Discrepancy-based networks for unsupervised domain adaptation: a comparative study

Gabriela Csurka[†], Fabien Baradel[‡], Boris Chidlovskii[†] and Stéphane Clinchant[†]
[†]Naver Labs Europe, Meylan, France
firstname.lastname@naverlabs.com
[‡] LIRIS, Lyon, France
fabien.baradel@insa-lyon.fr

Abstract

Domain Adaptation (DA) exploits labeled data and models from similar domains in order to alleviate the annotation burden when learning a model in a new domain. Our contribution to the field is three-fold. First, we propose a new dataset, LandMarkDA, to study the adaptation between landmark place recognition models trained with different artistic image styles, such as photos, paintings and drawings. The new LandMarkDA proposes new adaptation challenges, where current deep architectures show their limits. Second, we propose an experimental study of recent shallow and deep adaptation networks, based on using Maximum Mean Discrepancy to bridge the domain gap. We study different design choices for these models by varying the network architectures and evaluate them on OFF31 and the new LandMarkDA collections. We show that shallow networks can still be competitive under an appropriate feature extraction. Finally, we also benchmark a new DA method that successfully combines the artistic image style-transfer with deep discrepancy-based networks.

1. Introduction

Domain adaptation (DA) has recently received a lot of attention in computer vision (see [6] for a comprehensive survey). Early DA models such as manifold-based feature augmentation [14, 15], feature space alignment [9, 26] and unsupervised feature transformation [24, 2, 22] exploit directly the data distribution in the source and target domains without the class labels; they build cross-domain representations that allow models learned with the labeled source data be suitable to classify instances in the target domain.

Recently, deep learning methods allowed a significant improvement of the categorization accuracy over the state-of-the-art solutions. Furthermore, it was shown that features extracted from the activation layers of the deep CNNs can be re-purposed to novel tasks [8] even when the new tasks differ significantly from the task originally used to train the model. The OFF31 dataset, frequently used to study the DA case, contains images similar to the ImageNet images usually used to train the CNN models. Therefore training or fine-tuning a classifier on the source with these features often performs well on the target domain even without specific adaptation [4, 26]. This is not anymore true when we want to reuse features/models between domains with important domain differences such as *e.g.* photos and paintings, drawings, clip art or sketches [19, 5, 3, 33]; hence the need for DA becomes more crucial.

On one hand, deep CNN architectures can be used as features extractors and shallow DA methods applied on source and target sets represented by these features [8, 26, 7, 25]. On the other hand, recently deep learning architectures have been designed for DA. These methods in general follow a Siamese architectures with two streams, representing the source and target models, and combine classification loss with DA specific losses. These include *discrepancy loss* [32, 21, 13, 27, 23], *confusion loss* [30], *inverted label GAN loss* [31] or integrates a *gradient reversal layer* into the standard architecture to promote features discriminative for the task and invariant with respect to the domain [10].

In this paper we focus on discrepancy-based adaptation networks for unsupervised DA. We propose a comparative experimental study of various shallow (SDAN) and deep (DDAN) architectures, under different weight sharing strategies and discrepancy choices.

To challenge these methods with stronger domain differences than in the OFF31 dataset, we propose a new dataset called LandMarkDA that contains photos, paintings and drawings of 25 landmark places and monuments around the world such as *Eiffel Tower* or *Machu Pitchu* (see Figure 1).

To further study the role of image style change between domains and the possibility of model adaptation between them, we propose a method that successfully combines DDAN with artistic image style-transfer [11, 12].

To summarize, the paper brings the following contributions: (1) a comparative experimental study of SDAN and DDAN models, under different weight sharing strategies and various marginal and joint discrepancies between the

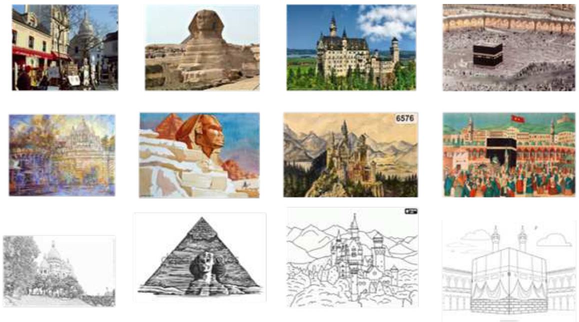


Figure 1. Examples from the LandMarkDA dataset. Each line corresponds to one image modality (domain) and each column refers to one landmark (class).

domains; (2) a new DA dataset to evaluate adaptation of landmark recognition model between photos, paintings and drawings; and (3) a new DA method that combines DDAN with artistic style-transfer between images.

