Deep Cocktail Network: Multi-source Unsupervised Domain Adaptation with Category Shift

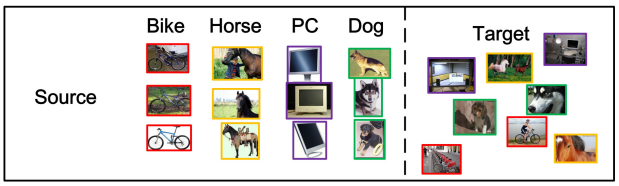
Ruijia Xu[1,†], Ziliang Chen[1,†], Wangmeng Zuo[2], Junjie Yan[3], Liang Lin[1,3*]
[1]Sun Yat-sen University [2]Harbin Institute of Technology [3]SenseTime Research
xurj3@mail2.sysu.edu.cn, c.ziliang@yahoo.com, wmzuo@hit.edu.cn,
yanjunjie@sensetime.com, linliang@ieee.org

Abstract

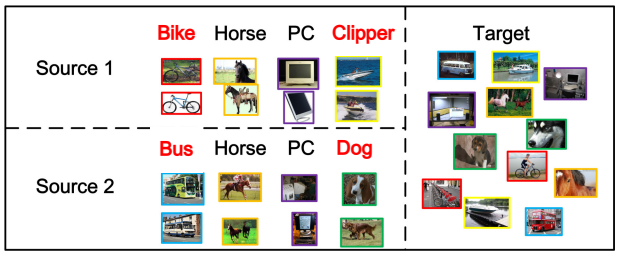
Unsupervised domain adaptation (UDA) conventionally assumes labeled source samples coming from a single underlying source distribution. Whereas in practical scenario, labeled data are typically collected from diverse sources. The multiple sources are different not only from the target but also from each other, thus, domain adaptater should not be modeled in the same way. Moreover, those sources may not completely share their categories, which further brings a new transfer challenge called category shift. In this paper, we propose a deep cocktail network (DCTN) to battle the domain and category shifts among multiple sources. Motivated by the theoretical results in [33], the target distribution can be represented as the weighted combination of source distributions, and, the multi-source UDA via DCTN is then performed as two alternating steps: i) It deploys multi-way adversarial learning to minimize the discrepancy between the target and each of the multiple source domains, which also obtains the source-specific perplexity scores to denote the possibilities that a target sample belongs to different source domains. ii) The multi-source category classifiers are integrated with the perplexity scores to classify target sample, and the pseudo-labeled target samples together with source samples are utilized to update the multi-source category classifier and the feature extractor. We evaluate DCTN in three domain adaptation benchmarks, which clearly demonstrate the superiority of our framework.

1. Introduction

Recent advances in deep learning have significantly improved the state-of-the-arts across a variety of visual learning tasks [25] [40] [28] [6] [47]. These achievements, to a great extent, should be attributed to the availability of large scale labeled data for supervised learning. When turning to (Unsupervised) domain adaptation (UDA, see Fig. 1(a)) [38] [37] [15], we do not have the labels of the data in target domain, but have massive labeled data in source domain. One natural solution is to learn a deep model on the labeled source data and deploy it to target domain. However, due to the presence of *domain shift* [18], the performance of the learned model tends to degrade heavily in the target domain. To mitigate the model damage caused by the domain shift, UDA learns to map the data from both domains into a common feature space by minimizing domain distribution discrepancy, the source classifier can then be directly ap-



(a) Single source domain adaptation



(b) Multi-source domain adaptation with category shift

Figure 1. (a). Single source domain adaptation (UDA) assumes that source samples are drawn from some underlying distribution under the i.i.d. condition. (b). Multiple source domain adaptation (MDA) assume source data are collected from different source distributions. Category shift is a new protocol in MDA, where domain shift and categorical disalignment co-exist among the sources.

*Corresponding author: Liang Lin. † indicates equal contribution. This work was supported by National Science Foundation of China under Grant U1611461 and Special Funds for Guangdong College Students' Science and Technology Innovation (under Grant pdjhb0009).

plied to target instances. While early UDA studies mainly focus on shallow models [37] [15], with the dramatic upsurge of convolutional neural networks (CNNs), deep UDA has emerged as a thriving solution and has achieved many stunning results [20] [4] [12].

However, most existing deep UDA methods assume that there is only a single source domain and the labeled source data are implicitly sampled from a same underlying distribution. In practice, it is very likely that we have multiple source domains. For example, when training object recognition models for household robots, one can exploit the labeled images either from Amazon.com (Source 1) or Flickr (Source 2). Moreover, the large scale dataset, e.g., *ImageNet* [7] may be built upon diverse sources from the Internet, and is inappropriate to be treated as a single domain in UDA. Consequently, multiple source unsupervised domain adaptation (MDA) is both feasible in practice and more valuable in performance, and has received considerable attention in application fields [49][8][22] [27].

Despite the rapid progress in deep UDA, seldom studies have been given to deep MDA which is much more challenging due to the following reasons. Firstly, with possible domain shifts among sources, it's improper to apply single source UDA via combining all source domains. Secondly, different source domains convey complimentary information to target domain. Based on Liebig's law of the minimum, it is too strict to eliminate the distribution discrepancy between target domain and each source domain, and may be harmful to the model performance. Finally, as illustrated in Fig.1(b), different source domains may not completely share their categories (i.e., *category shift*), some category of samples may appear in one source domain but not in another. MDA should take both category shift and domain shift into account, and is thus more challenging to handle.

In this paper, we propose the *deep cocktail network* (DCTN) for MDA. Inspired by the *distribution weighted combining rule* in [33], the target distribution can be represented as the weighted combination of the multi-source distributions. Suppose the classifier for each source domain is known. An ideal target predictor can be obtained by integrating all source predictions based on the corresponding source distribution weights. Therefore, besides of the feature extractor, DCTN also includes a (multi-source) category classifier to predict the class from different sources, and a (multi-source) domain discriminator to produce multiple source-target-specific perplexity scores as the approximation of source distribution weights. Analogous to make cocktails, the multi-source class predictions are integrated with the perplexity scores to classify the target sample, and thus the proposed method is dubbed by deep cocktail network (DCTN).

