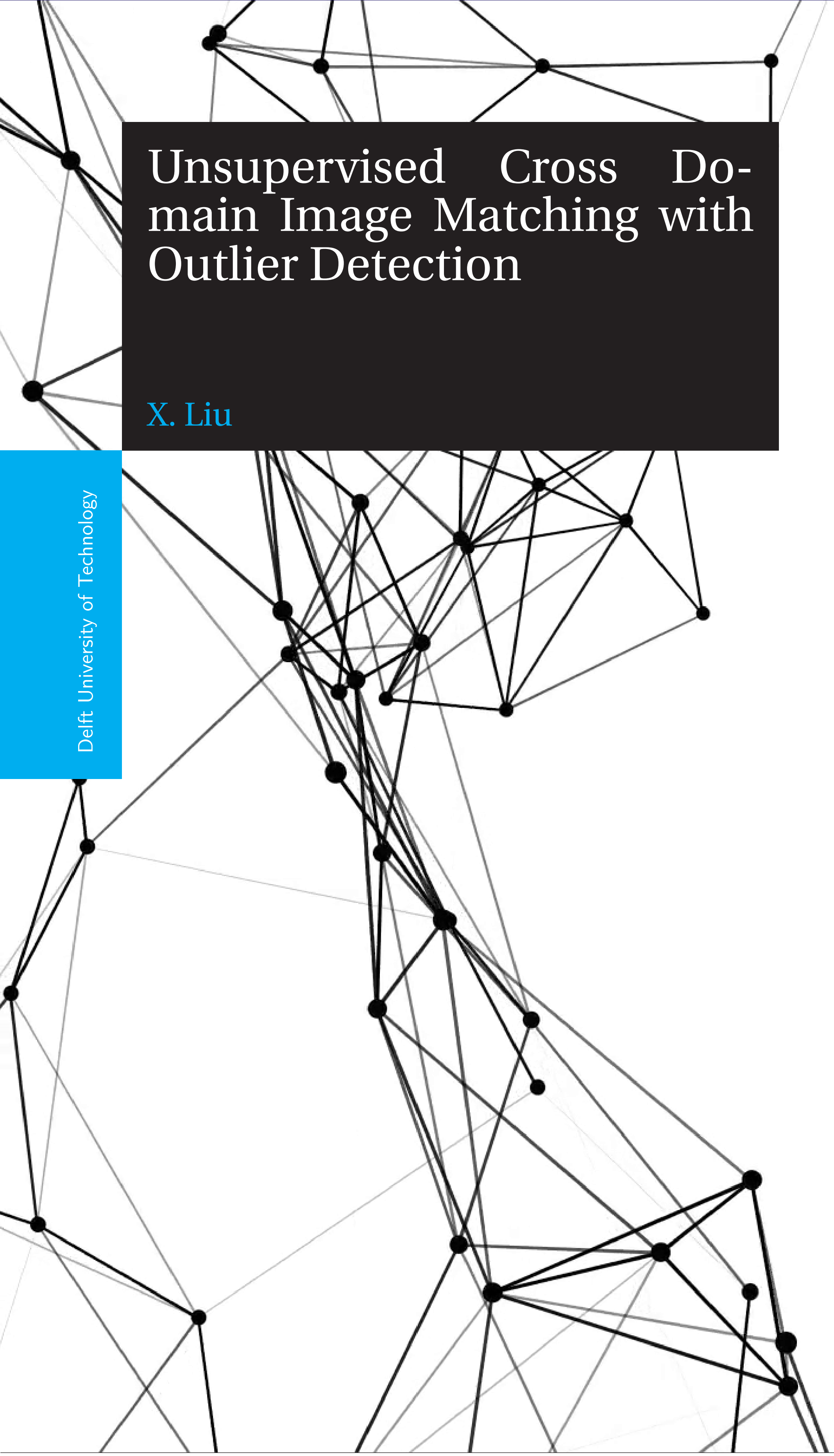


Unsupervised Cross Domain Image Matching with Outlier Detection

X. Liu

Delft University of Technology

TU Delft Delft University of Technology

Challenge the future

Unsupervised Cross Domain Image Matching with Outlier Detection

by

Xin Liu

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday August 31, 2018 at 10:00 AM.

Student number:	4623886	
Project duration:	November 2017 – August 2018	
Thesis committee:	Prof. dr. M. J. T. Reinders,	TU Delft, chair
	Dr. J. C. van Gemert,	TU Delft, supervisor
	Dr. S. Khademi,	TU Delft, daily supervisor
	Dr. ir. S. E. Verwer,	TU Delft, committee member

Preface

The report documents the findings of my master thesis work. The main content of the report is the scientific paper. It includes the motivation, methodology, implementation, and results of this research project. The chapters after the scientific paper introduce the background knowledge needed to understand this work.

This work would not have been possible without Dr. ir. Jan van Gemert and Dr. Seyran Khademi who were kind, considerate and supportive. They provided me with invaluable guidance while still affording me the freedom to test my own ideas. I feel truly grateful to have them as my supervisors.

Many thanks to all the wonderful friends I met in Delft. I was fortunate to be accompanied by them on my journey towards an MSc.

Most importantly, none of these would have been possible without the support of my parents. I would like to thank them for always being by my side no matter what.

Xin Liu
Delft, August 2018

Contents

1

Scientific Paper

Unsupervised Cross Domain Image Matching with Outlier Detection

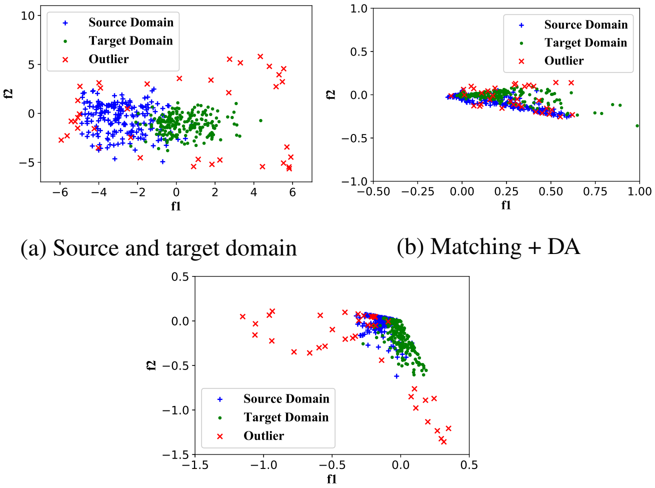
Xin Liu
Delft University of Technology
Delft, The Netherlands
xliu-16@student.tudelft.nl

