Attribute-assisted Domain Transfer from Image to Sketch

Yikun Sheng[1,2] Xiaoshan Yang[2,3] Xueliang Liu[1] Changsheng Xu[1,2,3] *

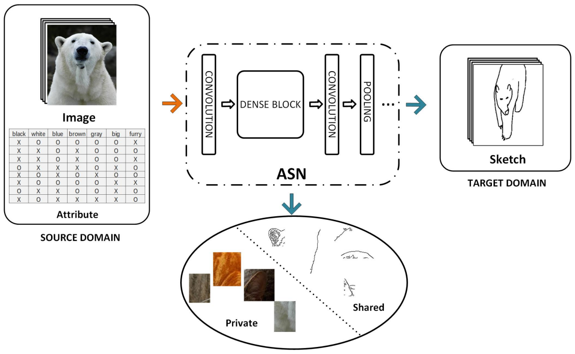
[1]Hefei University of Technology
[2]Institute of Automation, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences
{shykoe,liuxueliang}@mail.hfut.edu.cn,{xiaoshan.yang,csxu}@nlpr.ia.ac.cn

Abstract

Data-inefficiency greatly hinders sketch-related applications and studies. Transferring knowledge from the real-image domain to the sketch domain may eliminate this obstacle. However, a huge domain gap exists between images and sketches. To reduce the domain shift between the image and the sketch, we propose an attribute-assisted domain transfer method. By separating the shared geometric features from the private semantic features of the real images, the proposed attribute-assisted networks (ASN) can learn more effective domain-invariant features for unsupervised domain adaptation. Extensive results on the Image-to-Sketch task demonstrate the effectiveness of the proposed method.

1. Introduction

With the popularity of tablets, e.g. iPad and Microsoft Surface, sketch-related applications become unprecedented popular nowadays. Many sketch-related tasks, such as sketch retrieval [5], sketch detection [29] and sketch recognition [2], are also extensively studied. However, a key weakness of these approaches is their data-inefficiency. Collecting, annotating, and crating such datasets is an extremely expensive and time-consuming process. While only a few datasets and approaches have been proposed for the sketch recognition or retrieval, there are many large scale labeled datasets of real images, such as ImageNet [7], Tiny-Images [31] and SUN [35]. To eliminate obstacles bring by data-inefficiency, it is not uncommon to rely on transferring knowledge from the real life images(source domain) to sketches (target domain).



black	white	blue	brown	gray	big	furry
X	O	O	O	O	O	X
X	X	O	X	O	O	O
X	X	O	X	O	O	X
O	X	X	X	O	X	O
X	O	X	O	X	O	O
X	O	O	O	O	X	X
O	X	X	O	O	X	O
X	O	X	X	X	X	O

Figure 1. We propose an attribute-assisted domain adaptation method that utilizes the attribute to separate private features which sketches do not have.

To transfer knowledge from the real life images, the principle problem is to reduce the domain gap exists between the sketch and the real image. The previous work attempts to either find a feature mapping from the source domain to the target [30], or find representations that are shared between the two domains [13, 32, 20]. However, in the image-to-sketch transfer, it is extremely difficult to obtain the mapping or shared representations. Because the domain shift between sketches and images is much larger than the domain difference in conventional domain adaptation task, such as the popularly used Office dataset [27] which contains images from Amazon.com and images from a webcam. The sketches are essentially different with the real life images in many aspects. For example, the information of sketches is mostly represented by geometric variations, in contrast, however, the things in real images are very likely to be with rich feature.

In this paper, we propose a novel architecture, which we call Attribute-assisted Networks (ASN), to learn domain-invariant representations. Our model learns a private subspace for the real image assisted by the attribute constraint, which captures domain specific properties, such as "color" and "fur" of animals. A shared subspace is also learned through the use of domain constraint to capture representations shared by the images and sketches, such as edge and

*This work was supported in part by National Natural Science Foundation of China (No. 61432019, 61572498, 61532009, 61702511, 61720106006, 61711530243), Beijing Natural Science Foundation (4172062), and Key Research Program of Frontier Sciences, CAS, Grant NO. QYZDJ-SSW-JSC039.