During training, the learning algorithm for DCTN performs the following two alternating adaptation steps: (i) the domain discriminator is updated by using multi-way adversarial learning to minimize the domain discrepancies between target and each source, then to predict multi-source perplexity scores; (ii) the feature extractor and the category classifier are discriminatively fine-tuned with multi-source labeled and target pseudo-labeled data. The multi-way adversarial adaptation implicitly reduces domain shifts among those sources. The discriminative adaptation helps to learn more classifiable features [42], and partially prevents the *negative transfer* [38] from the mis-matching categories. Empirical studies on three domain adaptation benchmarks also demonstrate the effectiveness of our DCTN framework.

Our work contributes in the three aspects: **1)** We present a novel and realistic MDA protocol termed *category shift* that relaxes the requirement on the shared category set among any source domains. **2)** Inspired from the distribution weighted combining rule, we proposed the *deep cocktail network* (DCTN) together with the alternating adaptation algorithm to learn transferable and discriminative representation. **3)** We conduct comprehensive experiments on three well-known benchmarks, and testify our model in both the vanilla and the *category shift* settings. Our method has achieved the state of the art across most transfer tasks.

2. Related Work

Unsupervised domain adaptation with single source. Provided a source domain with ground truth and target domain without labels, unsupervised domain adaptation (UDA) aims at learning a model well-performing on target distribution. Since the source and the target belong to different distributions, the technical problem in UDA is how to reduce the domain shift across the source and the target. Inspired by the two-sample test [17], domain discrepancy based methods, e.g., shallow-model-based TCA [37], JDA [1]; deep-model-based DAN [29], WMMD [48], RTN [30], leverage different distribution measures as domain regularizer to attain domain-invariant feature. Adversarial learning behaves effective to learn more transferable representations. It defines a couple of networks and trains them in the opposite direction: a domain discriminator minimizes the classification error to distinguish samples from source and target, while domain mapping learns transferable representations indistinguishable by the domain discriminator. Recent relevant researches perform superior in visual recognition cross domain [30] [12] and task [34] and transfer structure learning [4] [21]. Besides of these two mainstreams, there are diverse methods to learn domain-invariant features: semi-supervised method [42], domain reconstruction [14], duality [19], alignments [9] [50] [44], manifold learning [15], tensor methods [24][31], etc.

Domain adaptation with multiple sources. The UDA methods mentioned above mainly consider target *vs.* single source. If multiple sources are available, the domain shift

among sources should also be account for. The research originates from A-SVM [49] that leverages the ensemble of source-specific classifiers to tune the target categorization model, and there have been a variety of shallow models invented to tackle the MDA problem [8] [22] [27]. MDA also develops with theoretical supports [3] [2] [33]. Blitzer et al [3] provides the first learning bound for MDA. Mansour et al [33] claims that an ideal target hypothesis can be represented by a distribution weighted combination of source hypotheses. This methodology termed *distribution weighted combining rule*, closely means that, if the relations between target and each source can be discovered, we are able to use multiple source-specific classifiers to obtain an ideal target class prediction.

Continual transfer learning, domain generalization. There are two branches of transfer learning closely relate to MDA. The first is continual transfer learning (CTL) [43] [39]. Similar to continual learning [23], CTLs train the learner to sequentially master multiple tasks across multiple domains. The second is domain generalization (DG) [13] [35], which solely uses the existing multiple labeled domains for training regardless of the unlabeled target samples. Both of the problems are solved by supervised learning approaches, and distinguished from MDA with unlabeled training samples.

3. Problem Setup

Vanilla MDA. In the context of multi-source transfer, there are N different underlying source distributions denoted as $\{p_{s_j}(x,y)\}_{j=1}^{N}$. The labeled source domain images $\{(X_{s_j},Y_{s_j})\}_{j=1}^{N}$ are drawn from those distributions respectively, where $X_{s_j}=\{x_i^{s_j}\}_{i=1}^{|X_{s_j}|}$ represents images from source j and $Y_{s_j}=\{y_i^{s_j}\}_{i=1}^{|Y_{s_j}|}$ is the corresponding ground-truth set. Besides, we have target distribution $p_t(x,y)$, from which target image set $X_t=\{x_i^t\}_{i=1}^{|X_t|}$ are sampled yet without label observation Y_t. Those N+1 datasets have been treated as an training set ensemble, and the test set $(X_{test},Y_{test})=\{x_i^{test},y_i^{text}\}_{i=1}^{|X_{test}|}$ are drawn in target distribution to evaluate the model adaptation performance.

Category Shift. Under the vanilla MDA setting, samples from diverse sources share a same category set. In contrast to this old fashion, we introduce a new MDA protocol where the categories from different sources might be also different. Formally speaking, given a category set $\mathcal{C}_s=\bigcup_{i=1}^{|Y_s|}\{y_i^s\}$ as a class set of Y_s for domain s, the relation between $\mathcal{C}_{s_{j_1}}$ and $\mathcal{C}_{s_{j_2}}$ has been generalized from $\mathcal{C}_{s_{j_1}}\cup\mathcal{C}_{s_{j_2}}=\mathcal{C}_{s_{j_1}}\cap\mathcal{C}_{s_{j_2}}$ to $\mathcal{C}_{s_{j_1}}\cap\mathcal{C}_{s_{j_2}}\subseteq\mathcal{C}_{s_{j_1}}\cup\mathcal{C}_{s_{j_2}}$, where $\mathcal{C}_{s_{j_1}}\cap\mathcal{C}_{s_{j_2}}$ denotes public classes between sources j_1 and j_2. Let target domain get labeled by the union of all categories in those sources ($\mathcal{C}_t=\bigcup_{j=1}^{M}\mathcal{C}_{s_j}$), then we term $\mathcal{C}_{s_{j_1}}\cap\mathcal{C}_{s_{j_2}}\neq\mathcal{C}_{s_{j_1}}\cup\mathcal{C}_{s_{j_2}}$ as *category shift* in multiple source domains $\{(X_{s_j},Y_{s_j})\}_{j=1}^{N}$.

