fei Sha. 2013. Connecting the Dots with Landmarks: Discriminatively Learning Domain-Invariant Features for Unsupervised Domain Adaptation. In *ICML*.

[15] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*. 2066–2073.

[16] Ian J Goodfellow, Jean Pougetabadie, Mehdi Mirza, Bing Xu, David Wardefarley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Zoubin Ghahramani, and Max Welling. 2014. Generative Adversarial Nets. *NIPS* 3 (2014), 2672–2680.

[17] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. 2011. Domain adaptation for object recognition: An unsupervised approach. 24, 4 (2011), 999–1006.

[18] Gregory Griffin, Alex Holub, and Pietro Perona. 2007. Caltech-256 Object Category Dataset. *California Institute of Technology* (2007).

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[20] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *ACM international conference on Multimedia*. ACM, 675–678.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *NIPS*. 1097–1105.

[22] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, Vol. 2. IEEE, 2169–2178.

[23] Ming Yu Liu and Oncel Tuzel. 2016. Coupled Generative Adversarial Networks. *NIPS* (2016).

[24] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. 2015. Learning Transferable Features with Deep Adaptation Networks. *ICML* (2015), 97–105.

[25] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S. Yu. 2013. Transfer Feature Learning with Joint Distribution Adaptation. In *ICCV*. 2200–2207.

[26] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. 2014. Recurrent models of visual attention. In *NIPS*, Vol. 3. 2204–2212.

[27] M Oquab, L. Bottou, I. Laptev, and J. Sivic. 2015. Is object localization for free? - Weakly-supervised learning with convolutional neural networks. In *CVPR*. 685–694.

[28] Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. 2011. Domain Adaptation via Transfer Component Analysis. *IEEE Transactions on Neural Networks* 22, 2 (2011), 199–210.

[29] Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko. 2015. Learning Deep Object Detectors from 3D Models. In *ICCV*. 1278–1286.

[30] Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko. 2015. What Do Deep CNNs Learn About Objects? *Computer Science* (2015).

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NIPS*.

[32] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. 2016. Beyond Sharing Weights for Deep Domain Adaptation. *arXiv preprint arXiv:1603.06432* (2016).

[33] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. 2010. Adapting Visual Category Models to New Domains. In *ECCV*.

[34] Li Shen, Shuhui Wang, Gang Sun, Shuqiang Jiang, and Qingming Huang. 2013. Multi-level discriminative dictionary learning towards hierarchical visual categorization. In *CVPR*. 383–390.

[35] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *Computer Science* (2013).

[36] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. 2014. Striving for Simplicity: The All Convolutional Net. *arXiv preprint arXiv:1412.6806* (2014).

[37] Baochen Sun, Jiashi Feng, and Kate Saenko. 2015. Return of Frustratingly Easy Domain Adaptation. *Computer Science* (2015).

[38] Baochen Sun and Kate Saenko. 2014. From Virtual to Reality: Fast Adaptation of Virtual Object Detectors to Real Domains. In *BMVC*.

[39] Baochen Sun and Kate Saenko. 2015. Subspace Distribution Alignment for Unsupervised Domain Adaptation. In *BMVC*. 24.1–24.10.

[40] Baochen Sun and Kate Saenko. 2016. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In *ECCV Workshop on Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV)*.

[41] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *CVPR*. 1–9.

[42] Antonia Torralba and Alexei A. Efros. 2011. Unbiased look at dataset bias. In *CVPR*. 1521–1528.

[43] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial Discriminative Domain Adaptation. In *NIPS Workshop on Adversarial Training, (WAT)*.

[44] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep Domain Confusion: Maximizing for Domain Invariance. *Computer Science* (2014).

[45] X. Wang and J. Schneider. 2014. Flexible transfer learning under support and model shift. *NIPS* 3 (2014), 1898–1906.

[46] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *Computer Science* (2015), 2048–2057.

[47] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked Attention Networks for Image Question Answering. In *CVPR*. 21–29.

[48] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? *NIPS* 27 (2014), 3320–3328.

[49] Sergey Zagoruyko and Nikos Komodakis. 2017. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In *ICLR*.

[50] Matthew D. Zeiler and Rob Fergus. 2013. Visualizing and Understanding Convolutional Networks. In *ECCV*, Vol. 8689. 818–833.

[51] Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, and Rob Fergus. 2010. Deconvolutional networks. In *CVPR*. 2528–2535.