Abstract

This work proposes a method for matching images from different domains in an unsupervised manner, and detecting outlier samples in the target domain at the same time. This matching problem is made difficult by i) the different domain images that are related but under different conditions (e.g. photos of the same location captured in different illuminations), ii) unsupervised settings with paired-image information available only for one of the domains, iii) the existing of outliers that makes the two domains not fully overlap. To this end, we propose an end-to-end architecture that can match cross domain images in an unsupervised manner and handle not fully overlapping domains by outlier detection. Our architecture is composed of three subnetworks, two of which are fed with pairs of source images to learn the "match" information. The other subnetwork is fed with target images, and works together with the other two subnetworks to learn domain invariant representations of the source samples and the target inlier samples by applying a weighted multi-kernel Maximum Mean Discrepancy (weighted MK-MMD). We propose the weighted MK-MMD, together with an entropy loss, for outlier detection. The entropy loss iteratively outputs the probability of a target sample to be an inlier during training. And the probabilities are used as weights in our weighted MK-MMD for aligning only the target inlier samples with the source samples. Extensive experimental evidence on Office [26] 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.

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 or seasons). It has wide application value in different areas, with research in location recognition over large time lags [4], e-commerce product image retrieval using images taken by smart phone [11], urban environment image matching for geo-localization [35], etc. But cross domain image matching is still a challenging problem when the paired-image information in one of the two domains is not available during training. And the two domains even do not fully overlap because of the existing of some outliers in at least one of the two domains, which affects the matching performance if not being detected.



(a) Source and target domain

(b) Matching + DA

(c) Matching + DA + Outlier Detection

Figure 1: Proposed method applied on a 2D toy dataset. (a) Source and target domain data distribution; (b) Source and Target domain data distribution after only applying domain adaptation (DA) and image matching; (c) Source and Target domain data distribution after applying outlier detection based on (b). Comparing (b) and (c), it shows that outlier detection helps separate the outliers from the aligned source samples and inlier target samples.

The standard method for image matching consists of two major steps. First, a feature vector is extracted to represent each image, including the query image and images in the database. Second, the similarity (e.g. Euclidean distance similarity) of the query image and the database images is computed based on the feature vectors. Then top-K similar images are returned to the user. State of art cross domain image matching systems focus on improving feature extraction, which falls into two main categories. One is the handcrafted way of extracting feature descriptors (SIFT [20], LIOP [43], Patch-CKN [23]) or representations (Fisher Vectors [24], DeCAF [3]) of images and matching with different metrics. These methods focused on designing feature extractors suitable for each domain, which yields domain invariant descriptors that can then be directly compared. The other category is based on deep learning models, which utilizes the deep convolutional neural networks (CNNs) as feature extractors. This method has proven incredibly effective in various visual tasks such as object classification [14, 21, 31, 32], scene recognition [52] and object detection [6]. We also utilize CNNs as our feature extractors to learn effective features for our matching task.

The previous works using deep learning frameworks for cross domain image matching are in supervised settings. Those research utilized the available information that image A from the source domain matches with image B from the target domain during training [13, 11, 35]. And they assume that the two domains fully overlap. But this assumption cannot be guaranteed in many cases when the datasets are collected from various sources. In our work, we perform unsupervised cross domain image matching which has the benefit of not needing expensive labeled samples, and handle not fully overlapping domains by outlier detection at the same time. The setting of our problem is shown in Figure 1. We only want to correctly align the inliers in the target domain with the source domain data, and reject those outliers in the target domain.

Our inspiration for matching the cross domain images in an unsupervised manner comes from Siamese network [2] for image matching and unsupervised domain adaptation used in image classification [38, 27, 41, 50, 18]. There are two main methods that have been investigated in unsupervised domain adaptation classification, which are domain adversarial network [38, 5, 29, 12] and statistic domain adaptive method [41, 50, 40, 18]. In short, domain adversarial method makes the network cannot distinguish whether one image is from the source domain or the target domain. Statistic method (e.g. MK-MMD [9]) compares the sample distribution in a latent space. In our work, we choose to investigate the statistic domain adaptive method to form a loss function, which could leverage the sample distributions for the matching of non-fully overlapping domains.

Outliers detection [44, 47] is the process of identifying the new or unexplained set of data to determine if they are within the norm (i.e., inliers) or outside of it (i.e., outliers). Outliers refer to the unusual observations that do not occur regularly or are simply different from the others. Outliers detection can be portrayed in the context of one-class classification, which aims to build classification models. For example, one-class support vector machines [30, 34] are widely used, effective unsupervised techniques to identify outliers. The adversarially learned one-class neural networks [25, 1] also become popular in recent years for outliers detection. It applies an encoder-decoder network architecture. For our problem, it is not easy to apply these outlier detection methods directly. Our approach needs to optimize the learning objective which jointly learns domain adaptive image matching and the distinguishing of inliers and outliers in the target domain. So for matching the pairs of images, it is hard to achieve with, for example, an encoder-decoder network.

Several works have focused on cross domain image matching, yet few have analyzed the impact of existing outliers in the target domain dataset on unsupervised image matching performance. And few public datasets are available for unsupervised cross domain image matching. For this purpose, we introduce two new datasets, a *Shape* toy dataset and *Pitts-CycleGAN* dataset. The *Shape* toy dataset consists of basic geometric shapes, such as triangle, square, circle, etc. The source domain of the toy dataset is shapes with solid lines, where the two images contain shapes from same categories are a pair. And the target domain contains shapes with colored dash line. The outliers in *Shape* dataset are images with single digits or alphabets. The *Pitts-CycleGAN* dataset consists of source domain from Pittsburgh (Pitts250k) [36] and target domain generated by applying CycleGAN [53] method to Pittsburgh (Pitts250k). The outliers in *Pitts-CycleGAN* dataset are images of random sky views or meaningless city views. In a nutshell, our main contributions are three-fold:

- We propose a loss function to train the deep network with the following components: (i) supervised contrastive loss for labeled source data, which helps the network learn how two images are matched; (ii) unsupervised entropy loss for unlabeled target data, which ensures the distinction between inliers and ourliers; (iii) a loss based on MK-MMD [9], which is to learn transferable features within the layers of the network to minimize the distribution difference between the source and the inlier part of target domain.
- We introduce two datasets, *Pits-CycleGAN* dataset and *Shape* toy dataset, for our research problem and evaluating our method.

