

Cross Domain Image Matching in Presence of Outliers

Liu, Xin; Khademi, Seyran; van Gemert, Jan C.

DOI

[10.1109/ICCVW.2019.00406](https://doi.org/10.1109/ICCVW.2019.00406)

Publication date

2019

Document Version

Accepted author manuscript

Published in

CCV workshop on Transferring and Adapting Source Knowledge in Computer Vision

Citation (APA)

Liu, X., Khademi, S., & van Gemert, J. C. (2019). Cross Domain Image Matching in Presence of Outliers. In *CCV workshop on Transferring and Adapting Source Knowledge in Computer Vision* (pp. 3250-3256) <https://doi.org/10.1109/ICCVW.2019.00406>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

Cross Domain Image Matching in Presence of Outliers

Xin Liu Seyran Khademi Jan C. van Gemert
Computer Vision Lab, Delft University of Technology
Delft, The Netherlands

Abstract

Cross domain image matching between image collections from different source and target domains is challenging in times of deep learning due to i) limited variation of image conditions in a training set, ii) lack of paired-image labels during training, iii) the existing of outliers that makes image matching domains not fully overlap. To this end, we propose an end-to-end architecture that can match cross domain images without labels in the target domain and handle non-overlapping domains by outlier detection. We leverage domain adaptation and triplet constraints for training a network capable of learning domain invariant and identity distinguishable representations, and iteratively detecting the outliers with an entropy loss and our proposed weighted MK-MMD. Extensive experimental evidence on Office [17] dataset and our proposed datasets Shape, Pitts-CycleGAN shows that the proposed approach yields state-of-the-art cross domain image matching and outlier detection performance on different benchmarks. The code will be made publicly available.

1. Introduction

Cross domain image matching is about matching two images that are collected from different sources (e.g. photos of the same location but captured in different illuminations, seasons or era). It has wide application value in different areas, with research in location recognition over large time lags [3], e-commerce product image retrieval [8], urban environment image matching for geo-localization [20], etc.

Even using deep feature representation learning, the automated cross domain image matching task remains challenging mainly due to the following difficulties. First, it is difficult to match varying observations of the same location or object, in general. Second, often the paired-image examples from two domains are not available for training neural networks. Third, the image samples in two domains may not fully overlap due to the existing of outlier images, which affects the matching performance if such outliers are not detected.

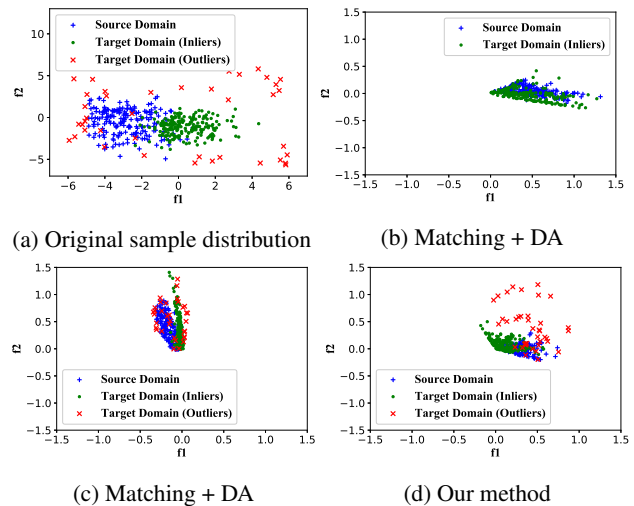


Figure 1: Domain adaptation (DA) and image matching applied on a 2D toy dataset generated with domain shift between source and target domains. (a) Original distribution, (b) no outliers, (c) with outliers, (d) our method. The result of (b) and (c) shows that outliers affect the alignment of source samples and inlier target samples. (c) and (d) show that our outlier detection helps separating the outliers from the aligned source samples and inlier target samples.