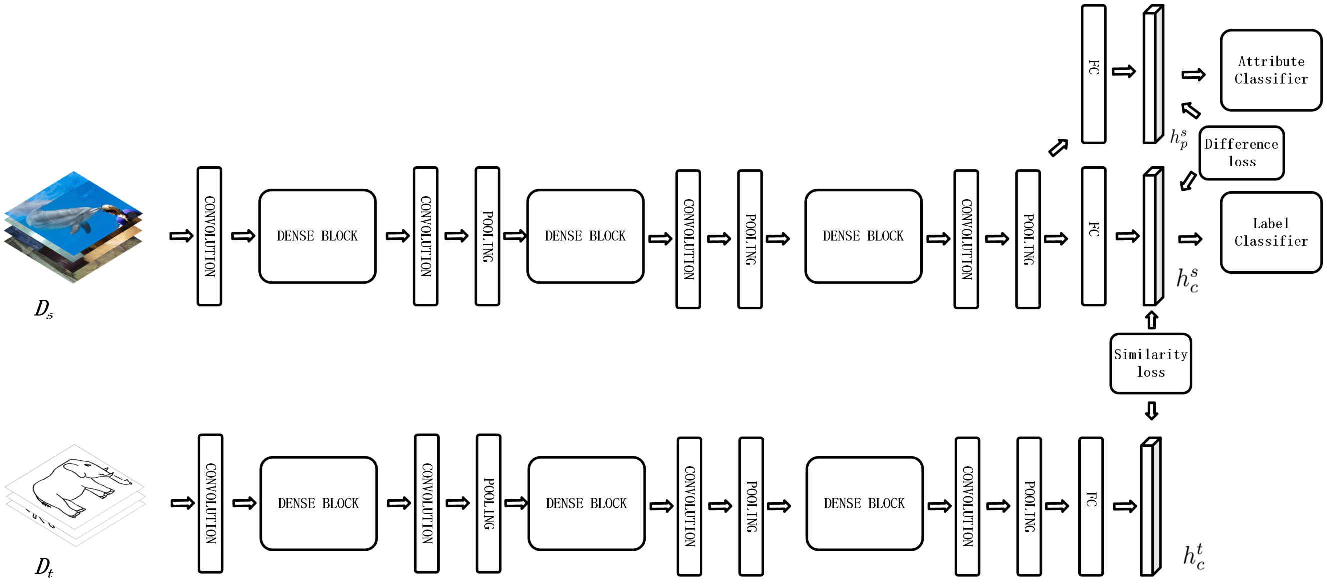


Figure 2. Architecture of the proposed Attribute-assisted Networks.

low level geometric. By separating the shared features from the private features of the real images, our model is able to separate the information that is unique to real images, and produce representations that are easier for transferring to the sketch domain. Figure 1 shows the key idea of the proposed attribute-assisted domain adaptation.

The main contributions of this paper are highlighted as follows. **(1)** We propose an attribute-assisted domain adaptation framework for the image-to-sketch transformation task. **(2)** We propose attribute-assisted networks (ASN) to reduce the huge domain shift between images and sketches by separating the shared features from the private features of the real image. **(3)** Extensive experiments demonstrate the effectiveness of the proposed ASN.

2 Related Work

In this section, we briefly review the existing work most related to our work, including domain adaptation and attributes.

2.1 Domain Adaptation

Domain adaptation is an open theoretical and practical problem which aims to transfer the knowledge acquired from labeled datasets to unlabeled datasets. Solutions to domain adaptation problems can be mainly categorized into three types. **(1)** Instance-based methods reweight or subsample the source samples to match the distribution of the target domain, thus training on the reweighted source samples guarantees classifiers with transferability [16, 6]. **(2)** Parameter-based methods transfer knowledge through shared or regularized parameters of source and target domain learners, or by combining multiple reweighted source learners to form an improved target learner [9, 25]. **(3)** As deep neural networks become the state of the art on a variety of visual tasks. The last but the most popular and effective methods are feature-based methods which attempt to either find a mapping from representations of the source domain to those of the target, or find representations that are shared between the two domains.