The research goal of our work is to investigate the unsupervised cross domain image matching, where the query target domain does not fully overlap with the source domain because of the outliers. Thus, the fundamental task of this work is matching cross domain images in an unsupervised manner and rejecting the outliers at the same time.

2. Related Work

Our unsupervised cross domain image matching with outlier detecting problem belongs to content based image matching. Many research works have been proposed to improve the matching performance from the feature extraction and similarity measurement perspectives. However, to our knowledge, none of existing research works considered the impact of outliers in query target dataset on the matching performance.

Image Matching Feature learning based matching methods become popular due to its superior performance over hand-crafted features (e.g. SIFT [20]), for many computer vision tasks such as image matching and classification. Among the feature learning based matching methods, a so-called ”Siamese network” architectures [2] is popular. It integrates feature extraction and comparison in a single differentiable model that can be optimized end-to-end. Past works have demonstrated that Siamese networks learn good features for person re-identification, face recognition, and stereo matching [49, 22, 45]. Many other research works also utilized Siamese architectures for image matching or retrieval tasks, especially for pairs comparison tasks. Lin *et al.* [15] investigated the deep learning method for cross-view image geo-localization. A deep Siamese network was used to learn feature embedding for image matching. Kong *et al.* [13] applied Siamese architecture to cross domain footprint matching. The network was trained with paired footprint images from two different domains. Tian *et al.* [35] utilized Siamese network for building matching, where the buildings come from two domains of street view images and bird's eye view images. However, our work differs from the existing cross domain image matching approaches since we do not have the paired information of images from two domains to train the network directly. In addition, we also consider the outliers existing in the query target domain. To solve this problem, we propose to apply domain adaptation methods in our unsupervised cross domain image matching and perform outliers distinction at the same time.

Domain Adaptation Domain Adaptation have been researched over recent years in unsupervised diverse domain classification tasks. Adversarial learning, GAN-based and statistic domain adaptation methods are main approaches that have been utilized in domain adaptation classification tasks. Ganin *et al.* [5] proposed domain-adversarial training of neural networks, which are trained on labeled data from the source domain and unlabeled data from the target domain for classification. Tzeng *et al.* [39] proposed a framework which combines discriminative modeling, untied weight sharing, and a GAN [7] loss for unsupervised adaptation classification problems. They called it Adversarial Discriminative Domain Adaptation (ADDA). Sankaranarayanan *et al.* [29] provided an adversarial image generation approach for unsupervised domain adaptation that directly learns a joint feature space in which the distance between source and target distributions is minimized. In [51], the authors proposed a domain adaptation method called 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) [8] nonparametric metric. Venkateswara *et al.* [41] investigated a novel deep learning framework that can exploit labeled source data and unlabeled target data to learn hash codes for classification. In their work, they applied MK-MMD [9], which seeks to learn transferable features to minimize the distribution difference between the source and target domains. Based on these successful applications of MK-MMD [9], we also adopt it to adapt different domains and perform instance level reweighting for outliers detection. It is easy to change MK-MMD [9] metric into a weighted form to deal with outliers.

Outlier Detection For outlier detection, there are many existing works, such as Liu *etal.* [17] proposed a kernel-based method jointly learning a large margin one-class classifier and a soft label assignment for inliers and outliers. Chalapathy *etal.* [1] proposed an one-class neural network (OC-NN) model to detect anomalies in complex data sets. The model is an encoder-decoder architecture. Sabokrou *etal.* [25] investigated the adversarially learned one-class classifier for novelty detection, which also applied encoder-decoder architecture as part of their network. We do not change our architecture into the encoder-decoder manner, instead we inspire by the soft label assignment technique, jointly implement outlier detection and unsupervised cross domain image matching in an iteratively sample-reweighted way.

3. Background on Unsupervised Domain Adaptive Image Matching

Before introducing our proposed method, we first explain the image matching method and domain adaptation approaches that are utilized in our work.

3.1. Image Matching

Our goal is to utilize the paired-image information of source domain to guide the network to learn the ”match” concept. It can help find good feature representations for cross domain images matching.

The Siamese network [2] has been successfully applied to image matching [16, 48], tracking [33] and retrieval [42].

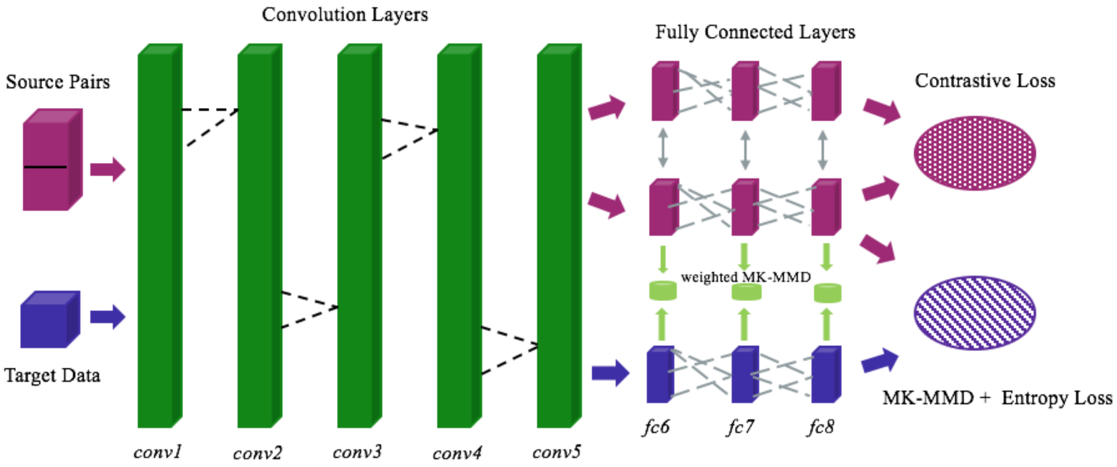


Figure 2: The network for learning unsupervised cross domain image matching and outlier detection. The convolution layers *conv4-conv5* and the fully connected layers *fc6-fc8* are fine tuned to fit specific tasks. The triplet network shares weights at the same layers. The contrastive loss makes the network to learn paired-image information from the source. The MK-MMD loss trains the network to learn transferable features between the source and the target. And the entropy loss helps the network learn to distinct the inlier and the outlier in the target domain.