In this work, we address the problem of domain adaptation for feature learning in a cross domain matching task when outliers are present. As is common in domain adaptation, we only have labeled image pairs from the source domain, but no labels from the target domain. To resolve the domain disparity between the train and the test data, we are inspired from Siamese network [2] for image matching and domain adaptation used in image classification [13, 18, 22, 23, 26]. We propose a triplet constraints network to learn the domain invariant and identity distinguishable representations of the samples. This is made possible by utilizing the paired-image information from the source domain, a weighted multi-kernel maximum mean discrepancy (weighted MK-MMD) method and an entropy loss. The setting of the problem and experiment results of

our method are depicted on a 2D toy dataset in Figure 1.

To verify our method, we introduce two new synthetic datasets as there are no publicly available datasets for our problem setting. Moreover, we believe outlier-aware algorithms are essential to design practical domain adaptation algorithms as many real data repositories contain irrelevant samples w.r.t. the source domain. In summary, our main contribution is two-fold:

- Joint domain adaptation and outlier detection.
- Two new datasets, *Pits-CycleGAN* dataset and *Shape* dataset, for cross domain image matching.

2. Related work

2.1. Image matching

Feature learning based matching methods become popular due to its improved performance over hand-crafted features (*e.g.* SIFT [15]). Siamese network architectures [2] are among the most popular feature learning networks, especially for pairs comparison tasks. We also adopt Siamese network as part of our framework. The purpose is to learn feature representations to distinguish matching and unmatching pairs in the source domain, which assists the network in learning to match cross domain images. In the cross-domain image matching context, Lin *et al.* [11] investigated a deep Siamese network to learn feature embedding for cross-view image geo-localization. Kong *et al.* [9] applied Siamese architecture to cross domain footprint matching. Tian *et al.* [20] utilized Siamese network for matching the building images from street view and bird’s eye view. Unlike the existing works on cross-domain image matching, we consider labeled paired-image information is only available in the source domain.

2.2. Domain adaptation

Domain adaptation have been researched over recent years in diverse domain classification tasks, in which adversarial learning and statistic methods are main approaches. Ganin *et al.* [4] proposed domain-adversarial training of neural networks with input of labeled source domain data and unlabeled target domain data for classification. In [26], the authors proposed a deep transfer network (DTN), which achieved domain transfer by simultaneously matching both the marginal and the conditional distributions with adopting the empirical maximum mean discrepancy (MMD) [5], which is a nonparametric metric. Venkateswara *et al.* [23] applied MK-MMD [6] to a deep learning framework that can learn hash codes for domain adaptive classification. In this setting MK-MMD loss promotes nonlinear alignment of data, which generates a nonparametric distance in Reproducing Kernel Hilbert Space (RKHS). The distance between two distributions is the distance between their means

in a RKHS. When two data sets belong to the same distribution, their MK-MMD is zero. Based on the successful performance of MK-MMD loss, we also adopt it to adapt different domains, this time for image matching task. This requires the marriage of Siamese network with MK-MMD loss, as we do later in our paper.

2.3. Outlier detection

Much work exists on outlier detection [1, 12, 16, 25]. Chalapathy *et al.* [1] proposed an one-class neural network (OC-NN) encoder-decoder model to detect anomalies. Sabokrou *et al.* [16] also applied the encoder-decoder architecture as part of their network for novelty detection. Zhang *et al.* [25] proposed an adversarial network for partial domain adaptation to deal with outlier classes in the source domain. Their network is for classification task, and they do not have the assumption that outliers originate from low-density distribution. Instead, we are inspired by the work of Liu *et al.* [12] which uses a kernel-based method to learn, jointly, a large margin one-class classifier and a soft label assignment for inliers and outliers. Using the soft label assignment, we implement outlier detection with cross domain image matching in an iterative sample reweighting way.

3. Domain adaptive image matching

3.1. Siamese loss

We introduce our proposal for domain adaptation for image matching task once labeled data is not available in the target domain. Let X_s denote the source domain image set. A pair of images $x_i, x_j \in X_s$ are used as input to part of our network, as shown in Figure 2. x_i, x_j can be a matching pair or an unmatching pair. The objective is to automatically learn a feature representation, $f(\cdot)$, that effectively maps the input x_i, x_j to a feature space, in which matching pairs are close to each other and unmatching pairs are far apart. We employ the contrastive loss as introduced in [7]:

$$L(x_i, x_j, y) = \frac{1}{2}yD^2 + \frac{1}{2}(1-y)\{\max(0, m-D)\}^2, \quad (1)$$

where $y \in \{0, 1\}$ indicates unmatching pairs with $y = 0$ and matching pairs with $y = 1$, D is the Euclidean distance between the two feature vectors $f(x_i)$ and $f(x_j)$, and m is the margin parameter acting as threshold to separate matching and unmatching pairs.