Among feature-based methods, minimizing the maximum mean discrepancy (MMD) [14] metric is effective to minimize the divergence of two distributions. The MMD metric is computed between features extracted from sets of samples from each domain. The Deep Domain Confusion Network [33] has an MMD loss at one layer in the CNN architecture. Deep adaptation network (DAN) [20] was proposed to enhance the feature transferability by minimizing multi-kernel MMD in several task-specific layers. Another class of feature-based methods uses an adversarial objective to reduce domain discrepancy. As suggest in [3, 4], a good cross-domain representation contains no discriminative information about the origin(i.e. domain) of the input. Domain-Adversarial Neural Networks (DANN) [1, 13] exhibit an architecture whose first few feature extraction layers are shared by two classifiers trained simultaneously. In order to back-propagate the gradients computed from the domain classifier, DANN employs a gradient reversal layer (GRL). In the proposed ASN, we adopt the MMD and DANN as two alternatives of the domain constraint. Different from the previous domain adaptation methods, we use attributes to improve the domain-invariant feature learning.

2.2 Attributes

Attributes have been widely exploited for developing effective representations in a variety of computer vision applications. Attribute learning allows prediction of color or texture types [11], and can also help obtain a mid-level cue for object or face recognition [18, 34, 19]. Several early approaches attempted to detect unseen object classes by describing objects using their attributes in [10, 19]. Some other approaches discovered visual relationships between the object categories by using the learned visual attributes [26]. Recently, semantically meaningful attributes were selected in a recommender system for fine-grained recognition [8]. Different from the previous methods, we utilize task-specific attributes to separate shared features from private features of the real images.

3 Methodology

In this section, we firstly introduce the overall formulation of the proposed attribute-assisted domain adaptation. Then, we illustrate the intra-domain constraint and inter-domain constraint which are adopted to learn domain-invariant features. Finally, we illustrate the learning details of the proposed attribute-assisted networks.

3.1 Problem Formulation

Given a labeled source domain $D_s = \{(X_i^s, y_i^s, a_i^s)\}_{i=1}^{n_s}$ and an unlabeled target domain $D_t = \{X_i^t\}_{i=1}^{n_t}$, where X_i^s and X_i^t represent the image and sketch data respectively, y_i^s represents the class label, and a_i^s represents the attribute label. The source domain and target domain are sampled from different probability distributions $\mathbb{P}$ and $\mathbb{Q}$ respectively, and $\mathbb{P} \neq \mathbb{Q}$. The challenge of unsupervised domain adaptation arises in that the target domain has no labeled data, while the source classifier trained on source domain D_s cannot be directly applied to the target domain D_t due to the distribution discrepancy $\mathbb{P}(X^s, y) \neq \mathbb{Q}(X^t, y)$ [22]. Our goal is to find a deep network architecture that separate the private features and find representations that are shared between the two domains and enables the knowledge learned on the source domain to be applied to the target domain classification task.

Given an image X^s in the source domain, we assume that the shared feature h_c^s and private feature h_p^s can be computed by $h_c^s = f_c(X^s)$ and $h_p^s = f_p(X^s)$ respectively. For a sketch X^t in the target domain, we assume that the shared feature h_c^t can also computed by $h_c^t = f_c(X^t)$. The functions f_c and f_p are implemented by two neural networks as shown in Figure 2. We employ DenseNet121 [15] as our backbone network. And the dense blocks are shown in Figure 3. We decide to adopt DenseNet in our network because it outperforms ResNet on the ImageNet dataset and we believe this structure could capture more structural information than other existing network architectures. For the image X^s, the class prediction $\hat{y}^s$ and attribute prediction $\hat{a}^s$ are computed by $\hat{y}^s = MLP_y(h_c^s)$ and $\hat{a}^s = MLP_a(h_p^s)$ respectively. Here, MLP_y and MLP_a are two multi-layer perceptions.

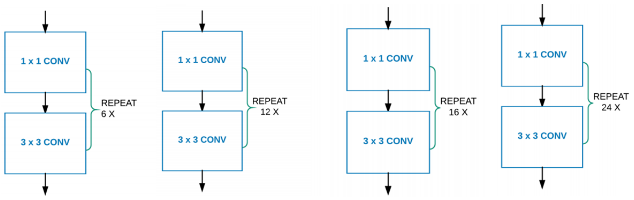


Figure 3. A simple demonstration of dense blocks. Each Conv represents the sequence of BatchNorm-ReLU-Conv.