We also adopt this architecture as part of our network to learn deep representations to distinguish matched and unmatched pairs in source domain. Even this only learns the paired-image information of the source domain, we want to utilize it to help learn the "match" concept between cross domain images. 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 the network, as shown in Figure 2. x_i, x_j can be a matched pair or unmatched 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 matched pairs are close to each other and unmatched pairs are far apart. To train the network towards this goal, the Euclidean distance of the matched pairs in the feature space should be small while the distance of the unmatched pairs should be large. We employ the contrastive loss in the form of [10]:

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

where $y \in \{0, 1\}$ indicates non-matching 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 non-matching pairs.

3.2. Reducing Domain Disparity

Domain adaptation has been studied in deep learning methods with state-of-the-art algorithms [5, 18, 19, 37] in recent years. In a deep CNN, the feature representations transition from generic to task-specific as one goes up from bottom layers to other layers [46]. The convolution layers *conv1* to *conv5* have been shown in [46] to be generic and thus readily transferable, whereas the fully connected layers are more task-specific and need to be adapted before they can be transferred [41]. To make the fully connected layers adaptive, it is possible to apply MK-MMD [9] on these layers. MK-MMD [9] produces a 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 an RKHS. When two data sets belong to the same distribution, their MK-MMD is zero.

Our approach attempts to minimize the MK-MMD [9] loss to reduce the domain disparity between the source and target feature representations for fully connected layers, $\mathcal{F} = \{fc6, fc7, fc8\}$. After applying MK-MMD [9] on the image matching network in section 3.1, we expect the learned paired-image information can be domain adaptive for cross domain image matching. 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, $u_s^l = \{\boldsymbol{u}_i^{s,l}\}_{i=1}^{n_s}$ and $u_t^l = \{\boldsymbol{u}_i^{t,l}\}_{i=1}^{n_t}$ are the set of output representations for the source and target data at layer l, $\boldsymbol{u}_i^{*,l}$ is the output representation of inuput image $\boldsymbol{x}_i^{*,l}$ 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 [9]. For a nonlinear mapping $\phi(\cdot)$ associated with a reproducing kernel Hilbert space $\mathcal{H}_k$ and kernel $k(\cdot)$, where $k(\boldsymbol{x}, \boldsymbol{y}) =< \phi(\boldsymbol{x}, \boldsymbol{y}) >$, the MMD is defined as,

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

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 set the kernel weights as $\beta_m = 1/\kappa$ according to [19].

4. Proposed Approach

This section presents the proposed method in details. In unsupervised cross domain image matching with outliers in the target domain, we are given a source domain data and a target domain data containing outliers. The source domain data denoted as $X_s \in \mathbb{R}^{D \times n_s}$ are drawn from distribution $p_s(x)$ and the target domain data donated as $X_t \in \mathbb{R}^{D \times n_t}$ are drawn from distribution $p_t(x)$, where D is the dimension of the data instance, n_s and n_t are number of samples in source and target domain respectively. Our problem focus on the setting which assumes that there are sufficient labeled source domain data and the label indicates if the two images are a matched pair or not, $D_s = \{(x_i^s, x_j^s, y_{ij}^s)\}_{i \neq j}^{n_s}$, $x_i^s, x_j^s \in \mathbb{R}^D$, $y_{ij} = 0$, negative pair, $y_{ij} = 1$, positive pair, and unlabeled target domain data, $D_t = \{(x_i^t)\}_{i=1}^{n_t}$, $x_i^t \in \mathbb{R}^D$, in the training stage. Especially, in the target domain, there are some low-density samples that not belong to any categories of source and target domain, which are called outliers. In this case, the feature spaces are assumed partly same: $\mathcal{X}_s \approx \mathcal{X}_t$ since the target domain space contains some samples that not belong to either feature space. And we have $p_s(y|x) = p_t(y|x_{inlier})$, $p_s(y|x) \neq p_t(y|x_{outlier})$. In addition, due to the domain shift, $p_s(x) \neq p_t(x)$ even when the label spaces between domains are the same.

4.1. Model

We implemented the neural network as a deep triplet network which is comprised of three instances of the same feed-forward network with shared parameters, as shown in Figure 2. The sub-network instance is a CNN which consists of 5 convolution layers *conv1 - conv5* and 3 fully connected layers *fc6 - fc8*. When fed with 3 samples (a pair from source domain, a single image from target domain), the network outputs three different representations. The representations of the pair are fed into *contrastive loss* and the representation of one image from the pair together with the representation of a target image are fed into *entropy loss*. The contrastive loss ensures the learned representations having a large difference between negative pairs, and having a small difference between positive pairs. The entropy loss aligns the target samples with source image and the inliers and outliers in the target domain, which is based on the similarity of their feature representations.

To address the issue of cross domain (domain shift), we align the feature representations of the target domain and the source domain. This is achieved by reducing the domain discrepancy between the source and the target samples feature representations at multiple layers of the network. In the following subsections, we discuss the design of our proposed network in detail.

4.2. Importance Weighted Domain Adaptation

We implement the linear MK-MMD loss according to [9], and apply MK-MMD to the fully connected layers as shown in Figurer 2. The output of i^{th} source data point at layer l is represented as $\boldsymbol{u}_i^s$ and the output of the i^{th} target data point is represented as $\boldsymbol{u}_i^t$. Unlike the conventional MMD loss which is $O(n^2)$, the MK-MMD loss outlined in [9] is $O(n)$ and can be estimated without requiring all the data. The loss is calculated over every batch of data points during the back-propagation. Let n be the number of source data points $u_s := \{\boldsymbol{u}_i^s\}_{i=1}^n$ and the number of target data points $u_t := \{\boldsymbol{u}_i^t\}_{i=1}^n$ in the batch. Equal number of source and target data points is assumed in a batch and n is even. Then, the MK-MMD can be defined over a set of 4 data points $\boldsymbol{z}_i = [\boldsymbol{u}_{2i-1}^s, \boldsymbol{u}_{2i}^s, \boldsymbol{u}_{2i-1}^t, \boldsymbol{u}_{2i}^t]$, $\forall i \in \{1, 2, ..., n/2\}$. Thus, the MK-MMD is given by,

$$\mathcal{M}(u_s, u_t) = \sum_{m=1}^{\kappa} \beta_m \frac{1}{n/2} \sum_{i=1}^{n/2} h_m(\boldsymbol{z}_i), \qquad (4)$$