3.2. Domain adaptation loss

It is known that in deep CNNs, the feature representations transition from generic to task-specific as one goes up from bottom layers to other layers [24]. Compared to the convolution layers *conv1* to *conv5*, the fully connected layers are more task-specific and need to be adapted before they can be transferred [23].

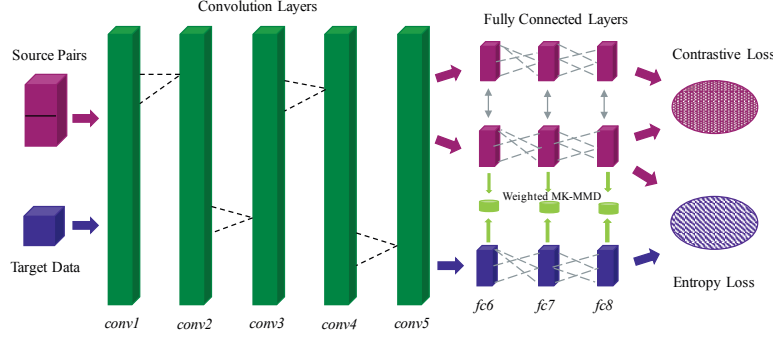


Figure 2: The network for cross domain image matching and outlier detection. The contrastive loss makes the network to learn paired-image information from the source. The weighted MK-MMD loss trains the network to learn transferable features between the source and the inliers of the target. The entropy loss helps distinguish inliers and outliers in the target domain.

Accordingly, our approach attempts to minimize the MK-MMD loss to reduce the domain disparity between the source and target feature representations for fully connected layers, $\mathcal{F} = \{fc6, fc7, fc8\}$. The multi-layer MK-MMD loss is given by,

$$\mathcal{M}(u_s, u_t) = \sum_{l \in \mathcal{F}} d_k^2(u_s^l, u_t^l), \quad (2)$$

where, $u_s^l = \{\mathbf{u}_i^{s,l}\}_{i=1}^{n_s}$ and $u_t^l = \{\mathbf{u}_i^{t,l}\}_{i=1}^{n_t}$ are the set of output representations for the source and target data at layer l , $\mathbf{u}_i^{*,l}$ is the output representation of input image $\mathbf{x}_i^{*,l}$ for the l^{th} layer. The MK-MMD measure $d_k^2(\cdot)$ is the multi-kernel maximum mean discrepancy between the source and target representations [6]. For a nonlinear mapping $\phi(\cdot)$ associated with a reproducing kernel Hilbert space \mathcal{H}_k and kernel $k(\cdot)$, where $k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$, the MK-MMD is defined as,

$$d_k^2(u_s^l, u_t^l) = \|\mathbb{E}[\phi(\mathbf{u}^{s,l})] - \mathbb{E}[\phi(\mathbf{u}^{t,l})]\|_{\mathcal{H}_k}. \quad (3)$$

The characteristic kernel $k(\cdot)$, is determined as a convex combination of κ PSD kernels, $\{k_m\}_{m=1}^{\kappa}$, $K := \{k : k = \sum_{m=1}^{\kappa} \beta_m k_m, \sum_{m=1}^{\kappa} \beta_m = 1, \beta_m \geq 0, \forall m\}$. In particular, we follow [14] and set the kernel weights as $\beta_m = 1/\kappa$.

4. Proposed method: Outlier-aware domain adaptive matching

The task is to match images with the same content but from different domains where the outliers are present in the target domain. We assume that in the source domain there are sufficient labeled image pairs and in the target domain low-density outliers are present. As in conventional domain adaptation setting labeled data is not available in the target domain. We propose a deep triplet network which is comprised of three instances of the same feed-forward network with shared parameters, as shown in Figure 2.