The remainder of the paper is organized as follows. In section 2 we describe the SDAN and DDAN models as well as marginal and joint distribution discrepancies used by these models. Then, in section 3 we present an artistic style-transfer based DDAN. We report experimental results and our findings in section 4 and we conclude in section 5.

2. Discrepancy-based adaptation networks

2.1. Deep Networks (DDAN). We consider Deep Discrepancy-based Adaptation Networks (DDAN) as the set of two stream Siamese deep networks (see an example in Figure 2), with the streams representing the source and target models, and the model is learned using a combination of *class prediction loss* with a *discrepancy loss* based on the Maximum Mean Discrepancy (MMD) criterion [17]. The models proposed in [32, 21, 27, 23] are different variants of the DDAN family.

2.2. Shallow Networks (SDAN). As an alternative to deep architectures, shallow networks (1-layer) can also be considered for domain adaptation as in [13]. The Shallow Discrepancy-based Adaptation Network (SDAN) is a Siamese architecture (see Figure 3), where source and target streams are shallow networks built on image representations, each with a fully connected layer corresponding to the transformation learning (latent feature layer $\widehat{\mathbf{X}}$) and with a corresponding class prediction layer $\widehat{\mathbf{Y}}$.

SDAN is built directly on vectorial image representations similarly to standard shallow DA methods [24, 2, 7]. As such, it is independent from the feature extraction process and can be applied to any type of data allowing vectorial representation.

2.3. Variants of Maximum Mean Discrepancy. The MMD is an effective non-parametric criterion that compares the distributions of two sets of data. Let $\mathbf{X}_s = \{\mathbf{x}_i^s\}, i = 1, \ldots, N_s$, and $\mathbf{X}_t = \{\mathbf{x}_i^t\}, j = 1, \ldots, N_t$ be the sample sets from distributions $p_X = P(\mathbf{X}_s)$ (source) and $q_X = P(\mathbf{X}_t)$ (target), respectively.

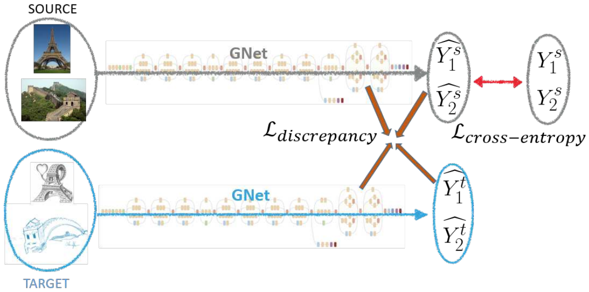


Figure 2. Diagram of the DDAN built on the GoogleNet Inception V3 model (denoted by GNet). The discrepancy loss is defined between either class prediction layer, activation layers or both.

In DA, the goal is to minimize the discrepancy between *marginal* distributions p and q to reduce the domain shift. In practice, this is done by estimating MMD by the square difference between two *empirical kernel mean embeddings*:

$$\mathcal{D}_{MMD} = \left\| \frac{1}{N} \sum_{i}^{N_s} \phi(\mathbf{x}_i^s) - \frac{1}{M} \sum_{i}^{N_t} \phi(\mathbf{x}_i^t) \right\|_{\mathcal{F}}^2 ,$$

where $\phi(\cdot) \in \mathcal{F}$ is the mapping of $\mathcal{X}$ to the RKHS, and $k(\cdot,\cdot) = \langle \phi(\cdot), \phi(\cdot) \rangle$ is the universal kernel associated with this mapping.

The MMD between the marginal distributions has been extended in [23] to *joint* distributions $P(\mathbf{X}, \mathbf{Y})$. The JDD measures the discrepancy between two joint distributions $p = P(\mathbf{X}_s, \mathbf{Y}_s)$ and $q = P(\mathbf{X}_t, \mathbf{Y}_t)$ as squared distance between the corresponding kernel embeddings, where $\mathbf{Y}_s = \{\mathbf{y}_i^s\}, i = 1, \ldots, N_s$, and $\mathbf{Y}_t = \{\mathbf{y}_i^t\}, j = 1, \ldots, N_t$ contain binary labels or class predictions corresponding to $\mathbf{X}_s$ and $\mathbf{X}_t$, respectively. Its empirical estimate, denoted by $\mathcal{D}_{JDD}$, is given by:

$$\left\| \frac{1}{N_s} \sum_{i}^{N_s} \phi(\mathbf{x}_i^s) \otimes \psi(\mathbf{y}_i^s) - \frac{1}{N_t} \sum_{i}^{N_t} \phi(\mathbf{x}_i^t) \otimes \psi(\mathbf{y}_i^s) \right\|_{\mathcal{F} \otimes \mathcal{G}}^2 ,$$

where $\psi(\cdot) \in \mathcal{G}$ is the mapping of $\mathcal{Y}$ to the RKHS.