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

$$\begin{aligned} h_m(\boldsymbol{z}_i) = {} & k_m(\boldsymbol{u}_{2i-1}^s, \boldsymbol{u}_{2i}^s) + k_m(\boldsymbol{u}_{2i-1}^t, \boldsymbol{u}_{2i}^t) \\ & - k_m(\boldsymbol{u}_{2i-1}^s, \boldsymbol{u}_{2i}^t) - k_m(\boldsymbol{u}_{2i}^s, \boldsymbol{u}_{2i-1}^t), \end{aligned} \qquad (5)$$

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

With the MK-MMD loss in form of equation 4, we can interpret that in the minimum calculation unit ($h_m(z_i)$), two target domain images are contributed to the MK-MMD loss for minimizing the difference between the source domain and the target domain. When considering the case that there are outliers in target domain, we can conclude that we only want to correctly align the source domain data with those target data that are in the same categories (inliers), but avoid involving the outliers. We consider the target sample weights w_i for inliers as 1, and for outliers as 0. But in our setting, the outlier-inlier information is not available for the target samples. 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,

$$\mathcal{M}_w(u_s, u_t) = \sum_{m=1}^{\kappa} \beta_m \frac{1}{n/2} \sum_{i=1}^{n/2} w_{2i-1} w_{2i} h_m(\boldsymbol{z}_i), \qquad (6)$$

where, w_{2i-1} and w_{2i} are the weights of the target data points $\boldsymbol{u}^t_{2i-1}$ and $\boldsymbol{u}^t_{2i}$ in $h_m(z_i)$ respectively, and $w_{2i-1}, w_{2i} \in [0,1]$. If we treat all the target domain data the same, with $w_i = 1$, then the outliers would introduce bias to domain adaptation. We will talk about how to obtain the weight to each target domain data point in next subsection.

4.3. Outlier detection

In the target domain, data contains some outliers, but we own no information about which samples are outliers. To solve this problem, we implement an entropy loss to iteratively reassign sample probability of being an inlier, which provides the weights for the weighted MK-MMD loss.

In the absence of target data inlier-outlier information, we use the similarity measure $< \boldsymbol{u}_i, \boldsymbol{u}_j >$, to guide the network to learn discriminative inlier-outlier information for the target data. We define there are three categories of reference data u_r for similarity measure, the source domain data $\boldsymbol{u}^1$, the pseudo inlier data of target domain $\boldsymbol{u}^2$, and the pseudo outlier data of target domain $\boldsymbol{u}^3$. An ideal target output $\boldsymbol{u}^t_i$ needs to be similar to many of the outputs from one of the categories, $\{\boldsymbol{u}^c_k\}^K_{k=1}$. Without loss of generality, we assume K data points for every category c where, $c \in \{1,2,3\}$ and $\boldsymbol{u}^c_k$ is the k^{th} output from category c. Moreover, $\boldsymbol{u}^t_i$ must be dissimilar to most other reference outputs $\boldsymbol{u}^c_k$ belonging to a different category. Enforcing similarity with all the K data points guarantee a robust target data category assignment. Then the probability measure for each target sample can be outlined as,

$$p_{ic} = \frac{\sum_{k=1}^{K} \exp(\boldsymbol{u}^t_i \top \boldsymbol{u}^c_k)}{\sum_{c=1}^{C} \sum_{k=1}^{K} \exp(\boldsymbol{u}^t_i \top \boldsymbol{u}^c_k)}, \tag{7}$$

where, p_{ic} is the probability that input target data point x_i is assigned to category c. The $\exp(\cdot)$ has been introduced for ease of differentiability and the denominator ensures $\sum_c p_{ic} = 1$. When the target data point output is similar to one category only, the probability vector $\boldsymbol{p}_i = [p_{i1}, ..., p_{iC}]^\top$ tends to be a one-hot vector. A one-hot vector can be viewed as a low entropy realization of $\boldsymbol{p}_i$, which means the target data outputs are similar to reference data outputs in one and only one category. Thus, we introduce a loss to capture the entropy of the target probability vectors. The entropy loss can be given by,

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

Minimizing the entropy loss gives us probability vectors $\boldsymbol{p}_i$ that tend to be one-hot vectors.

In subsection 4.2, we discussed the weighted form of MK-MMD loss with weights w_{2i-1} and w_{2i}. Here, the sample probabilities of target domain data calculated from equation 7 are assigned as target sample weights for the weighted MK-MMD loss. Because only the target samples that are categorized as source data or pseudo inlier can be used for reducing the domain disparity in MK-MMD. Therefore, the probability of target domain sample categorized as ”inlier” is projected into MK-MMD as the weight. In practice, the weights for different samples in target domain are calculated as,

$$w_i = \begin{cases} \frac{p_{i1}+p_{i2}}{p_{i1}+p_{i2}+p_{i3}} & \text{if } x^t_i \text{ categorized as source} \\ \frac{p_{i2}}{p_{i1}+p_{i2}+p_{i3}} & \text{if } x^t_i \text{ categorized as others} \end{cases} \tag{9}$$

in which, p_{i1}, p_{i2}, p_{i3} means the probability of sample x^t_i categorized as *source, pseudo-inlier, pseudo-outlier*, respectively. We assign the the weight of target sample that is classified as similar to source data with the sum of p_{i1} and p_{i2}. The reason is that we want to align the inliers of target domain with source data as much as possible. If a target sample is categorized as ”similar to source”, then it has a high probability of being an inlier, and therefore should contribute more to reducing the domain difference.

Algorithm The weighted MK-MMD loss determines how much the target samples should contribute to reducing domain disparity according to their characteristics (*source, pseudo-inlier, pseudo-outlier*). Since the domain adaptation is learned during epochs of training gradually, we decide to iteratively update the target data probabilities after each epoch for guiding and correcting the distinction between outliers and inliers.