4.1. Importance weighted domain adaptation

In our implementation, the MK-MMD loss in subsection 3.2 is calculated over every batch of data points during the back-propagation. Let n (even) be the number of source data points $u_s := \{\mathbf{u}_i^s\}_{i=1}^n$ and the number of target data points $u_t := \{\mathbf{u}_i^t\}_{i=1}^n$ in the batch. Then, the MK-MMD can be defined over a set of 4 data points $\mathbf{z}_i = [\mathbf{u}_{2i-1}^s, \mathbf{u}_{2i}^s, \mathbf{u}_{2i-1}^t, \mathbf{u}_{2i}^t]$, $\forall i \in \{1, 2, \dots, n/2\}$. Thus, the MK-MMD is given by,

$$d_k^2(u_s, u_t) = \sum_{m=1}^{\kappa} \beta_m \frac{1}{n/2} \sum_{i=1}^{n/2} h_m(\mathbf{z}_i), \quad (4)$$

where, κ is the number of kernels and $\beta_m = 1/\kappa$ is the weight for each kernel. And we can expand $h_m(\cdot)$ as,

$$h_m(\mathbf{z}_i) = k_m(\mathbf{u}_{2i-1}^s, \mathbf{u}_{2i}^s) + k_m(\mathbf{u}_{2i-1}^t, \mathbf{u}_{2i}^t) - k_m(\mathbf{u}_{2i-1}^s, \mathbf{u}_{2i}^t) - k_m(\mathbf{u}_{2i-1}^t, \mathbf{u}_{2i}^s), \quad (5)$$

in which, the kernel is $k_m(\mathbf{x}, \mathbf{y}) = \exp(-\frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{\sigma_m})$.

With equations 4 and 5, we can interpret that in the minimum calculation unit ($h_m(\mathbf{z}_i)$), two target domain images contribute to MK-MMD loss calculation. When there are outliers in the target domain, we only want the inliers to contribute to the calculation, but not the outliers. Therefore, we could assign the target samples with weights w_i as 1 for inliers, and 0 for outliers. Because we have no ground truth labels, we can only treat the weights as the probability of the target samples to be inliers. Hence, we can introduce the weighted MK-MMD as,

$$d_{w_k}^2(u_s, u_t) = \sum_{m=1}^{\kappa} \beta_m \frac{1}{n/2} \sum_{i=1}^{n/2} w_{2i-1} w_{2i} h_m(\mathbf{z}_i), \quad (6)$$

where, w_{2i-1} and w_{2i} are the weights of the target data points \mathbf{u}_{2i-1}^t and \mathbf{u}_{2i}^t in $h_m(\mathbf{z}_i)$ respectively, and $w_{2i-1}, w_{2i} \in [0, 1]$. We will explain how to obtain the weight for each target domain sample in next subsection.

4.2. Outlier detection

Since the inlier-outlier label is not available, we implement an entropy loss to iteratively reassign target domain sample probability of being an inlier, which provides the weights for the weighted MK-MMD.

We use the similarity measure $\langle \mathbf{u}_i, \mathbf{u}_j \rangle$ to learn discriminative inlier-outlier information for the target domain data. We define three classes of reference data u_r for similarity measure, the source domain class \mathbf{u}^1 , the pseudo inlier class \mathbf{u}^2 and the pseudo outlier class \mathbf{u}^3 . An ideal target output \mathbf{u}_i^t needs to be similar to many of the outputs from one of the classes, $\{\mathbf{u}_k^c\}_{k=1}^K$. We assume K data points for every class $c \in \{1, 2, 3\}$ and \mathbf{u}_k^c is the k^{th} output from class c . Then the probability measure for each target sample can be outlined as,

$$p_{ic} = \frac{\sum_{k=1}^K \exp(\mathbf{u}_i^{t\top} \mathbf{u}_k^c)}{\sum_{c=1}^C \sum_{k=1}^K \exp(\mathbf{u}_i^{t\top} \mathbf{u}_k^c)}, \quad (7)$$

where, p_{ic} is the probability that a target domain sample x_i^t is assigned to category c . When the sample output is similar to one category only, the probability vector $\mathbf{p}_i = [p_{i1}, \dots, p_{iC}]^\top$ tends to be a one-hot vector. A one-hot vector can be viewed as a low entropy realization of \mathbf{p}_i . Thus, we introduce a loss to capture the entropy of the probability vectors. The entropy loss can be given by,

$$S(u_r, u_t) = -\frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{c=1}^C p_{ic} \log(p_{ic}). \quad (8)$$