Compared with Open Set DA. Open set domain adaptation (DA) [5] is a new single-source transfer protocol, where the classes between the source and the target domains are allowed to be different. The uncommon classes are unified as a negative category called "unknown". In contrast, category shift consider the specific disaligned categories among multiple sources to enrich the classification in transfer. In fact, the open set DA can also be developed to our category shift setting, where the unshared classes are viewed unobservable. Such study will be investigated in our future work.

4. Deep Cocktail Network

Irrespective of either vanilla or category shift scenarios, MDAs are challenging to tackle. In this section, we introduce *deep cocktail network* (DCTN), an adversarial domain adaptation framework for both MDA protocols. It connects to the *distribution weighted combining rule* [33], and what's more, can be easily transplanted to suit the shifted categories without model reconfiguration.

4.1. Architecture

Our framework consists of four components: three subnets, i.e., *feature extractor*, *(multi-source) domain discriminator*, *(multi-source) category classifier*, and a non-learnable *target classification operator*, as shown in Fig.2.

Feature extractor F incorporates deep convolution nets as the backbone, and is supposed to map all images from N sources and target into a common feature space. We employ adversarial learning to obtain the optimal mapping, because it can successfully learn both domain-invariant features and each target-source-specific relations.

(Multi-source) domain discriminator D is built upon N source-specific discriminators $\{D_{s_j}\}_{j=1}^{N}$ for adversary. Given image x from the source j or the target domain, the domain discriminator D receives the features $F(x)$, then the source-specific discriminator D_{s_j} classifies whether $F(x)$ originates from the source j or the target. The data flow from source j doesn't trigger other source discriminators, yet for the data flow from each target instance x^t, the domain discriminator D yields the N source-specific discriminative results $\{D_{s_j}(F(x^t))\}_{j=1}^{N}$. They are used to update the domain discriminator D, also to supply the target-source perplexity scores $\{\mathcal{S}_{cf}(x^t;F,D_{s_j})\}_{j=1}^{N}$ to the target classification operator

$$\mathcal{S}_{cf}(x^t;F,D_{s_j})=-\log(1-D_{s_j}(F(x^t)))+\alpha_{s_j} \qquad (1)$$

where α_{s_j} is the *source-specific concentration constant*. It is obtained by averaging the source j discriminator losses over X_{s_j}.

(Multi-source) category classifier C is a multi-output net composed by N source-specific predictors $\{C_{s_j}\}_{j=1}^{N}$. Each predictor C_{s_j} is a softmax classifier configured by the

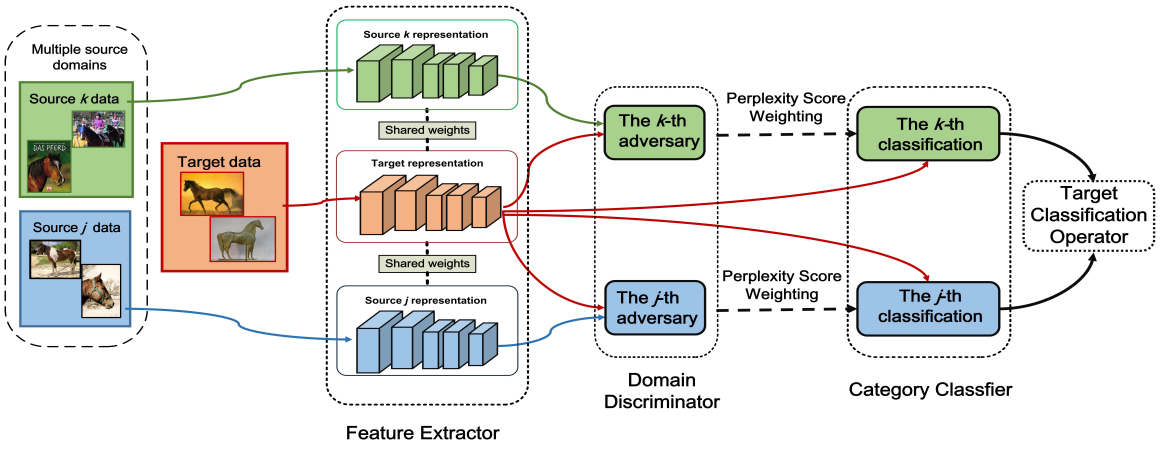


Figure 2. An overview of the proposed Deep Cocktail Network (DCTN). Our framework receives multi-source instances with annotated ground truth and adapts to classify the target samples. Let's consider the source j and k for simplicity. **i)** The feature extractor maps target, source j and k into a common feature space. **ii)** The category classifier receives target feature and produces the j-th and k-th classifications based upon the categories in source j and k respectively. **iii)** The domain discriminator receives features from source j, k and target, then offers the k-th advesary between target and source k, as well as the j-th advesary between target and source j. The j-th and k-th advesary provide source j and k perplexity scores to weight the j-th and k-th classifications correspondingly. **iv)** The target classification operator integrates all weighted classification results then predicts the target class across category shifts. Best viewed in color.

category set in the corresponding source j. The category classifier takes an image mapping as input, then for the image from source j, only the value from C_{s_j} get activated and provides the gradient for training. For a target image x^t instead, all source-specific predictors provide N categorization results $\{C_{s_j}(F(x^t))\}_{j=1}^N$ to the target classification operator.