The proposed algorithm for unsupervised cross domain image matching and outlier detection is showed below. Firstly, we initialize the target sample weights by calculating the average Euclidean distance of each target sample with all the source samples, and sort the average distances in ascending way. The proposed method is built upon the intuitive assumption that outliers originate from low-density samples. Thus, we can assume that the percentage of outliers to all the target data is no more than 50%. For initialization, we set the samples as pseudo-inliers and pseudo-outliers according to the sorted average Euclidean distance. So the target samples with distances in the first half of sorting are pseudo-inliers, and the rest are pseudo-outliers. After each epoch, we update all the target data probabilities using equations 7 and 9, where the reference of pseudo-inlier category and pseudo-outlier category is inherited from the previous epoch's prediction. We initialize inliers with probability 0.7 and outliers with probability 0.3 since we assume to have three reference categories (*source, pseudo-inlier, pseudo-outlier*). And the equal probability of being one of the three categories is $\frac{1}{3}$.

```
Algorithm 1
Input: source domain training data and target domain training data
Output: target domain training data with sample probability of being an inlier
Initialization $i = 0$, calculate the average Euclidean distance of each target data point with all the source data points, sort the distances in ascending order and assign training target samples' probabilities according to the sorted distances ($x_i \in$ first half: $p_i = 0.7$ (pseudo inliers), $x_i \in$ second half: $p_i = 0.3$ (pseudo outliers))
Repeat:
    $i = i + 1$
    make new mini batches
    minimize the overall loss function objective (10)
    update the target samples' probabilities of being an inlier by equation 7 and 9
Until target samples' probabilities are unchanged or $T > T^{'}$
```

4.4. Overall Objective

We propose a model for unsupervised cross domain image matching and outlier detection, which incorporates learning image matching information from source domain (1), unsupervised weighted domain adaptation between the source and the target (6) and outlier detection (8) in a deep convolutional neural network. The network is trained to minimize the overall objective:

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

where, $u := \{u_s \bigcup u_t\}$ and (γ, η) control the importance of domain adaptation (6) and entropy loss (8) respectively. The loss terms (1) and (8) are determined in the final layer of the network with the network output u. The weighted MK-MMD loss (6) is determined between layer outputs $\{u_s^l, u_t^l\}$ at each of the fully connected layers $\mathcal{F} = \{fc6, fc7, fc8\}$.

5. Experiments

In this section, we conduct extensive experiments to evaluate the proposed method on three different image datasets. Since we proposed an unsupervised cross domain image matching method with outlier detection, the experiments are about evaluating image matching accuracies alongside the discriminatory capability of the outlier detection. The performance results are analyzed in details.

5.1. Datasets

We used three datasets to evaluate our algorithm. To the best of our knowledge, there are no available datasets that can be used directly for both cross domain image matching and outlier detection, therefore, we made our own datasets to proceed the test. Two synthetic image datasets and one real image dataset are used for our experiments. The synthetic datasets are *Shape* dataset and *Pitts-CycleGAN* datasets. For the real dataset, we have modified the *Office* [26] dataset to form pair images as we need. Sample images from the three datasets are shown in Figure 3.

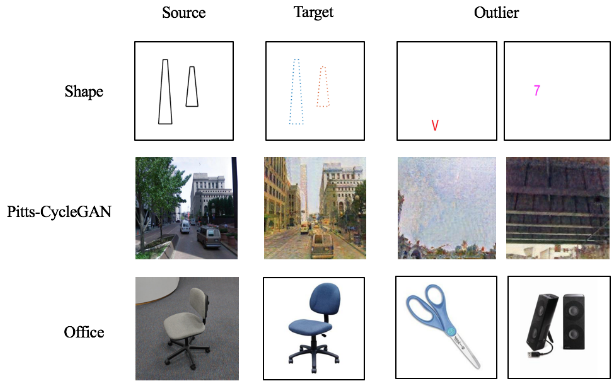


Figure 3: Sample images from *Shape*, *Pitts-CycleGAN* and *Office* datasets. The samples display the source domain, target domain and outlier images in these three datasets, respectively.

Shape Dataset is one of the artificial datasets we generate. It contains 60k source (database) images and 30k target (query) images, and there are 2800 outliers in the 30k images. The outlier images are made up of single alphabets or digits. Other "inlier" images are made up of two basic geometric shapes, such as *square, circle, diamond, triangle* etc. We define two images are a pair if the combination of shapes in the two images are the same, for example, both images contain two circles. To make two different domains, the geometric shapes of source domain are drawn with black solid lines, while shapes of target domain are drawn with colored dot lines. For training, there are 12k positive pairs and 12k negative pairs from source domain, and 24k images from target domain. We make two query datasets from source domain and target domain, with 6k images in each.

Pitts-CycleGAN contains 204k Pittsburgh Google Street View images from Pittsburgh dataset [36] as the source (database) domain, and 157k target (query) domain artificial images generated by applying the CycleGAN technique [53] to the Pittsburgh images. So the target domain images are in a painting style. In the target domain, there are around 52k outliers out of 157k target domain images, which are random sky images or city views not containing any useful landmark information. Since the original Pittsburgh dataset does not have pairs of images available, we also make pairs by selecting the images taken at the same location with similar views. There are 40k positive pairs and 40k negative pairs from source domain, and 80k images from target domain for training. For testing, one query set contains 6k source images and the other query set contains 6k target images.

Office [26] is currently one of the most popular benchmark dataset for object recognition in the domain adaptation computer vision community for image classification. The dataset consists of images from 31 different categories of everyday objects in an office environment. It has 3 domains, *Amazon, Dslr, Webcam*. The dataset has around 4,100 images with a majority of the images (2816 images) in the *Amazon* domain. Here we only choose *Dslr* as source domains and *Amazon* as target domain for our evaluation. We make pairs with images from the same categories. In our case, the outliers come from 2 randomly chosen categories ('speaker', 'scissors') out of the 31 categories, so the rest 29 categories are used as source of "inlier" images. There are 16k pairs from source domain and 16k images (including 3k outliers) from target domain for training, and 3k source query images and 3k target query images for testing.

5.2. Implementation Details

For our triplet network, the three sub-networks share the same architecture and weights. Pre-trained AlexNet [14] is used for the sub-networks. We finetune the weights of *conv4-conv5, fc6, fc7, fc8*. We set the learning rate of *fc8* 10 times the learning rate of the rest. We vary the learning rate between 10^{-4} to 10^{-5} during training with a momentum 0.9 and weight decay 5×10^{-4}. We use batch size of 64. The image features obtained by the top two sub-networks are fed into an L2 normalization layer separately before they are used to compute contrastive loss. The L2 normalization layer normalizes the two feature vectors to the same scale and makes the network easier to learn. In this way, the Euclidean distance between two feature vectors is thus upper-bounded by 2. Then we do not need to train with different margin values in a large range for the contrastive loss term. In our experiments, we train on margin values [0.5, 1.0, 2.0] on *Shape* dataset to determine the margin value. For the entropy loss and the weighted MK-MMD loss, we set $\eta = 1.5\gamma$ to help preventing the network cheating by classifying all the samples to outliers. For the weighted MK-MMD loss, we train with different γ values [0.1, 0.5, 1.0, 2.0] on *Shape* dataset to determine the proper parameter value. For 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 [41]. The mean average precision (MAP) is used as our evaluation metric.