3. Style-transfer for domain adaptation

One major motivation of this paper is to cope with adaptation when we have different artistic image styles such as photos, paintings and drawings. In addition to evaluating various SDAN and DDAN models on the LandMarkDA dataset, we also design a new DA method based on recent artistic style-transfer methods [11, 12, 18].

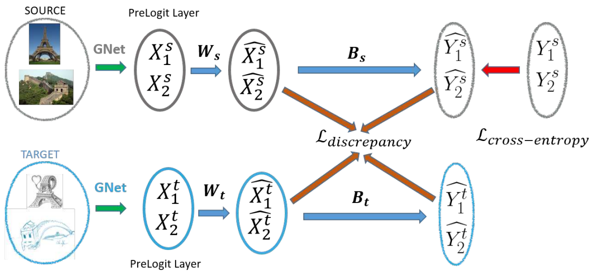


Figure 3. Diagram of SDAN built on the features extracted from the Prelogit layer of GNet ($\mathbf{X}$). The discrepancy loss is defined between either the class prediction layers ($\widehat{\mathbf{Y}}$), the latent feature layer ($\widehat{\mathbf{X}}$) or both (see details in section 2.3.).

The main idea is the following. We consider a set of labeled source images, for example *Photos*, on which we train a source model. The aim is to adapt this model to target *Paintings* or *Drawings* for which the labels are not available. Given a source-target image pair, we can apply an artistic style-transfer method [11, 12] to generate a new style-transfered source image with the semantic content unchanged but the artistic style borrowed from the target image[1]

As the style-transfer is completely unsupervised, a new image can be generated from any source-target pair of images (see examples in Figure 4). The generated images inheriting the label from the source images, form a new labeled ”source” set, called *style-transferred (ST) source set* that can be used to train or refine the model for the target set. As experiments in Section 4 show, these new models outperform significantly the fine-tuned or DA based models obtained with the original source data.

4. Experimental Results

4.1. Datasets. First, in our experiments we consider the Office 31 (OFF31) [14] dataset as it is the most used to evaluate visual DA methods. We also propose a new DA dataset, called LandMarkDA[2], created to assess how DA methods can cope with adaptation of landmark recognition models between domains such as *Photos* (Ph), *Paintings* (Pt) and *Drawings* (Dr). The dataset contains images of 25 different touristic landmark places (classes) such as The Eiffel Tower, Golden Gate, The Statue of Liberty, Taj Mahal (see Figure 1). The dataset includes in average 60 images per class and modality, in total about 1500 per modalities.

4.2. Features. We consider several deep architectures pretrained on ImageNet, such as AlexNet (ANet), VGGNET (VNet), ResNet50 (RNet) and GoogleNet Inception V3 (GNet) as feature extractors. The output activations of the fully connected layer fc6 is considered for ANet and VNet, and the activations of the last fully connected layer (PreLogit) preceding the class prediction layer (Logit) in the case of RNet and GNet. In what follows, we use the notations ANet, VNet, RNet and GNet both to design the deep models as well as the considered activation features. Note that SDAN can be applied to any vectorial representation. In addition, we consider the Regional Maximum Activations of Convolutions (RMAC) [29] model trained on ImageNet as well as fine-tuned (RMAC-FT) on a large Landmark retrieval (LandmarkRet) dataset [1] and the Learned RMAC (LRMAC) model [16] trained in an end-to-end manner on the LandmarkRet using a ranking loss within a three-stream Siamese network. These models were used as black box feature extractors, and the features were used only to perform the SDAN experiments. The original RMAC was also used in the case of OFF31.