Target classification operator is the key to classify target samples. In specific, for each target feature $F(x^t)$, the target classification operator takes each source perplexity score $\mathcal{S}_{cf}(x^t; F, D_{s_j})$ to re-weight the corresponding source-specific prediction $C_{s_j}(F(x^t))$, then accumulates the results to classify target x^t. If the class $c \in \bigcup_{j=1}^{N} \{\mathcal{C}_{s_j}\}$ is considered, the confidence x^t belongs to c presents as

$$Confidence(c|x^t) := \sum_{c \in \mathcal{C}_{s_j}} \frac{\mathcal{S}_{cf}(x^t; F, D_{s_j})}{\sum_{c \in \mathcal{C}_{s_k}} \mathcal{S}_{cf}(x^t; F, D_{s_k})} C_{s_j}(c|F(x^t)) \tag{2}$$

where $C_{s_j}(c|F(x^t))$ denotes the softmax value of source j corresponding to class c. x^t is categorized into the class with the highest confidence. The sum $\sum_{c \in \mathcal{C}_{s_j}}$ means only those sources with class c can join the perplexity score weighting. It's invented to incorporate both the vanilla and the *category shift* settings. Since the module independently estimates each class confidence, the variation in shifting categories merely modifies the class combination in the target classification operator, but not the structures or the parameters in the three subnets.

Connection to distribution weighted combining rule. Let $\{\mathcal{D}_{s_j}\}_{j=1}^N$ and $\mathcal{D}_t$ denote sources and target distributions[1], and given an instance x, $\{\mathcal{D}_{s_j}(x)\}_{j=1}^N$ and $\mathcal{D}_t(x)$ denote the probabilities that x is generated from $\{\mathcal{D}_{s_j}\}_{j=1}^N$ and $\mathcal{D}_t$, respectively. In the *distribution weighted combining rule* [33], the target distribution is treated as a mixture of the multi-source distributions with the coeffients by normalized source distributions weighted by unknown positive $\{\lambda_j\}_{j=1}^N$, namely $\mathcal{D}_t(x) = \sum_{c \in \mathcal{C}_{s_k}}^{N} \lambda_k \mathcal{D}_{s_k}(x)$. The ideal target classifier $C_t(c|x^t)$ presents as the weighted combination of source classifiers $\{C_{s_j}(c|F(x^t))\}_{j=1}^M$:

$$C_t(c|x^t) = \sum_{c \in \mathcal{C}_{s_j}} \frac{\lambda_j \mathcal{D}_{s_j}(x^t)}{\sum_{c \in \mathcal{C}_{s_k}} \lambda_k \mathcal{D}_{s_k}(x^t)} C_{s_j}(c|F(x^t)) \tag{3}$$

Note a fact that, with the increase of the probability that x^t from source j, x^t becomes similar to the sample from source j. It holds $D_{s_j}(F(x^t)) \to 1$ and results in $-\log(1 - D_{s_j}(F(x^t)))$ increasing. Hence it maintains $\lambda_j \mathcal{D}_{s_j}(x^t) \propto \mathcal{S}_{cf}(x^t; F, D_{s_j})$ in the multiple source domains. Replace the source distributions with the normalized source perplexity scores, then $C_t(c|x^t)$ corresponds to the target classification operator in Eq.2. The formula physically implies that target images should be categorized by the classifiers from multiple sources, with whose features more similar to target, the source classifiers' prediction are more trustful.

[1]Since each sample x corresponds to an unique class y, $\{\mathcal{D}_{s_j}\}_{j=1}^N$ and $\mathcal{D}_t$ can be viewed as an equivalent embedding from $\{p_{s_j}(x, y)\}_{j=1}^N$ and $p_t(x, y)$ that we have discussed.

4.2. Learning

Our framework admits an alternative adaptation pipeline. Briefly, after a proper pre-training, DCTN employs a multi-way adversary to acquire a mutual mapping from all domains, then further, the feature extractor and the category classifier are trained with multiple sources labeled and target pseudo-labeled images. The two stages repeat until the maximal epoch is reached.

Pre-training Pre-trained feature extractor and category classifier are the prerequisites for the alternative process. At the very start, we take all source images to jointly train the feature extractor F and the category classifier C. Those networks and the target classification operator then predict categories for all target images[2] and annotate those with high confidences. Finally, we obtain the pre-trained feature extractor and category classifier via further fine-tuning them with sources and the pseudo-labeled target images. The alternative paradigm begins after this pretraining.

4.2.1 Multi-way Adversarial Adaptation

Our first stage multi-source domain adaptation are now described as follow:

$$\min_{F} \max_{D} V(F, D; \overline{C}) = \mathcal{L}_{adv}(F, D) + \mathcal{L}_{cls}(F, \overline{C}) \tag{4}$$

where

$$\begin{aligned} \mathcal{L}_{adv}(F, D) = \frac{1}{N} \sum_{j}^{N} \mathbb{E}_{x \sim X_{s_j}} [\log D_{s_j}(F(x))] \\ + \mathbb{E}_{x^t \sim X_t} [\log(1 - D_{s_j}(F(x^t)))] \end{aligned} \tag{5}$$

where the first term denotes our adversarial mechanism, and the second term is a multi-source classification losses. The classifier C is fixed as $\overline{C}$ to provide stable gradient values.

The optimization based on Eq.4 works well for D but not F. Since the feature extractor learns the mapping from the multiple sources and the target, the domain distributions become simultaneously changing in adversary, which results in an oscillation then spoils our feature extractor. Towards such concern, Tzeng et al.[45] mentioned when source and target feature mappings share their architectures, the domain confusion can be introduced to replace the adversarial objective, which performs stable to learn the mapping F. Extend it to our scenario, we have the following multi-domain confusion loss:

$$\begin{aligned} \mathcal{L}_{adv}(F, D) = \frac{1}{N} \sum_{j}^{N} \mathbb{E}_{x \sim X_{s_j}} \mathcal{L}_{cf}(x; F, D_{s_j}) \\ + \mathbb{E}_{x \sim X_t} \mathcal{L}_{cf}(x^t; F, D_{s_j}) \end{aligned} \tag{6}$$

[2]Since the domain discriminator hasn't been trained, we take the uniform distribution simplex weight as the perplexity scores to the target classification operator.

Algorithm 1 Mini-batch Learning via online hard domain batch mining

Input: Mini-batch $\{x_i^t, \{x_i^{s_j}, y_i^{s_j}\}_{j=1}^N\}_{i=1}^M$ sampled from X_t and $\{(X_{s_j}, Y_{s_j})\}_{j=1}^N$ respectively; feature extractor F; domain discriminator D; category classifier $\overline{C}$.
Output: Updated F'.