In subsection 4.1, we discussed the weighted MK-MMD loss with weights w_{2i-1} and w_{2i} . With the sample probabilities of target domain data calculated from equation 7, the weights are calculated as,

$$w_i = \begin{cases} \frac{p_{i1} + p_{i2}}{p_{i1} + p_{i2} + p_{i3}} & \text{if } x_i^t \text{ is classified as source} \\ \frac{p_{i2}}{p_{i1} + p_{i2} + p_{i3}} & \text{if } x_i^t \text{ is classified as others} \end{cases} \quad (9)$$

If a target domain sample is classified as "source", then it has a high probability of being an inlier, and therefore should contribute more to reducing the domain disparity. So we calculate the weight of such a target domain sample with the sum of p_{i1} and p_{i2} .

Algorithm We iteratively update the target domain data weights after each epoch during training, which works together with domain adaptation for guiding and correcting the detection of outliers and inliers.

The proposed algorithm for outlier detection is showed in the following. The proposed method is built upon the intuitive assumption that outliers originate from low-density distribution. Thus, we can assume that the ratio of outliers to all the target domain data is no more than 50%.

Algorithm 1

Input: source domain and target domain training data

Output: target domain training data probabilities

- 1: **Initialization** $i = 0$, calculate the average Euclidean distance of each target domain training sample between all the source domain training samples, sort the distances in ascending order and initialize target domain training samples' weights according to the sorted distances, $x_i \in$ first half: $w_i = 0.7$ (pseudo inlier class), $x_i \in$ second half: $w_i = 0.3$ (pseudo outlier class). Inlier class consists of source domain training data, which has the same number of samples with pseudo inlier and pseudo outlier classes.
 - 2: **Repeat:**
 - 3: $i = i + 1$
 - 4: make new mini batches
 - 5: minimize the overall loss function objective (10)
 - 6: update the samples' weights by equation 7 and 9
 - 7: update the sets of pseudo inlier class and pseudo outlier class
 - 8: **Until** target samples' probabilities are unchanged or training time ends
-

4.3. Overall objective

We propose a model for cross domain image matching and outlier detection, which incorporates learning image matching information from source domain (1), weighted domain adaptation between the source and the target (6) and outlier detection (8) in a deep CNN. The overall objective is given by:

$$\min_u J = L(u_s) + \gamma M_w(u_s, u_t) + \eta S(u_r, u_t), \quad (10)$$

where, $u := \{u_s \cup u_t\}$ and (γ, η) control the importance of domain adaptation (6) and entropy loss (8) respectively.

5. Experiments

5.1. Datasets

There are no publicly available datasets for our task. Therefore, we propose two datasets for evaluation. Sample images from the three datasets are shown in Figure 3.

Shape is one of the synthetic datasets we generate. It contains 60k source domain images, 30k target domain images (including 2800 outliers). The outlier images are made up of single alphabets or digits. The source domain and inlier images are combinations of two geometric shapes, drawn with black solid lines and colored dot lines, respectively. We define two images are a matching pair if the combination of shapes is the same.

Pitts-CycleGAN is the other synthetic dataset, which contains 204k Pittsburgh Google Street View images from

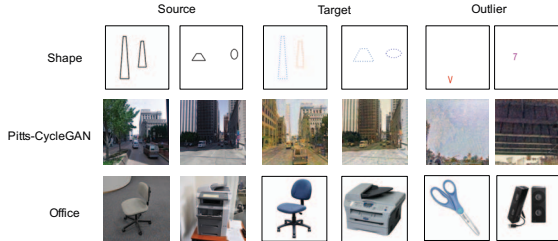


Figure 3: Examples from *Shape*, *Pitts-CycleGAN* and *Office* sets.