5.3. Baseline Methods

For the setting of our research goal, there are no existing baselines to compare with directly. Thus, we separate our experiments to research on unsupervised domain adaptive image matching 5.4 and outlier detection 5.5, with our three datasets *Shape, Pitts-CycleGAN* and *Office*.

For unsupervised domain adaptive image matching, we assume there are no outliers in the target domain data. We want to evaluate the unsupervised domain adaptive matching performance. The baselines are,

- the conventional ***SIFT + Fisher Vector*** [20, 28] method
- the ***Siamese*** network [2] method, which is only trained on the source domain pair images

and 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 outlier detection, the target contains inliers and outliers for evaluating our proposed method. Our method in the experiments is ***DA+OutlierDetection***, which learns on the objective 10 from subsection 4.4. The baselines are,

- (lower bound) ***SiameseDA*** trained on the case that the target domain contains outliers, called ***SiameseDAOut*** in our experiments
- (upper bound) ***SiameseDA*** trained on the case that the target domain does not contain outliers

5.4. Unsupervised Domain Adaptive Image Matching

In this section, we study the performance of the proposed method for unsupervised cross image matching when there are no outliers in the target domain. In this case, the learning objective is

$$min_u J = L(u_s) + \gamma M(u_s, u_t), \tag{11}$$

where, the MK-MMD loss term $M(u_s, u_t)$ is the unweighted version as explained in subsection 3.2. We compare our unsupervised cross domain image matching method with the conventional *SIFT + Fisher Vector* method [20, 28] and *Siamese* network [2] method.

Parametric Exploration For our learning objective (equation 11), two main hyperparameters have to be determined: the margin of *contrastive loss* and the γ for MK-MMD loss. With *Shape* dataset, we explore the impact of these two hyperparameters on performance respectively.

As mentioned in subsection 5.2, we apply L2 normalization to the features of the pairs from the source domain before they are fed into contrastive loss. Then the Euclidean distance between two feature vectors is in range from 0 to 2. Thus, we choose to train our network on margin values [0.5, 1.0, 2.0] to determine the margin hyperparameter. At this point, we set $\gamma = 1.0$ for MK-MMD, and only change the margin value. The performance is evaluated on queries and database images both from the source domain (mark as $S \rightarrow S$), and we choose rank $K = 5$ closest retrieval images for evaluation.

Margin Value	MAP ($S \rightarrow S$)
0.5	0.517 ± 0.010
1.0	**0.531 ± 0.007**
2.0	0.465 ± 0.011

Table 1: Exploration of different margin values for *contrastive loss* on *Shape* dataset.

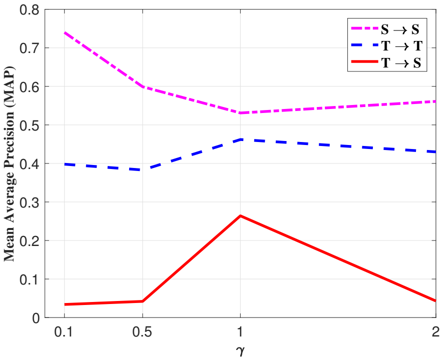


Figure 4: Empirical analysis: sensitivity of γ for MK-MMD loss term.

Dataset	$Source$	$Target$
Shape	0.950 ± 0.002	0.635 ± 0.001
Pitts-CycleGAN	0.813 ± 0.001	0.616 ± 0.0007
Office	0.992 ± 0.001	0.821 ± 0.0004

Table 2: Source and target domain difficulty validation on Siamese network for our three datasets. Mean average precision (MAP) for matching source domain query to source domain database $S \rightarrow S$, and target domain query to target domain database $T \rightarrow T$.

The impact of margin value on performance is given in Table 1. With $margin = 1.0$, we get the best performance in this case.

To determine hyperparameter γ, we keep $margin = 1.0$, and evaluate with i) query from the source domain and database from the source domain ($S \rightarrow S$), ii) query from the target domain and database from the source domain ($T \rightarrow S$), iii) query from the target domain and database from the target domain ($T \rightarrow T$), respectively. The rank is $K = 5$ for MAP measure.

We determine the γ based on the MAP of matching query target image to database source images ($T \rightarrow S$). The red line in Figure 4 indicates that when $\gamma = 1.0$, the performance is relatively optimal comparing to that with other γ values. We can notice that the performance change of $T \rightarrow T$ matching is consistent with that of $T \rightarrow S$ as expected. It shows the domain adaptation technique works for both matching cross domain images and in-domain matching for the target domain. And it is not surprising to see that the performance of matching $S \rightarrow S$ decreases a lot when applying domain adaptation. Because the network now is trying to learn the common features of the source and target domain images, the learned features for matching $S \rightarrow S$ thus is less informative than that from network without applying domain adaptation.

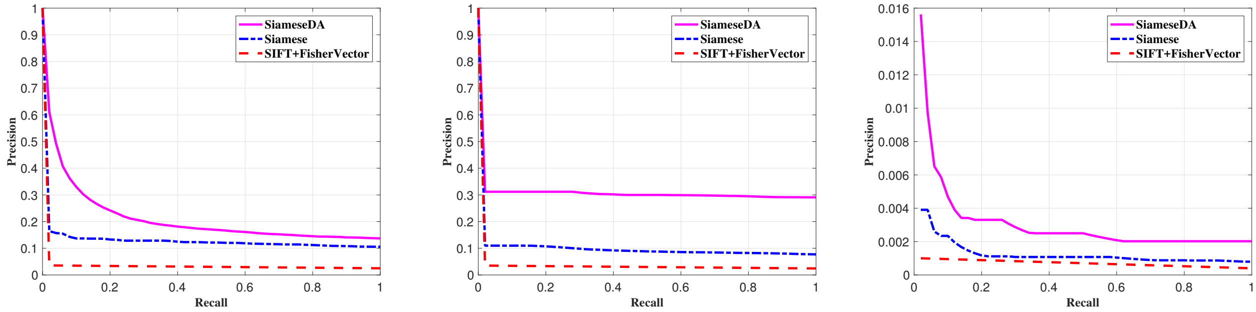
Domain Difficulty When taking cross domain image matching experiments, we first ensure that source domain images and target domain images have similar matching difficulty. We train Siamese network on target domain images and source domain images separately to test the difficulty, which is shown in Table 2. The matching results indicate that the two domains of our datasets have small difference in terms of matching difficulty, which ensure a relatively rational comparison. Because if one of the domains is too difficult to learn, the domain adaptation might not work well.