1: Select the source domain $j^* \in [N]$, where $j^* = \arg\max_j^N \{\sum_i^M - \log D_{s_j}(F(x_i^{s_j})) - \log(1 - D_{s_j}(F(x_i^t)))\}_{j=1}^N$;
2: $\mathcal{L}_{adv}^{s_{j^*}} = \sum_i^M \mathcal{L}_{cf}(x_i^{s_{j^*}}; F, D_{s_{j^*}}) + \mathcal{L}_{cf}(x_i^t; F, D_{s_{j^*}})$
3: Replace $\mathcal{L}_{adv}$ in Eq.4 with $\mathcal{L}_{adv}^{s_{j^*}}$, update F by Eq.4.
4: **return** $F' = F$.

where

$$\begin{aligned} \mathcal{L}_{cf}(x; F, D_{s_j}) = \\ \frac{1}{2} \log D_{s_j}(F(x)) + \frac{1}{2} \log(1 - D_{s_j}(F(x))) \end{aligned} \tag{7}$$

Online hard domain batch mining In the stochastic gradient manner, the multi-way adversarial learning receive N samples from N sources respectively to update F in each iteration. However, the samples from different sources are sometimes useless to improve the adaptation to the target, and as the training proceeds, more redundant source samples turn to draw back the whole model performance. To mitigate this negative effect, we proposed a simple yet effective multi-source batch mining technique to improve the training. For a specific target batch $\{x_i^t\}_{i=1}^M$, we consider N sources batches $\{\{x_i^{s_1}\}_{i=1}^M, \cdots, \{x_i^{s_N}\}_{i=1}^M\}$. Each source-target discriminator loss $\{\sum_i^M - \log D_{s_j}(F(x_i^{s_j})) - \log(1 - D_{s_j}(F(x_i^t)))\}_{j=1}^N$, is viewed as the degrees to distinguish x_i^t from N source samples. Hence F performs worst to transform the target samples to confuse source j^*, which results in $j^* = \arg\max_j^N \{\sum_i^M - \log D_{s_j}(F(x_i^{s_j})) - \log(1 - D_{s_j}(F(x_i^t)))\}_{j=1}^N$. Based upon the domain confusion loss, we use the source j^* and the target samples in the mini-batch to train the feature extractor. This stochastic learning method is represented by the Algorithm.1.

4.2.2 Target Discriminative Adaptation

Aided by the multi-way adversary, DCTN has been able to obtain good domain-invariant features, yet not surely classifiable in the target domain. David et al [2] demonstrates that, to apply source classifier in the target domain, it must acquiesces in a classifier that works well on both the domains. However, in the MDA setting, such ideal across-domain classifier must account for the non-consistency among different sources, even with their shifting categories. It's obvious that such MDA-based classifier is too difficult to access.

To further approach an ideal target classifier, we directly incorporate target samples to learn discriminative features with multiple sources. We propose an auto-labeling strategy to annotate target samples, then jointly train our feature extractor and multi-source category classifier with source and target images by their (pseudo-) labels. Hence, the discriminative adaptation of DCTN presents as

$$\min_{F,\,C}\ \mathcal{L}_{cls}(F,C)=\sum_{j}^{N}\mathbb{E}_{(x,y)\sim(X_{s_j},Y_{s_j})}[\mathcal{L}(C_{s_j}(F(x)),y)] + \mathbb{E}_{(x^t,\hat{y})\sim(X_t^p,Y_t^p)}[\sum_{\hat{y}\in\mathcal{C}_{\hat{s}}}\mathcal{L}(C_{\hat{s}}(F(x^t)),\hat{y})] \tag{8}$$

where the first and second terms denote the classification losses from multiple source images $\{X_{s_j},Y_{s_j}\}_{j=1}^{N}$, and target images with pseudo labels $\{X_t^P,Y_t^P\}$ respectively. We apply the target classification operator to assign pseudo labels, and the samples with the confidence higher than a preseted threshold γ will be selected into X_t^P.

Since the target predictions come from the integration of multi-source predictions, there is no explicit learnable target classifier. As illustrated in the second term of Eq.8, we apply the multi-source category classifier to back-propagate pseudo target classification errors. Concretely, given a target instance x^t with pseudo-labeled class $\hat{y}$, we find those sources $\hat{s}$ include this class ($\hat{y}\in\mathcal{C}_{\hat{s}}$), then update our network via the sum of the multi-source classification losses, namely, $\sum_{\hat{y}\in\mathcal{C}_{\hat{s}}}\mathcal{L}(C_{\hat{s}}(F(x^t)),\hat{y})$ in the second term.

The alternative adaptation pipline of DCTN has been summarized in Algorithm.2.

Algorithm 2 Learning algorithm for DCTN

Input: N source labeled datasets $\{X_{s_j},Y_{s_j}\}_{j=1}^{N}$; target unlabeled dataset X_t; initiated feature extractor F; category classifier C and domain discriminator D; confidence threshold γ; adversarial iteration threshold β.
Output: well-trained feature extractor F^*, domain discriminator D^* and category classifier C^*.

Pre-train C **and** F
while not converged **do**
  Multi-way Adversarial Adaptation:
  for 1:β **do**
    Sample mini-batch from $\{X_{s_j}\}_{j=1}^{N}$ and X_t;
    Update D by Eq.4;
    Update F by Algorithm.1;sequentially
  end for
  Target Discriminative Adaptation:
  Estimate confidence for X_t by Eq.2 with perplexity scores offered by Eq.1. Samples $X_t^P\subset X_t$ with confidence larger than γ get annotations Y_T^P;
  Update F and C by Eq.8.
end while
return $F^*=F; C^*=C; D^*=D$.

5. Experiments

In the context of MDA for visual classification, we evaluate the accuracy of the predictions from the target classification operator in all experiments, and both of the vanilla setting and the category shift have been validated. Our DCTN are all implemented in the PyTorch[3] platform. We report the major results in the paper, and more implementation information and results have been detailed in the Appendix.