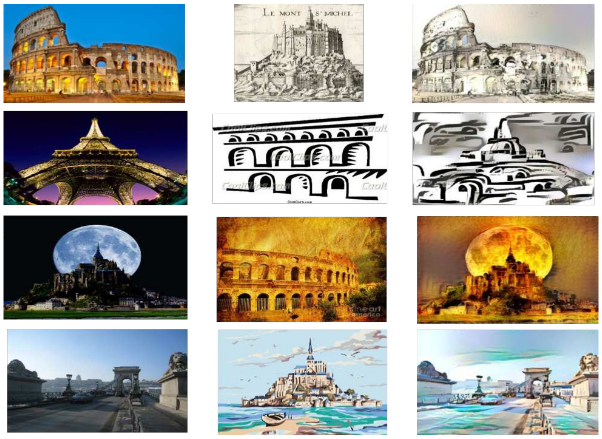


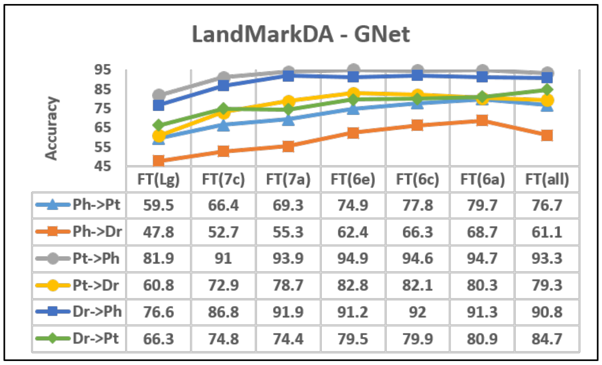
Figure 4. Style-transfered (ST) examples. The source image (left) provides the content, the unlabeled target image (middle) the artistic style. The resulting ST source image (right) inherit the label from the source, but is closer in artistic style to the target images.

4.3. Discrepancy. When we compute the empirical estimates of the discrepancies, we consider for both classes of functions $\mathcal{F}$ and $\mathcal{G}$, a set of Gaussian kernels [13, 21, 23]. We experiment with the *marginal* MMD between the activation layers $\widehat{\mathbf{X}}$ (the corresponding discrepancy loss denoted by MMD-X) and between class predictions $\widehat{\mathbf{Y}}$ (MMD-Y). In addition to considering them individually, we compare their sum (MMD-XY) to the corresponding joint discrepancy (JDD-XY). In all cases, we use an average of Gaussian kernel embeddings with the following range of σ values: $k_\phi(\mathbf{x}_i,\mathbf{x}_j) = 1/9\sum_{m=0}^{8} e^{-\|\mathbf{x}_i-\mathbf{x}_j\|/10^m}$ and $k_\psi(\mathbf{y}_i,\mathbf{y}_j) = 1/4\sum_{m=-3}^{0} e^{-\|\mathbf{y}_i-\mathbf{y}_j\|/10^m}$.

4.4. Experimental Setup. Both SDAN and DDAN are initialized by training independently the source stream (up-

[1] Note that the method works also well with a small accuracy decrease if we use random images corresponding to the target style, such as paintings or photos, but not necessarily the target images.

[2] The dataset is available at https://www.researchgate.net/publication/319208011_LandMarkDA_domain_adaptation_dataset.



	FT(Lg)	FT(7c)	FT(7a)	FT(6e)	FT(6c)	FT(6a)	FT(all)
Ph->Pt	59.5	66.4	69.3	74.9	77.8	79.7	76.7
Ph->Dr	47.8	52.7	55.3	62.4	66.3	68.7	61.1
Pt->Ph	81.9	91	93.9	94.9	94.6	94.7	93.3
Pt->Dr	60.8	72.9	78.7	82.8	82.1	80.3	79.3
Dr->Ph	76.6	86.8	91.9	91.2	92	91.3	90.8
Dr->Pt	66.3	74.8	74.4	79.5	79.9	80.9	84.7

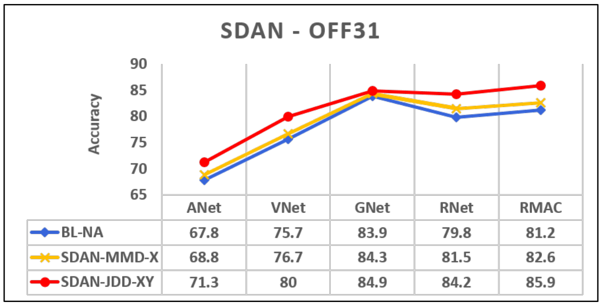
Figure 5. Fine-tuning GNet up to inception block 7c, 7d, *etc.* or training only the classifier layer (Lg) on source.

per parts in Figures 2 and 3)with the source data using *cross-entropy loss* only. Then both streams in the Siamese architecture are initialized with the weights of the source network and the model weights are learned with the union of the labeled source and unlabeled target set. This is done by error back-propagation based on the average between the *cross-entropy loss* on the source data and the *discrepancy loss* defined between the source and target sets. We compared different discrepancy losses MMD-X, MMD-Y, MMD-XY and JDD-XY (see definitions above).

To evaluate a particular architecture, we train the model several times, using a predefined number of iterations (early stopping criteria) and average their results. We used the overall classification accuracy as evaluation measure. If not explicitly mentioned, results are averaged over all source-target configurations of a dataset.