The loss function of the proposed attribute-assisted domain adaptation is defined as:

$$\mathcal{L} = \sum_i^{n_s} \mathcal{L}_{task}(y_i^s, \hat{y}_i^s) + \alpha \sum_i^{n_s} \mathcal{L}_{attr}(a_i^s, \hat{a}_i^s) + \beta \mathcal{L}_{difference}(\boldsymbol{H}_c^s, \boldsymbol{H}_p^s) + \gamma \mathcal{L}_{similarity}(\boldsymbol{H}_c^s, \boldsymbol{H}_c^t) \tag{1}$$

where $\mathcal{L}_{task}$ and $\mathcal{L}_{attr}$ and $\mathcal{L}_{difference}$ are three intra-domain constraints which will illustrated in Section 3.2. The $\mathcal{L}_{similarity}$ is an inter-domain constraint which will be illustrated in Section 3.3. The $\boldsymbol{H}_c^s$ consists of shared features of all source samples. The $\boldsymbol{H}_p^s$ consists of private features of all source samples. The $\boldsymbol{H}_c^t$ consists of shared features of all target samples. Specifically, each row of $\boldsymbol{H}_c^s$ is the transpose of a feature vector h_c^s. The $\boldsymbol{H}_p^s$ and $\boldsymbol{H}_c^t$ are created from h_p^s and h_c^t in the same manner. The α, β, γ are weights that control the balance of the loss terms. With the defined loss $\mathcal{L}$, the proposed attribute-assisted networks can be trained to learn shared features of two domains. In the test phase, we apply the trained function f_c to extract features of the target domain D_t and use MLP_y for classification.

3.2 Intra-domain Constraint

The task loss $\mathcal{L}_{task}$ constraints the model to predict correct class labels. Because the target domain is unlabeled, the loss is applied only to the source domain. This loss is adopted to minimize the negative log-likelihood of the ground truth class for each source domain sample:

$$\mathcal{L}_{task}(y_i^s, \hat{y}_i^s) = -y_i^s \log \hat{y}_i^s \tag{2}$$

where y_i^s is the one-hot vector of the groundtruth class label and $\hat{y}_i^s$ is the prediction.

The attribute loss $\mathcal{L}_{attr}$ constraints the model to predict correct attribute labels. We use a sigmoid binomial cross-entropy:

$$\mathcal{L}_{attr}(a_i^s, \hat{a}_i^s) = -\{a_i^s \log \hat{a}_i^s + (1 - a_i^s) \log(1 - \hat{a}_i^s)\} \tag{3}$$

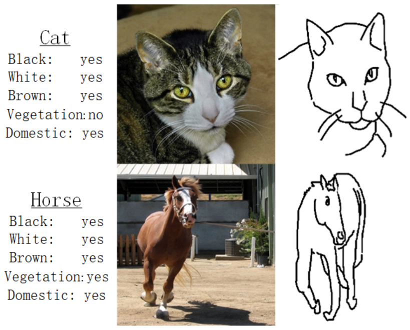


Figure 4. Examples from the Image-to-Sketch dataset.

where a_i^s is the groundtruth attribute label and $\hat{a}_i^s$ is the prediction. This loss allows the model to learn the private feature h_p^s, which is highly related to the attributes.

The difference loss $\mathcal{L}_{difference}$ is also applied to the source domain. It encourages the shared feature h_c^s and private h_p^s of the source samples to be different. Let $\boldsymbol{H}_c^s$ and $\boldsymbol{H}_p^s$ be the matrices whose row are h_c^s and h_p^s. The difference loss constraints orthogonality between the private features and shared features:

$$\mathcal{L}_{difference} = \|\boldsymbol{H}_c^{s\top}\boldsymbol{H}_p^s\|_F^2 \tag{4}$$

where$\| \cdot \|_F^2$ is the squared Frobenius norm.