5.1. Benchmarks

Three widely used UDA benchmarks *Office-31* [41], *ImageCLEF-DA*[4] and *Digits-five* have been introduced for the MDA experimental evaluation. *Office-31* is a object recognition benchmark with 31 categories and 4652 images unevenly spread in three visual domains **A** (*Amazon*), **D** (*DSLR*), **W** (*Webcam*). *ImageCLEF-DA* derives from ImageCLEF 2014 domain adaptation challenge, and is organized by selecting 12 object categories (aeroplane, bike, bird, boat, bottle, bus, car, dog, horse, monitor, motorbike, and people) shared in the three famous real-world datasets, **I** (*ImageNet ILSVRC 2012*), **P** (*Pascal VOC* 2012), **C** (*Caltech-256*). It includes 50 images in each category and totally 600 images for each domain. *Digits-five* includes five digit image sets respectively sampled from following public datasets, **mt** (*MNIST*) [26], **mm** (*MNIST-M*) [11], **sv**(*SVHN*) [36], **up** (*USPS*) and **sy** (*Synthetic Digits*) [11]. Towards the images in *MNIST*, *MNIST-M*, *SVHN* and *Synthetic Digits*, we draw 25000 for training and 9000 for testing in each dataset. There are only 9298 images in *USPS*, so we choose the entire dataset as our domain.

5.2. Evaluations in the vanilla setting

Baselines. The existing works of MDA lack comprehensive evaluations on real-world visual recognition benchmarks. In our experiment, we introduce two shallow methods, sparse FRAME (**sFRAME**) [46] and **SGF** [16] as the multi-source baselines in the *Office-31* experiment. Besides, we evaluate DCTN with single-source visual UDA methods including the conventional, e.g., Transfer Component Analysis (**TCA**) [37] and Geodesic Flow Kernel (**GFK**) [15], as well as state-of-the-art deep methods: Deep Domain Confusion (**DDC**) [20], Deep Reconstruction-classification Networks (**DRCN**) [14], Reversed Gradient (**RevGrad**) [10], Domain Adaptation Network (**DAN**) [29], and Residual Transfer Network (**RTN**) [30]. Since those methods perform in single-source setting, we introduce two MDA standards for different purposes: 1). *Source combine*: all source domains are combined into a traditional

[3]http://pytorch.org/
[4]http://imageclef.org/2014/adaptation

Table 1. Classification accuracy (%) on Office-31 dataset for MDA in the vanilla setting.

Standards	Models	A,W→D	A,D→W	D,W→A	Avg
Single best	TCA	95.2	93.2	51.6	68.8
	GFK	95.0	95.6	52.4	68.7
	DDC	98.5	95.0	52.2	70.7
	DRCN	99.0	96.4	**56.0**	73.6
	RevGrad	99.2	96.4	53.4	74.3
	DAN	99.0	96.0	54.0	72.9
	RTN	**99.6**	96.8	51.0	73.7
Source combine	Source only	98.1	93.2	50.2	80.5
	RevGrad	98.8	96.2	54.6	83.2
	DAN	98.8	95.2	53.4	82.5
Multi-source	Source only	98.2	92.7	51.6	80.8
	sFRAME	54.5	52.2	32.1	46.3
	SGF	39.0	52.0	28.0	39.7
	DCTN (ours)	**99.6**	**96.9**	54.9	**83.8**

Table 2. Classification accuracy (%) on ImageCLEF-DA dataset for MDA in the vanilla setting.

Standards	Models	I,C→P	I,P→C	P,C→I	Avg
Single best	RevGrad	66.5	89.0	81.8	78.2
	DAN	67.3	88.4	80.5	76.9
	RTN	67.4	89.5	81.3	78.4
Source combine	Source only	68.3	88.0	81.2	79.2
	RevGrad	67.0	90.7	81.8	79.8
	DAN	**68.8**	88.8	81.3	79.6
Multi-source	Source only	68.5	89.3	81.3	79.7
	DCTN (ours)	**68.8**	**90.0**	**83.5**	**80.8**

Table 3. Classification accuracy (%) on Digits-five dataset for MDA in the vanilla setting.

Standards	Models	mm, mt, sy, up → sv	mt, sy, up, sv → mm	Avg
Source combine	Source only	72.2	64.1	68.2
	RevGrad	68.9	**71.6**	70.3
	DAN	71.0	66.6	68.8
Multi-source	Source only	64.6	60.7	62.7
	RevGrad	61.4	71.1	66.3
	DAN	62.9	62.6	62.8
	DCTN (ours)	**77.5**	70.9	**74.2**

single-source *v.s.* target setting. 2). *Single best*: in the muli-source domains, we report the single source transfer result best-performing in the test set. The first standard testify whether the multi-source is valuable to exploit; the second evaluates whether we can further improve the best single source UDA via introducing another source transfer. Additionaly, as baselines in the *Source combine* and multi-source standards, we use all images from sources to train backbone-based multi-source classifiers and directly apply them to classify target images. They are termed *Source only* and used to confirm whether our multi-source transfers are available. For a fair comparison, all deep model baselines in *Office-31* and *ImageCLEF-DA* use the Alexnet architectures, and share the same backbone model in *Digits-five*.

In the object recognition, we report all combinations of domain shifts and compare DCTN with the baselines. Tables.1-2 show that DCTN yields the best results in the *Office-31* transfer tasks **A**,**W**→**D** and **A**,**D**→**W**, performs compelling in **D**,**W**→ **A** and outperforms conventional MDA baselines by large margins. In the *ImageCLEF-DA*, DCTN attains the state of the art in all transfer tasks. These validate that, no matter domain size is equal or not, DCTN can learn more transferable and discriminative features from multi-source transfer.

In the digit recognition, there are four source domains and we convey the results in the domain shifts as **mm, mt, sy, up** → **sv** and **mt, sv, sy, up** → **mm**. We compare them with DAN under the source-combine and the multi-source average accuracy of its four single source transfer combinations. The results have been shown in Table.3. Despite of involving multiple source domain shifts, DCTN still can improve the source combine performance by 6.0%.