4.5. Fine-tuning on the source domain. First, we evaluate how pre-trained ImageNet models, fine-tuned on the source set perform on the target. This can be seen as a baseline which uses no adaptation to the target (NA). Note that previous studies on deep model fine-tuning were done in the context of image categorization, *i.e*. the training and test sets come from the same domain. The DA case is different from it because the model is fine-tuned on one domain and tested on another one. Additional difficulty in the case of unsupervised DA is that no labels are available to select the parameters using cross validation and cross validating the parameters on the source often works poorly (probably due to over-fitting on the source).

In this paper we mainly focus on GoogleNet [28], as it yields much better results than AlexNet or VGGNet and as it was less studied in the context of DA. We experimented with fine-tuning[3], the whole ImageNet model, fine-tuning part of it (*e.g*. up to inception block 7a) or training only the Logit layer. We denote any model by the layer up to which the model was fine-tuned: *e.g*. FT(7a) means that all layers preceding the inception block 7a were frozen and the others including 7a were fine-tuned. FT(all) means that all layers were fine tuned.



	ANet	VNet	GNet	RNet	RMAC
BL-NA	67.8	75.7	83.9	79.8	81.2
SDAN-MMD-X	68.8	76.7	84.3	81.5	82.6
SDAN-JDD-XY	71.3	80	84.9	84.2	85.9

Figure 6. SDAN results on the OFF31. using different image representations described in Independently of the image representation used (see section 4.2), SDAN outperforms the baseline without adaptation (BL-NA). However, while the gain over BL-NA using SDAN-MMD-X is relatively small, SDAN-JDD-XY significantly outperformed both.

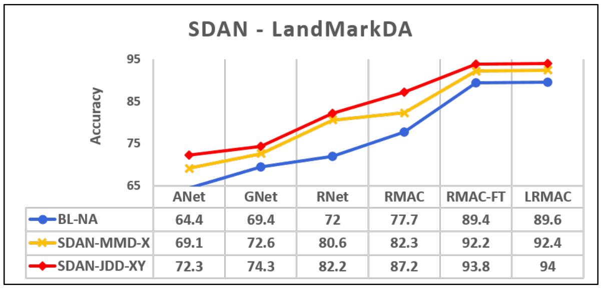
In the case of OFF31, the best results were obtained either with FT(Lg) or FT(7c). The main reason is that images in OFF31 are similar to ImageNet and the main focus should be on the classifiers and not on the features. In the LandMarkDA dataset, where both the images and tasks are different from the ImageNet, we observe a very different behavior. We show our results in Figure 5. From these results we can see that in general the accuracy increases with the number of layers fine-tuned, but after a while the gain becomes less important while the training cost continues to increase and we might even observe a slight decrease.

4.6. Comparison of SDAN architectures. First, preliminary experiments have shown that results with non shared parameters, *i.e*. using domain specific transformations and classifiers ($\mathbf{W}_s \neq \mathbf{W}_t$ and $\mathbf{B}_s \neq \mathbf{B}_t$ in Figure 3) were in general similar or below the results obtained with shared parameters[4] ($\mathbf{W}_s \neq \mathbf{W}_t$ and $\mathbf{B}_s \neq \mathbf{B}_t$). Therefore, we only consider here SDAN with shared parameters.

Willing to compare different discrepancy choices, we consider a variety of feature representations, for OFF31 and LandMarkDA. We show results averaged over all source-target pairs in Figures 6 and 7, respectively. For the sake of better visualization, we only show the results for MMD-X and JDD-XY, as JDD-XY performed on average the best, and MMD-X is similar to the model proposed in [13]. Note

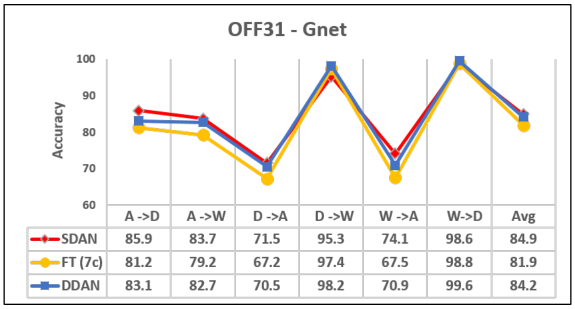
[3] We used tensorflow implementation and pre-trained model from `https://github.com/tensorflow/models/tree/master/slim` with 10K batches of 128 images, a momentum of 0.9 and initial learning rate of 0.1 with a decay of 0.95.