3.3 Inter-domain Constraint

The $\mathcal{L}_{similarity}$ is deployed to encourages the shared features h_c^s and h_c^t from different domains to be as similar as possible. We consider two alternatives $\mathcal{L}_{similarity1}$ and $\mathcal{L}_{similarity2}$ which are designed based on maximum mean discrepancy (MMD) and domain adversarial similarity respectively.

(1) MMD loss. This is a kernel-based distance function between pairs of samples [14]. The $\mathcal{L}_{similarity1}$ is defined as:

$$\begin{aligned}\mathcal{L}_{similarity1}(\boldsymbol{H}_c^s, \boldsymbol{H}_c^t) &= \sum_i^{n_s}\sum_j^{n_s}\frac{\boldsymbol{k}(h_{ci}^s, h_{cj}^s)}{n_s^2} \\ &+ \sum_i^{n_t}\sum_j^{n_t}\frac{\boldsymbol{k}(h_{ci}^t, h_{cj}^t)}{n_t^2} - 2\sum_i^{n_s}\sum_j^{n_t}\frac{\boldsymbol{k}(h_{ci}^s, h_{cj}^t)}{n_t n_s}\end{aligned} \tag{5}$$

where h_{ci}^s and h_{ci}^t represents the shared features of the i^{th} source sample and i^{th} target sample respectively. The $\boldsymbol{k}$ represents a kernel function. In our work, we use the standard Gaussian kernel function.

(2) Domain adversarial loss. This loss is used to train a model to produce representations such that a classifier cannot reliably predict the domain. Maximizing such "confusion" is achieved via a gradient reversal layer (GRL) and a domain classifier trained to predict the domain [12, 13]. The GRL has the same output as the identity function, but reverses the gradient direction. Formally, for some function $f(\boldsymbol{u})$, the GRL is defined as $Q(f(\boldsymbol{u})) = f(\boldsymbol{u})$ with a gradient $\frac{\partial}{\partial \boldsymbol{u}}Q(f(\boldsymbol{u})) = -\frac{\partial}{\partial \boldsymbol{u}}f(\boldsymbol{u})$.

In our work, we define domain classifier as $Z(Q(h_c)) \rightarrow \hat{d}$ which maps the shared feature h_c of a input sample in two domains to a prediction of the domain label $\hat{d} \in \{0, 1\}$. The h_c denotes either h_c^s or h_c^t. Learning with a GRL is adversarial in that Z is optimized to discriminate hidden representations from the source or target domains, while the reversal of the gradient results in the model f_c learning representation h_c from which domain classification accuracy is reduced. Essentially, we maximize the binomial cross-entropy for the domain prediction task with respect to Z, while minimizing it with respect to f_c:

$$\begin{aligned}&\mathcal{L}_{similarity2}(\boldsymbol{H}_c^s, \boldsymbol{H}_c^t) = \\ &-\sum_i^{n_s+n_t}\{d_i \log \hat{d}_i + (1 - d_i)\log(1 - \hat{d}_i)\}\end{aligned} \tag{6}$$

where $d_i \in \{0, 1\}$ is the ground truth domain label for sample i.

4 Experiments

In this section, we firstly introduce the dataset and implementation details. Then we evaluate the performance of the proposed method through quantitative results and qualitative analyses.

4.1 Datasets

To demonstrate the effectiveness of the proposed model, we extend the Sketchy [28] dataset with attributes. Sketchy dataset consists of 12,500 images and 75,471 sketches (roughly 5 sketches per photo), spanning 125 categories of common objects like horse, apple, axe, guitar etc. And the attributes were collected from several existing attribute based datasets including the database of animals with attributes (AWA [19]), attribute databases for scene understanding (OSR [23], SUN [24]). The Sketchy and those attribute datasets shares 21 common categories and we build the transfer task: Image-to-Sketch with these 21 common categories. The held out test part consists of 1,100 images and 5,500 sketches, still spanning the same 21 categories. Figure 4 shows examples of some classes with the values of exemplary attributes assigned to these classes.