Pittsburgh dataset [21] as the source domain, and 157k target domain images (including 52k outliers) generated by applying CycleGAN [27] to the Pittsburgh images. So the target domain images are in a painting style. The outliers are sky images or city views not containing any useful landmark information.

Office [17] consists of 3 domains, *Amazon*, *Dslr*, *Webcam*. We choose *Dslr* as source domain and *Amazon* as target domain. We make pairs with images from the same category. The outliers come from two randomly chosen categories ('speaker', 'scissors') out of the 31 categories.

5.2. Implementation details

For our triplet network, the three sub-networks share the same architecture and weights. Pre-trained AlexNet [10] is used for the sub-networks. We finetune the weights of *conv4-conv5*, *fc6*, *fc7*, *fc8*. For the weighted MK-MMD, we use a Gaussian kernel with a bandwidth σ given by the median of the pairwise distances in the training data. To incorporate the multi-kernel, we vary the bandwidth $\sigma_m \in [2^{-8}\sigma, 2^8\sigma]$ with multiplicative factor of 2 [23]. For performance evaluation, we sort the Euclidean distance between the query and all the gallery features (L2-normalized) to obtain the ranking result. Moreover, we employ the standard metric mean average precision (MAP).

5.3. Baseline methods

There are no available baselines to directly compare with our method, thus, we separate our experiments to research on domain adaptive image matching 5.4 and effectiveness of outlier detection 5.5.

In the experiment on domain adaptive image matching, we assume no outliers exist in the target domain. Our method is to jointly learn the contrastive loss $L(u_s)$ and MK-MMD loss $M(u_s, u_t)$. It is trained with pairs from the source domain and images from the target domain, we call it *SiameseDA*.

For evaluating the effectiveness of outlier detection, the target domain contains outliers. Our method is called *DA+OutlierDetection*, which learns on the objective 10.

The baselines for each experiment are shown in Table 1.

Baseline	Experiment
	Domain adaptive image matching
<i>SIFT + Fisher Vector</i> [15, 19]	trained on the source domain data
<i>Siamese</i> network [2]	trained on the source domain image pairs
	Effectiveness of outlier detection
<i>SiameseDA</i> (upper bound)	trained without outliers
<i>SiameseDAOut</i> (lower bound)	<i>SiameseDA</i> trained with outliers

Table 1: Baseline methods for our experiments.

5.4. Domain adaptive image matching

In this section, we assume the target domain does not contain outliers. We explore if applying domain adaptation improves the performance of cross domain image matching. In this case, the learning objective is

$$\min_u J = L(u_s) + \gamma M(u_s, u_t), \quad (11)$$

where, the MK-MMD loss term $M(u_s, u_t)$ is the un-weighted version as explained in subsection 3.2.

The MAP results are given in Table 2. Our method consistently outperforms the baselines across all the datasets. With applying MK-MMD loss for domain adaptation, the performance of matching $S \rightarrow S$ decreases comparing to that of Siamese method. This is within our expectation since the network may need to learn less from the source domain to be domain adaptive. Moreover, it is worth to notice that our method also improves the in-domain image matching ($T \rightarrow T$) of the target domain.

5.5. Effectiveness of outlier detection

Here we assume the target domain contains outliers, which is to show if the presence of outliers reduces the accuracy of cross domain image matching, and our method could improve it.

The performance of our method (*DA+OutlierDetection*), upper bound (*SiameseDA*) and lower bound (*SiameseDAOut*) are given in Table 3. In terms of testing, we only take the classified inliers in the query set in calculation. From Table 3 we can see, our method outperforms the lower bound for all the three datasets, but is not better than the upper bound (except for Pitts-CycleGAN) as expected. It shows that the presence of outliers reduces the accuracy of cross domain image matching, and our method helps improve the performance in this case.

In Figure 4, we also show the retrieval performance in terms of the trade-off between precision and recall at different thresholds on our three datasets. The interpolated average precision is used for the precision-recall curves. We can see that our method gains over the lower bound method.