[4] The latter is strongly related to feature transformation based shallow DA methods that learns a common projection of the data into a latent space where the domain shift is minimized and a cross-domain classifier is trained in this space using the source set [24, 2, 7]. SDAN combined these two steps learning the feature transformation and the source classifier simultaneously.



	ANet	GNet	RNet	RMAC	RMAC-FT	LRMAC
BL-NA	64.4	69.4	72	77.7	89.4	89.6
SDAN-MMD-X	69.1	72.6	80.6	82.3	92.2	92.4
SDAN-JDD-XY	72.3	74.3	82.2	87.2	93.8	94

Figure 7. SDAN results on LandMarkDA. Best results are obtained with SDAN-JDD-XY which significantly outperforms both BL-NA and SDAN-MMD-X independently of the image representation used (see section 4.2).



	A ->D	A ->W	D ->A	D ->W	W ->A	W->D	Avg
SDAN	85.9	83.7	71.5	95.3	74.1	98.6	84.9
FT (7c)	81.2	79.2	67.2	97.4	67.5	98.8	81.9
DDAN	83.1	82.7	70.5	98.2	70.9	99.6	84.2

Figure 8. OFF31 results based on GNet. DDAN performs similarly to SDAN built on the features extracted from the PreLogit layer and both outperforms FT.

that MMD-Y (not shown) outperforms MMD-X, suggesting that minimizing the discrepancy between class predictions is more important than minimizing the discrepancy between the latent features. Nevertheless, they complement each other within MMD-XY or JDD-XY.

These models are compared to the following baseline. We consider the pretrained source stream (the upper part of SDAN in Figure 3), used to initialize the target stream in SDAN, and evaluate it on the target set. This model, denoted by BL-NA, is trained only on the source without any adaptation to the target. From the results in Figures 6 and 7, we can see that both SDAN-MMD-X and SDAN-JDD-XY are able to take advantage of the unlabeled target data with, SDAN-JDD-XY performing much better than SDAN-MMD-X.

Finally, we compare the OFF31 results to shallow methods using deep features reported in the literature. These methods use in general DeCAF6 features, which are the same as our ANet features except an L2 normalization. Using these features, [26] reports an average of 62.2% without adaptation (NA-fc6), 49.1% using GFK [14] or SA [9], 50.9% using TCA [24] and 64% using CORAL-fc6 [26]. First, we can observe that both our BL-NA baseline (67.8%) and SDAN-JDD-XY (71.3%) perform significantly better than these shallow methods. The SDAN-JDD-XY performs similarly to several deep DA architectures such as DDC [32] (70.6%), DeepCORAL [27] (72.1%), DAN [22] (72.9%) but remain below JAN-xxy [23] (76%).

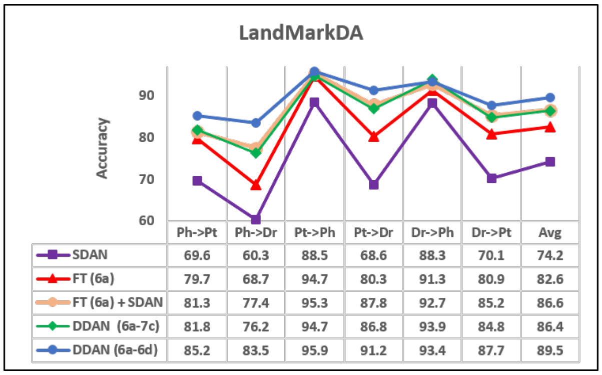
4.7. Comparison of DDAN architectures. In our DDAN experiments we mainly focus on the GNet model, and we initialize the two streams in the Siamese architecture with the best fine-tuned models, *i.e.* FT(7c) for OFF31 and FT(6a) for LandMarkDA. Then the model is trained with batches of source and target instances where the error backpropagation relies on both the source cross-entropy loss and the discrepancy loss.

As best performances were obtained in general with JDD-XY, we only consider DDAN-JDD-XY compared to SDAN-JDD-XY. We refer to them as DDAN and SDAN. To compute the JDD-XY discrepancies we consider the PreLogit layer as $\widehat{\mathbf{X}}$ and the Logit layer as $\widehat{\mathbf{Y}}$. Note that we ran complementary experiments where we added to this loss the discrepancies defined between other layers. On one hand, we tried to add JDD-XY defined between the auxiliary PreLogit and auxiliary Logit layers, but the gain compared to the cost was relatively small. On the other hand, when we added discrepancies using additional activation layers either as MMD-X or JDD-XY (combined with the Logit layer), the gain was higher but it came with a significant extra cost. Therefore, we believe that considering only the JDD-XY defined with the PreLogit and Logit layers is a good compromise between accuracy and cost.