Table 4. Evaluations on Office-31 (A,D→ W) for MDA in the category shift setting.

Category Shift	Models	Accuracy	Degraded Accuracy	Transfer Gain
Overlap	Source only	84.4	-8.3	0
	RevGrad	86.3	-7.9	1.9
	DAN	87.8	**-6.4**	3.4
	DCTN(ours)	90.2	-6.7	**5.8**
Disjoint	Source only	78.1	-14.6	0
	RevGrad	78.6	-15.6	0.5
	DAN	75.5	-18.7	-2.6
	DCTN(ours)	82.9	**-14.0**	**4.8**

Table 5. Evaluations on ImageCLEF-DA (I,P→ C) for MDA in the category shift settings.

Category Shift	Models	Accuracy	Degraded Accuracy	Transfer Gain
Overlap	Source only	86.3	-3.0	0
	RevGrad	85.7	-4.5	-0.6
	DAN	85.5	-4.0	-0.8
	DCTN(ours)	88.7	**-1.3**	**2.4**
Disjoint	Source only	81.5	**-7.8**	0
	RevGrad	71.5	-18.7	-10.0
	DAN	71.0	-18.5	-10.5
	DCTN(ours)	82.0	-8.0	**0.5**

5.3. Evaluations in the category shift setting

How to evaluate? Since category shift is a brand new MDA protocol, in order to evaluate the model in this protocol, the multiple sources are amended to satisfy categorical disalignments. We consider the two-source adaptation in object recognition. In the category order of the benchmarks, we take the first and the last one third classes as the private classes of both source domains respectively, and the rest are the public classes shared in both sources. This organization in category shift is termed *Overlap*. In the same order, we depart all categories into two non-overlapped class sets and define them as the private classes. Since no classes are common, we named it as *Disjoint*. We testify DCTN on both the source domain organizations, and compare the results with Source only, RevGrad and DAN. The accuracy degradation compared to the performance in the vanilla setting and the transfer gain compared to *Source only* are also appended.

The evaluations have been shown in Table.4-5. Category

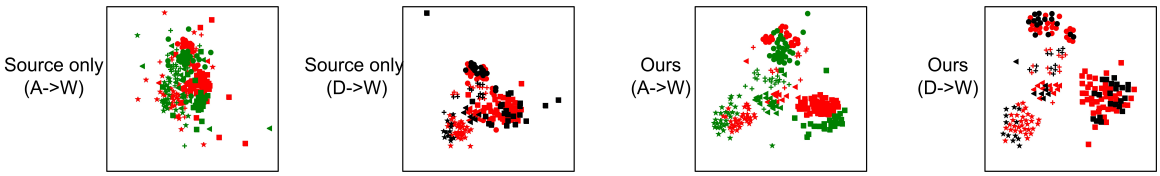


Figure 3. The t-SNE [32] visulization of A,D→W. Green, black and red represent domains A, D and W respectively. We use different markers to denote 5 categories, e.g., bookcase, calculator, monitor, printer, ruler. Best viewed in color.

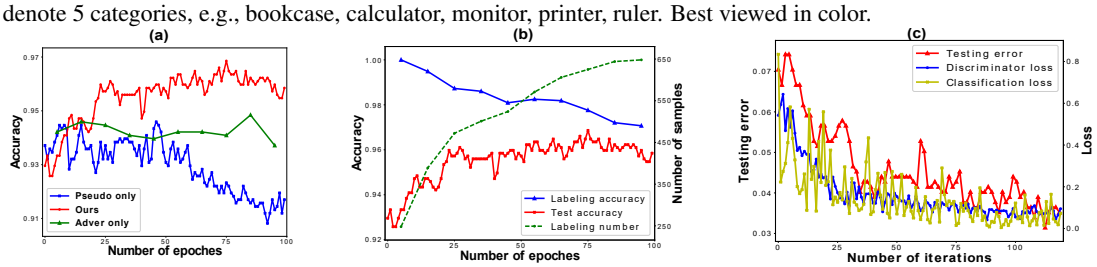


Figure 4. Analysis:(a) the accuracies of DCTN, adversarial-only and pseudo-only models; (b) the accuracies of testing samples and pseudo-labeled target samples; (c) the convergence performance on different losses. Best viewed in color.

shift is very challenging, and under the *Overlap*, the accuracies of DAN got slashed by -6.4 in the *Office-31* and -4.0 in the *ImageCLEF-DA*. The performance deteriorate to -18.7 and -18.5 under the *Disjoint*. Moreover, DAN also suffers negative transfer gains in most situations, which indicates the transferbility of DAN cripled in the category shift. In contrast, DCTN reduces the performance drops compared to the model in the vanilla setting, and obtains positive transfer gains in all situations. It reveals that DCTN can resist the negative transfer caused by the category shift.

5.4. Further Analysis

Feature visualization. In the experiment of adaptation task **A,D** $\rightarrow$ **W** in Office-31, we visualize the DCTN activations before and after adaptation. For simplicity, both the source domains have been separated to emphasize the contrast of target. As we can see in Fig.3, compared with the activations given by the source only, both of the activations from **A** $\rightarrow$ **W** and **D** $\rightarrow$ **W** have shown good adaptation patterns. It means DCTN can successfully learn transferable features with multiple sources. Besides, the target activations become more clear to categorize, which suggests that the features learned by DCTN attains desirable discriminative property. Finally, even if the multi-source transfer has been composed of hard transfer task (**A** $\rightarrow$ **W**), DCTN is still able to adapt to target domain without the degradation in the performance of **D** $\rightarrow$ **W**.