Impact of outlier proportion We also report the F_1 -score to measure the performance of outlier detection of

Method	Shape			Office			Pitts-CycleGAN		
	$T \rightarrow S$	$S \rightarrow S$	$T \rightarrow T$	$T \rightarrow S$	$S \rightarrow S$	$T \rightarrow T$	$T \rightarrow S$	$S \rightarrow S$	$T \rightarrow T$
SIFT + Fisher Vector	2.5 ± 0.4	3.6 ± 0.3	3.4 ± 0.3	3.5 ± 0.2	12.0 ± 0.5	3.5 ± 0.1	0.04	0.8 ± 0.05	0.3 ± 0.03
Siamese	8.3 ± 0.1	95.0 ± 0.2	31.7 ± 0.6	10.7 ± 0.5	99.2 ± 0.2	77.2 ± 0.3	0.2 ± 0.01	81.3 ± 0.3	60.6 ± 0.5
SiameseDA	26.4 ± 0.2	53.1 ± 0.1	46.2 ± 0.1	29.1 ± 0.1	99.7 ± 0.1	77.5 ± 0.2	0.4 ± 0.01	80.4 ± 0.1	59.5 ± 0.1

Table 2: MAP performance for cross domain image matching and in-domain image matching experiments on three datasets. T means target domain, S means source domain. $T \rightarrow S$ implies matching target domain images to source domain images, similar for $S \rightarrow S$, $T \rightarrow T$. Our method **SiameseDA** outperforms the baselines across all the datasets.

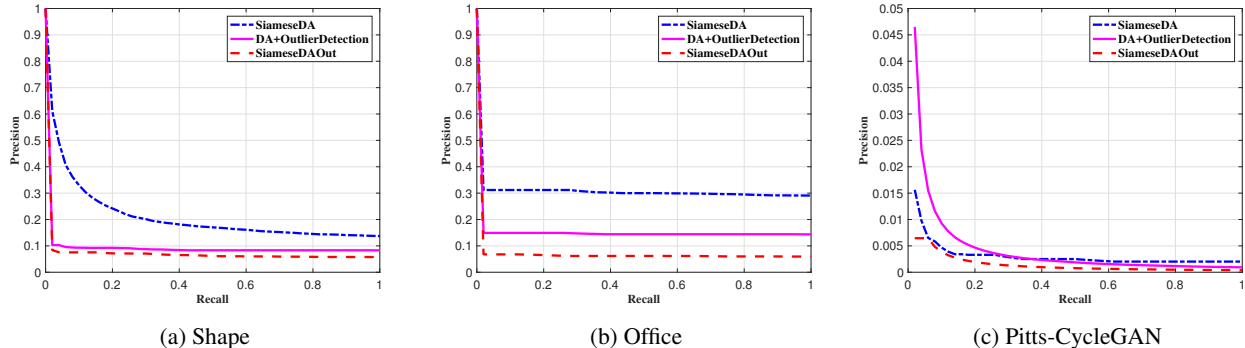


Figure 4: Precision-Recall results of our method **DA+OutlierDetection**, SiameseDA and SiameseDAOut for the experiment of cross domain image matching with outlier detection on the three datasets. Our method gains over the lower bound method.

Method ($T \rightarrow S$)	Shape	Office	Pitts-CycleGAN
SiameseDA	26.4 ± 0.2	29.1 ± 0.1	0.4 ± 0.01
DA+OutlierDetection	11.9 ± 0.1	15.9 ± 0.2	1.1 ± 0.03
SiameseDAOut	5.4 ± 0.1	6.8 ± 0.1	0.2 ± 0.01

Table 3: MAP performance for cross domain image matching with outlier detection on our three datasets. The proportion of outliers is 10%. Our method **DA+OutlierDetection** outperforms the lower bound, but does not surpass the upper bound.

our method. Figure 5 shows the F_1 -score of our method as a function of the portion of outlier samples for the three datasets. As can be seen, with the increase in the number of outliers, our method operates consistently robust.

It is important to notice the limitation of our method, which classifies some inlier samples as outliers during training. This is mainly caused by the way of initializing the probabilities of the target domain training data.