We ran another set of experiments to compare weight sharing strategies. We observed that letting the parameters in the two branches to be domain specific, yields 1-2% of accuracy increase in average on LandMarkDA compared to the shared case, but affects little the OFF31 results, probably because most layers were initialized with the same weights and frozen during the training. Hence, in contrast to SDAN, DDAN results below were obtained with the non-shared case.

From results for the OFF31 shown in Figure 8 we see that DDAN (84.2%) built on FT(7c) performed better than FT (80.7%) and similarly to SDAN using the GNet features (84.9%). We have already observed that FT(7c) performed similarly to FT(Lg) where only the source classifier is learned. All this suggests that in the case of OFF31 the "*domain shift*" can be solved by using strong deep architectures pretrained on ImageNet as feature extractors combined with shallow DA methods.

In case of LandMarkDA, after initializing the two streams with FT(6a), we trained different models, varying the amount of layers in the two streams we selected to freeze during the training, *i.e.* we update the parameters in the model up to inception block 7c, 7a, 6d or 6a, respectively. What we observed was that while updating more layers was beneficial, in general it was sufficient to focus on the few



	Ph->Pt	Ph->Dr	Pt->Ph	Pt->Dr	Dr->Ph	Dr->Pt	Avg
SDAN	69.6	60.3	88.5	68.6	88.3	70.1	74.2
FT (6a)	79.7	68.7	94.7	80.3	91.3	80.9	82.6
FT (6a) + SDAN	81.3	77.4	95.3	87.8	92.7	85.2	86.6
DDAN (6a-7c)	81.8	76.2	94.7	86.8	93.9	84.8	86.4
DDAN (6a-6d)	85.2	83.5	95.9	91.2	93.4	87.7	89.5

Figure 9. LandmarkDA results based on GNet model. Deep models (FT and DDAN) significantly outperformed SDAN trained on the GNet features, but SDAN with feature from the fine-tuned model performed similarly to to DDAN(6a-7c). Best results were obtained with DDAN(6a-6d).

last inception blocks. Close to best results were obtained with DDAN(6a-6d), where FT(6a) was used to initialize the two streams of the DA model, then the parameters were updated in both streams up to inception block 6d included without sharing those weights[5].

From the results shown in Figure 9, we can see that the deep models, both FT(6a) and DDAN, perform significantly better than SDAN and the gain from FT to DDAN is always important. In addition we can observe that DDAN(6a-6d) outperforms significantly DDAN(6a-7c) meaning that going deeper in adaptation was beneficial. Finally, we also considered the features extracted from the fine-tuned model FT(6a) and used in combination with SDAN. While the results were below DDAN (6a-6d), the method performed similarly to DDAN (6a-7c) and better than FT(6a). This means that fine-tuning the deep model on the source and combining it with shallow method is a good and lower cost alternative to the deep DA model[6].

4.8. Style-transfer based DA. Finally, we evaluate the artistic style transfer based preprocessing described in section 3 on the LandMarkDA dataset. While it is possible to generate a style transfered (ST) image from all source-target pairs; instead, we generate only a single image for each source instance, for which we select randomly a target image (which in general is from a different class than the source) and we transfer the artistic style of this target image to the source (see examples in Figure 4). This new set of images is referred as ST-source.

We consider the GNet based models FT(6a) and DDAN(6a-6d) trained similarly as above, but instead of using the the original source set we used the ST-source set. We did this for the configurations Ph→Pt and Ph→Dr. In both cases we observed significant improvements compared to the case when the original source set was used. In the case of DDAN we get 89.4% on Pt and 91.1% on Dr instead of 85.2% respectively 83.5%. More surprisingly, fine-tuning on the ST-source set outperformed DDAN trained on the original source set (as we obtain 86.9% on Pt and 83.9% on Dr) confirming that with the style-transfer the shift between the domains was already significantly reduced. The results confirm experimentally the findings in [20] where it was shown that the artistic style transfer can be casted as domain adaptation with a specific MMD. Indeed, as the ST-source and the target set share the same style, the problem is reduced to transfer of knowledge on image content (*i.e.* the related category labels).

4.9. Findings. First, given a deep architecture, best results in general are obtained with deep DA models, however using a fine-tuned deep model as feature extractor combined with SDAN is a good compromise between accuracy and computational cost. Note that for any representation, SDAN remains a low cost solution that allows improvement over baselines obtained without adaptation.