4.2 Implementation

We use the deep learning library Pytorch to implement the proposed attribute-assisted networks. The network parameters are initialized according to the pre-trained model on ImageNet. The optimizer is Adam [17] with an initial learning rate of 4×10^{-3}, weight decay of 1×10^{-4}. As learning rate annealing strategy could self adjust the learning during the optimization, which could help boost the

Method	Repeat 1	Repeat 2	Repeat 3	Average
Source-only	0.1196	0.1090	0.1151	0.1146
DANN [13]	0.1568	0.1850	0.1523	0.1647
MMD [20]	0.1657	0.1329	0.1331	0.1439
ASN(MMD)	0.2100	0.1719	0.1950	0.1923
ASN(DANN)	0.2571	0.2055	0.2650	**0.2425**

Table 1. Accuracy results

Method	Classification Accuracy
All terms	0.2650
Without attribute	0.2280

Table 2. Effect of the attribute-related constraints

performance of networks, we anneal the learning rate by a factor of 0.8 every five epochs. In each batch, we randomly select 64 samples from each domain. Input images and sketches are mean-centered and re-scaled to $[0, 1]$ and resized to 227 × 227 uniformly with random-cropped. In order to avoid distractions for the main classification task during the early training procedure, we set $\alpha = 0.3$, $\beta = 0$ and $\gamma = 0.3$ before 10, 000 steps of training. Then we activate additional domain adaptation losses by setting $\alpha = 0.3$, $\beta = 0.3$ and $\gamma = 0.3$. We train all models for 60,000 iterations. We have tried various number of iterations and experimental results show that the accuracy always saturate before 60,000 iterations.

4.3 Evaluation

We compare the proposed ASN model with several baselines. Two variants of our model are also compared to explore the impact of $\mathcal{L}_{similarity1}$ and $\mathcal{L}_{similarity2}$. **Source-only** is trained on the source domain without any domain adaptation, which we consider as an empirical lower bound. For the **DANN**, we applied the Gradient Reversal Layer (GRL) and then a domain classifier with one hidden layer of 100 nodes. For the **MMD**, we follow the suggestions in [20] and use multiple Gaussian kernels $\{k_u\}_{u=1}^{3}$ by varying bandwidth γ_u between 2^{-8} and 2^{8} with a multiplicative step-size of $2^{1/2}$. For our method **ASN(MMD)**, The $\mathcal{L}_{task}$ and $\mathcal{L}_{attr}$, $\mathcal{L}_{difference}$ and $\mathcal{L}_{similarity1}$ are used to train the proposed attribute-assisted networks. The $\mathcal{L}_{similarity1}$ is a single MMD loss. And we found applying MMD loss on more than one layers did not show any significant improvement for our experiments and architectures. And for our method **ASN(DANN)**, The $\mathcal{L}_{task}$ and $\mathcal{L}_{attr}$, $\mathcal{L}_{difference}$ and $\mathcal{L}_{similarity2}$ are used to train the proposed ASN.

4.3.1 Quantitative Result

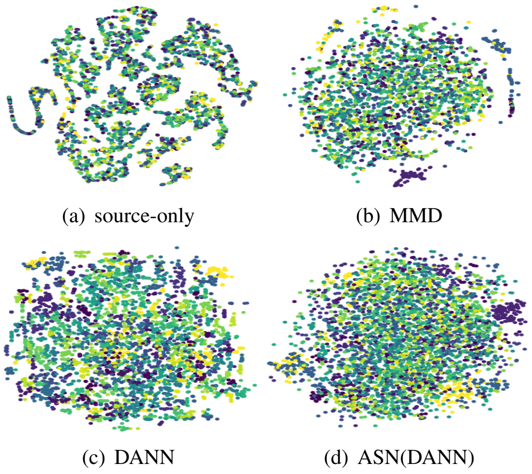
The challenge of domain transfer from real-images to sketches is that the sketches do not have any private features of the real images. Table 1 shows the detailed comparison results of all methods we experimented. As we can see, the proposed ASN model with DANN outperforms all the other methods. With the extra information attribute, our model separates the private features of the images, and retains the shared features of two domains. Compared with the conventional DANN and MMD methods, the proposed ASN is able to improve the performance with a large margin. We find that the DANN [13] outperforms MMD. This is consistent with our experiment where ASN (DANN) outperforms ASN (MMD).

In order to examine the effect of attribute-related constraints $\mathcal{L}_{attr}$ and $\mathcal{L}_{difference}$, we remove these two constraints from our best model ASN (DANN) by setting $\alpha = 0$ and $\gamma = 0$. As shown in Table 2, the accuracy of the proposed ASN without attribute-related constraints decrease about 4%.