6. Conclusion

We have proposed a network that is trained for cross domain image matching with outlier detection in an end-to-end manner. The two main parts of our approach are (i) domain adaptive image matching subnetwork with contrastive loss and weighted MK-MMD loss, (ii) outlier de-

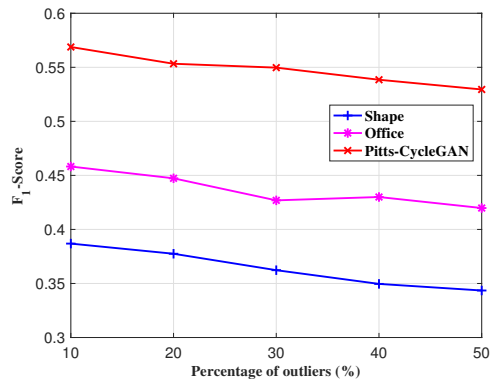


Figure 5: F_1 -scores for outlier detection on three datasets with different outlier proportion in the target domain. Our method is consistently robust.

tection with entropy loss by updating the probability of target domain data during training. The results on several datasets demonstrate that the proposed method is capable of detecting outlier samples and achieving cross domain image matching at the same time. But our method still needs improvement to overcome the problem of wrongly classifying inliers as outliers.

References

- [1] R. Chalapathy, A. K. Menon, and S. Chawla. Anomaly detection using one-class neural networks. *arXiv:1802.06360*, 2018. [2](#)
- [2] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern*, pages 539–546, 2005. [1](#), [2](#), [5](#)
- [3] B. Fernando, T. Tommasi, and T. Tuytelaars. Location recognition over large time lags. *Computer Vision and Image Understanding*, 139:21–28, 2015. [1](#)
- [4] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:2096–2030, 2016. [2](#)
- [5] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, pages 513–520, 2006. [2](#)
- [6] A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B. K. Sriperumbudur. Optimal kernel choice for large-scale two-sample tests. In *Advances in Neural Information Processing Systems 25*, pages 1205–1213. Curran Associates, Inc., 2012. [2](#), [3](#)
- [7] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR '06*, pages 1735–1742, 2006. [2](#)
- [8] X. Ji, W. Wang, M. Zhang, and Y. Yang. Cross-domain image retrieval with attention modeling. *2017 ACM Multimedia Conference*, 2017. [1](#)
- [9] B. Kong, J. Supancic, D. Ramanan, and C. C. Fowlkes. Cross-domain image matching with deep feature maps. *International Journal of Computer Vision*, 2018. [2](#)
- [10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, pages 1097–1105, 2012. [5](#)
- [11] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays. Learning deep representations for ground-to-aerial geolocation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. [2](#)
- [12] W. Liu, G. Hua, and J. R. Smith. Unsupervised one-class learning for automatic outlier removal. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3826–3833, 2014. [2](#)
- [13] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning*, pages 97–105, 2015. [1](#)
- [14] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 136–144, 2016. [3](#)
- [15] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision, ICCV '99*, pages 1150–1157, 1999. [2](#), [5](#)
- [16] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli. Adversarially learned one-class classifier for novelty detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3379–3388, 2018. [2](#)
- [17] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Proceedings of the 11th European Conference on Computer Vision: Part IV, ECCV'10*, pages 213–226, 2010. [1](#), [5](#)
- [18] K. Saito, Y. Ushiku, and T. Harada. Asymmetric tri-training for unsupervised domain adaptation. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2988–2997, 2017. [1](#)
- [19] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *Int. J. Comput. Vision*, 105(3):222–245, 2013. [5](#)
- [20] Y. Tian, C. Chen, and M. Shah. Cross-view image matching for geo-localization in urban environments. In *CVPR*, 2017. [1](#), [2](#)
- [21] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi. Visual place recognition with repetitive structures. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 883–890, 2013. [5](#)
- [22] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971, 2017. [1](#)
- [23] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5385–5394, 2017. [1](#), [2](#), [5](#)
- [24] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 3320–3328, 2014. [2](#)
- [25] J. Zhang, Z. Ding, W. Li, and P. Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. [2](#)
- [26] X. Zhang, F. X. Yu, S. Chang, and S. Wang. Deep transfer network: Unsupervised domain adaptation. *arXiv:1503.00591*, 2015. [1](#), [2](#)
- [27] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017. [5](#)