Second, while we observed the importance to minimize the discrepancy on the class predictions, the data in both datasets is relatively uniformly distributed over the classes. Therefore, we run recent experiments with modified distributions and observed a drop both in MMD-Y and JDD-XY compared to MMD-X. This suggests that for datasets with unbalanced data, MMD-X remains a safer option.

Third, in the case of DDAN, while considering several activation layers in the discrepancy space might improve the results, the gain remains low compared to the high additional cost. Also we found that while sharing weights performed better for SDAN, allowing DDAN to learn domain specific weights was in general beneficial.

Finally, we have seen that transferring the target style to source images and using them to fine-tune the GNet model allowed a better adaptation to the target than DANN using the original source set. Using ST-source set to train the DDAN allows to further improve the results.

5. Conclusion

In this paper we proposed a comparative experimental study for DA by comparing for both shallow and deep adaptation networks different deep architectures and discrepancies. These models were tested on a standard DA dataset, and on a new DA dataset we proposed. In addition, for the landMarDA dataset, we have shown that applying artistic style-transfer from random target images to the source set reduces significantly the domain shift yielding to further improvements.

[5] Note that the weights preceding the block 6d are shared and the weights preceding 6a are the ones from the Imagnet model.

[6] Note that similar behavior can be observed if we compare CORAL [26] results obtained with (69.4%) or without (66.9%) fine-tuned features and the results of DeepCORAL [27] (72.1%).

References

[1] A. Babenko, A. Slesarev, A. Chigorin, and V. S. Lempitsky. Neural codes for image retrieval. In *ECCV*, 2014. 3

[2] M. Baktashmotlagh, M. Harandi, B. Lovell, and M. Salzmann. Unsupervised domain adaptation by domain invariant projection. In *ICCV*, 2013. 1, 2, 4

[3] L. Castrejón, Y. Aytar, C. Vondrick, H. Pirsiavash, and A. Torralba. Learning aligned cross-modal representations from weakly aligned data. In *CVPR*, 2016. 1

[4] S. Chopra, S. Balakrishnan, and R. Gopalan. DLID: Deep learning for domain adaptation by interpolating between domains. In *ICML Workshop on Challenges in Representation Learning (WREPL)*, 2013. 1

[5] E. J. Crowley and A. Zisserman. In search of art. In *ECCV Workshop on Computer Vision for Art Analysis*, 2014. 1

[6] G. Csurka. Domain adaptation for visual applications: A comprehensive survey. *CoRR*, arXiv:1702.05374, 2017. 1

[7] G. Csurka, B. Chidlovskii, S. Clinchant, and S. Michel. Unsupervised domain adaptation with regularized domain instance denoising. In *ECCV workshop on Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV)*, 2016. 1, 2, 4

[8] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014. 1

[9] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2013. 1, 5

[10] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 2016. 1

[11] L. Gatys, A. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In *NIPS*, 2015. 1, 2, 3

[12] L. Gatys, A. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. 1, 2, 3

[13] M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *ICCV*, 2015. 1, 2, 3, 4

[14] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012. 1, 3, 5

[15] R. Gopalan, R. Li, and R. Chellappa. Unsupervised adaptation across domain shifts by generating intermediate data representations. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 36(11), 2014. 1

[16] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In *ECCV*, 2016. 3

[17] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(1):723–773, 2012. 2

[18] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 2

[19] B. F. Klare, S. S. Bucak, A. K. Jain, and T. Akgul. Towards automated caricature recognition. In *ICB*, 2012. 1

[20] Y. Li, N. Wang, J. Liu, and X. Hou. Demystifying neural style transfer. *CoRR*, arXiv:1701.01036, 2017. 6

[21] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015. 1, 2, 3

[22] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer feature learning with joint distribution adaptation. In *ICCV*, 2013. 1, 5

[23] M. Long, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. *CoRR*, arXiv:1605.06636, 2016. 1, 2, 3, 5

[24] S. J. Pan, J. T. Tsang, Ivor W.and Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *Transactions on Neural Networks*, 22(2):199 – 210, 2011. 1, 2, 4, 5

[25] S. Saxena and J. Verbeek. Heterogeneous face recognition with cnns. In *ECCV Workshop on Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV)*, 2016. 1

[26] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, 2016. 1, 5, 6

[27] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV Workshop on Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV)*, 2016. 1, 2, 5, 6

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 4

[29] G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of CNN activations. In *ICML*, 2016. 3

[30] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, 2015. 1

[31] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *NIPS Workshop on Adversarial Training, (WAT)*, 2016. 1

[32] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, arXiv:1412.3474, 2014. 1, 2, 5

[33] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017. 1