Ablation study. DCTN contains two major parts: the multi-way adversary and the auto-labeling scheme. To further reveal their function, we decompose DCTN into two variants: The **adversarial-only** model excludes the pseudo-labels and updates the category classifier with source samples. The **pseudo-only** model forbids the adversary and categorize target samples with average multi-source results. As shown in Fig.4(a), the accuracy of adversary behaves stable in each iteration, but lack of target class guidance, its final performance hits a bottleneck. But without the adversary, the accuracy of pseudo labels significantly drops and pulls down the DCTN accuracy. It indicates that the both adaptations cooperate with each other to achieve desirable transfer behaviors. Diving deeper in Fig.4(b), the test accuracy and the pseudo label accuracy show converged in the alternative learning, which implicitly reveals the consistency between the both adaptations. We also provide the ablative study result to the domain batch mining technique (see Table.6), which testify the method's efficacy.

Table 6. Ablation study of Algorithm.1 in Office-31.

	A,W→D	A,D→W	D,W→A	Overlap	Disjoint
w	**99.6**	**96.9**	54.9	**90.2**	**82.9**
w/o	99.0	96.1	**55.0**	89.3	82.6

Convergence analysis. As DCTN involves a complex learning procedure including adversarial learning and alternative adaptation, we testify the convergence performance of different losses. During the process of hard sub transfer A→W, Fig.4(c) demonstrates that, despite of the frequent deviation, the classification loss, adversarial loss and testing error gradually converge.

6. Conclusion

In this paper we have explored the unsupervised domain adaptation with multiple source domains. We raise a new MDA protocol termed *category shift*, where classes from different sources are non-consistent. Furthermore, we proposed *deep cocktail network*, a novel framework to obtain transferable and discriminative features from multiple sources. The approach can be applied to the ordinary MDA setting and category shift, and more, achieves state-of-the-art results in most of our evaluation protocols.

.

References

[1] M. Baktashmotlagh, M. Harandi, and M. Salzmann. Distribution-matching embedding for visual domain adaptation. *The Journal of Machine Learning Research*, 17(1):3760–3789, 2016. 2

[2] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010. 3, 5

[3] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman. Learning bounds for domain adaptation. In *Advances in neural information processing systems*, pages 129–136, 2008. 3

[4] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. *arXiv preprint arXiv:1612.05424*, 2016. 2

[5] P. P. Busto and J. Gall. Open set domain adaptation. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 1, 2017. 3

[6] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050*, 2016. 1

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 2

[8] L. Duan, D. Xu, and I. W.-H. Tsang. Domain adaptation from multiple sources: A domain-dependent regularization approach. *IEEE Transactions on Neural Networks and Learning Systems*, 23(3):504–518, 2012. 2, 3

[9] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967, 2013. 2

[10] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015. 6

[11] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. *Domain-Adversarial Training of Neural Networks*. 2017. 6

[12] T. Gebru, J. Hoffman, and L. Fei-Fei. Fine-grained recognition in the wild: A multi-task domain adaptation approach. *arXiv preprint arXiv:1709.02476*, 2017. 2

[13] 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. 3

[14] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016. 2, 6

[15] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073. IEEE, 2012. 1, 2, 6

[16] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 999–1006. IEEE, 2011. 6

[17] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520, 2007. 2

[18] A. Gretton, A. J. Smola, J. Huang, M. Schmittfull, K. M. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. 2009. 1

[19] P. Haeusser, T. Frerix, A. Mordvintsev, and D. Cremers. Associative domain adaptation. *arXiv preprint arXiv:1708.00938*, 2017. 2

[20] J. Hoffman, E. Tzeng, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *Domain Adaptation in Computer Vision Applications*, pages 173–187. Springer, 2017. 2, 6

[21] J. Hoffman, D. Wang, F. Yu, and T. Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016. 2

[22] I.-H. Jhuo, D. Liu, D. Lee, and S.-F. Chang. Robust visual domain adaptation with low-rank reconstruction. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2168–2175. IEEE, 2012. 2, 3

[23] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *arXiv preprint arXiv:1612.00796*, 2016. 3

[24] P. Koniusz, Y. Tas, and F. Porikli. Domain adaptation by mixture of alignments of second-or higher-order scatter tensors. *arXiv preprint arXiv:1611.08195*, 2016. 2

[25] 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. 1

[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 6

[27] H. Liu, M. Shao, and Y. Fu. Structure-preserved multi-source domain adaptation. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 1059–1064. IEEE, 2016. 2, 3

[28] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 1

[29] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105, 2015. 2, 6

[30] 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. 2, 6

[31] H. Lu, L. Zhang, Z. Cao, W. Wei, K. Xian, C. Shen, and A. van den Hengel. When unsupervised domain

adaptation meets tensor representations. *arXiv preprint arXiv:1707.05956*, 2017. 2

[32] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008. 8

[33] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation with multiple sources. In *Advances in neural information processing systems*, pages 1041–1048, 2009. 1, 2, 3, 4

[34] S. Motiian, Q. Jones, S. M. Iranmanesh, and G. Doretto. Few-shot adversarial domain adaptation. *arXiv preprint arXiv:1711.02536*, 2017. 2

[35] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. Unified deep supervised domain adaptation and generalization. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 2, 2017. 3

[36] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. *Nips Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. 6

[37] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011. 1, 2, 6

[38] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010. 1, 2

[39] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, pages 506–516, 2017. 3

[40] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1

[41] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. *Computer Vision–ECCV 2010*, pages 213–226, 2010. 6

[42] K. Saito, Y. Ushiku, and T. Harada. Asymmetric tri-training for unsupervised domain adaptation. *arXiv preprint arXiv:1702.08400*, 2017. 2

[43] H. Shin, J. K. Lee, J. Kim, and J. Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2994–3003, 2017. 3

[44] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, volume 6, page 8, 2016. 2

[45] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. *arXiv preprint arXiv:1702.05464*, 2017. 5

[46] J. Xie, W. Hu, S.-C. Zhu, and Y. N. Wu. Learning sparse frame models for natural image patterns. *International Journal of Computer Vision*, 114(2-3):91–112, 2015. 6

[47] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015. 1

[48] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. *arXiv preprint arXiv:1705.00609*, 2017. 2

[49] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 188–197. ACM, 2007. 2, 3

[50] J. Zhang, W. Li, and P. Ogunbona. Joint geometrical and statistical alignment for visual domain adaptation. *arXiv preprint arXiv:1705.05498*, 2017. 2