4.3.2 Qualitative Analysis

A popular method to visualize high-dimensional data in 2D is t-SNE [21]. We are interested in the distribution of embeddings for target domain, when the models are only learned from source domain. Figure 5 shows such visualizations. In these figures, different color points represent different categories, and same color points should cluster well to present good performance.

As shown in Figure 5(a), the model trained only on the source domain without any domain adaptation presents worse performance, and we can see that the feature points are really confused. All domain adaptation methods, MMD, DANN and the proposed ASN(DANN) can learn discriminative feature representations to some extend. We could observe that DANN and our method have good ability to discriminate the clusters. And our method has nicely clustered features with learned shared representations.



(a) source-only

(b) MMD

(c) DANN

(d) ASN(DANN)

Figure 5. Feature visualization of three baselines and the proposed ASN (DANN)

5 Conclusions

In this paper, we proposed an attribute-assisted network to learn domain-invariant representations for image-to-sketch domain adaptation. By separating the private features of the real images, the proposed ASN captured representations shared by images and sketches. The proposed ASN performed well on the Image-to-Sketch domain adaptation. In future work, we will explore more variants of the network architecture and apply our method on other domain adaptation tasks.

References

[1] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, and M. Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *TPAMI*, 24(4):509–522, 2002.

[3] 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.

[4] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Nips*, pages 137–144, 2007.

[5] Y. Cao, H. Wang, C. Wang, Z. Li, L. Zhang, and L. Zhang. Mindfinder: interactive sketch-based image search on millions of images. In *MM*, pages 1605–1608. ACM, 2010.

[6] M. Chen, K. Q. Weinberger, and J. Blitzer. Co-training for domain adaptation. In *Nips*, pages 2456–2464, 2011.

[7] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li. Imagenet: A large-scale hierarchical image database. In *CVPR 2009*, pages 248–255, 2009.

[8] K. Duan, D. Parikh, D. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *CVPR 2012*, pages 3474–3481. IEEE, 2012.

[9] L. Duan, D. Xu, and S.-F. Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *CVPR 2012*, pages 1338–1345. IEEE, 2012.

[10] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR 2009*, pages 1778–1785. IEEE, 2009.

[11] V. Ferrari and A. Zisserman. Learning visual attributes. In *Nips*, pages 433–440, 2008.

[12] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189, 2015.

[13] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17(59):1–35, 2016.

[14] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *JMLR*, 13(Mar):723–773, 2012.

[15] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

[16] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola. Correcting sample selection bias by unlabeled data. In *Nips*, pages 601–608, 2007.

[17] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, pages 365–372. IEEE, 2009.

[19] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR 2009*, pages 951–958. IEEE, 2009.

[20] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105, 2015.

[21] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *JMLR*, 9(Nov):2579–2605, 2008.

[22] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Nips*, pages 3111–3119, 2013.

[23] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, pages 503–510. IEEE, 2011.

[24] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR 2012*, pages 2751–2758. IEEE, 2012.

[25] A. Rozantsev, M. Salzmann, and P. Fua. Beyond sharing weights for deep domain adaptation. *arXiv preprint arXiv:1603.06432*, 2016.

[26] O. Russakovsky and F.-F. Li. Attribute learning in large-scale datasets. In *ECCV Workshops (1)*, volume 6553, pages 1–14, 2010.

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

[28] P. Sangkloy, N. Burnell, C. Ham, and J. Hays. The sketchy database: learning to retrieve badly drawn bunnies. *TOG*, 35(4):119, 2016.

[29] J. Shotton, A. Blake, and R. Cipolla. Multiscale categorical object recognition using contour fragments. *TPAMI*, 30(7):1270–1281, 2008.

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

[31] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *TPAMI*, 30(11):1958–1970, 2008.

[32] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, pages 4068–4076, 2015.

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

[34] Y. Wang and G. Mori. A discriminative latent model of object classes and attributes. *Computer Vision–ECCV 2010*, pages 155–168, 2010.

[35] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR 2010*, pages 3485–3492, 2010.