Comparative Results The results for evaluating the performance of our unsupervised cross domain image matching method are given in Table 3. Our method consistently outperforms the baselines across all the datasets for cross domain matching. With the applying of MK-MMD loss for domain adaptation, the performance of matching $S \rightarrow S$ decreases comparing to that of Siamese method without domain adaptation. This is within our expectation since the network may need to learn less from the source domain to be domain adaptive. In addition, the performance of matching $T \rightarrow S$ improves a lot for *Shape* dataset (+0.181) and *Office* dataset (+0.184), but for *Pitts-CycleGAN* it is not evident, with only +0.0014. This indicates that real life complex images are more difficult to be aligned to a common feature space. Moreover, it is worth to notice that our method also improves the in-domain image matching ($T \rightarrow T$) of the target domain.

In Figure 5, we also show the retrieval performance in terms of the trade-off of precision and recall at different match thresholds for our three datasets. To give informative results, the interpolated average precision of query images is used for drawing the precision-recall curves. We can

Methods	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	0.025 ± 0.004	0.036 ± 0.003	0.034 ± 0.003	0.035 ± 0.002	0.120 ± 0.005	0.035 ± 0.001	0.0004	0.008 ± 0.0005	0.003 ± 0.0003
Siamese	0.083 ± 0.001	0.950 ± 0.002	0.317 ± 0.006	0.107 ± 0.005	0.992 ± 0.002	0.772 ± 0.003	0.0025 ± 0.0001	0.813 ± 0.003	0.606 ± 0.005
SiameseDA	$\mathbf{0.264} \pm \mathbf{0.002}$	0.531 ± 0.001	0.462 ± 0.001	$\mathbf{0.291} \pm \mathbf{0.001}$	0.997 ± 0.001	0.775 ± 0.002	$\mathbf{0.0039} \pm \mathbf{0.0001}$	0.804 ± 0.001	0.595 ± 0.001

Table 3: Mean average precision (MAP) of rank $= 5$ for unsupervised cross 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.



(a) Shape

(b) Office

(c) Pitts-CycleGAN

Figure 5: Comparing our method to baselines for unsupervised cross domain image matching on datasets (a) *Shape*, (b) *Office* and (c) *Pitts-CycleGAN*. The curves represent interpolated average precision-recall.

clearly see that our method gains over the baseline methods.

5.5. Outlier Detection

As discussed earlier, we assume that the target domain contains outliers. We evaluate the performance of our proposed method for outlier detection and unsupervised cross domain image matching at the same time, using *Shape, Pitts-CycleGAN* and *Office* datasets.

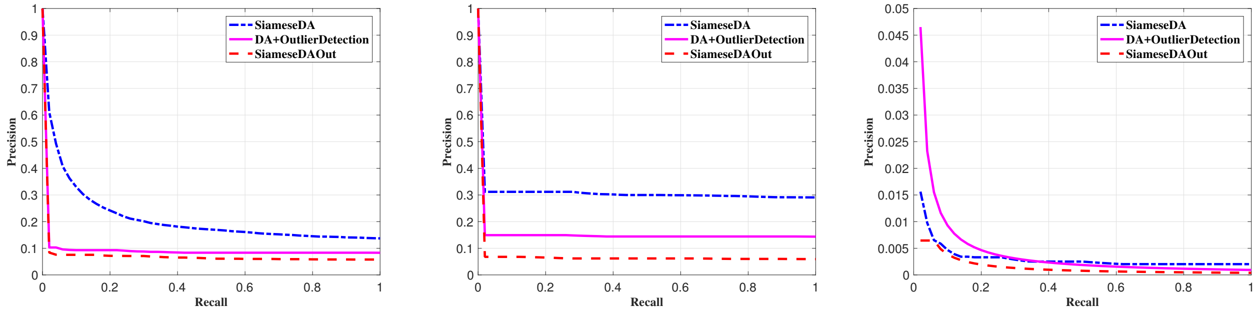
Comparative Results With outlier detection, the results of comparing the performance of our method with upper bound (*SiameseDA*) and baseline (*SiameseDAOut*) are given in Table 4. The performance is the MAP of matching target domain query images to source domain database. For this measurement, the proportion of outliers is 10% in both training target dataset and testing target dataset. The retrieval level is rank top-5. In terms of evaluation, we first distinguish and separate the outliers and inliers in testing query set. Then we only take the recognized inlier queries into mean average precision calculation. From Table 4 we can see, our method outperforms the baseline for all the three datasets, but is not better than the upper bound as expected. Interestingly, the exception exists in the performance of our method on *Pitts-CycleGAN* dataset, which even exceeds the upper bound performance. This remains to be investigated to see if the outlier detection even helps unsupervised cross domain image matching when encountering complex dataset.

Datasets ($T \rightarrow S$)	SiameseDA	SiameseDAOut	DA+OutlierDetection
Shape	0.264 ± 0.001	0.054 ± 0.0005	$\mathbf{0.119} \pm \mathbf{0.001}$
Office	0.291 ± 0.001	0.068 ± 0.0005	$\mathbf{0.159} \pm \mathbf{0.0007}$
Pitts-CycleGAN	0.0039 ± 0.0001	0.0017 ± 0.00004	$\mathbf{0.011} \pm \mathbf{0.0003}$

Table 4: Unsupervised cross domain matching MAP performance with outlier detection for our three datasets. SiameseDA is the upper bound performance without outliers, SiameseDAOut means target training set contains outliers.

To show the retrieval performance at different match thresholds, we present the precision-recall curves for all the three datasets, as shown in Figure 6. Similarly, the interpolated average precision of query images is used here for evaluating the performance. The precision-recall performance shows our method gains over the baseline for the three datasets. The comparative results illustrate that the outlier detection is necessary to be considered when outliers may exist in the target domain, which decreases the bias of cross domain matching evaluation. Some samples of query and retrievals are given in Figure 7 (a) for *Shape* dataset, (b) for *Office* dataset and (c) for *Pitts-CycleGAN* dataset.

Impact of Outlier Proportion We also report the F_1-score as a measure to evaluate the performance of outlier-inlier distinction of our method. Figure 8 shows the F_1-score of out method as a function of the portion of outlier samples for the three datasets. As can be seen, with the in-

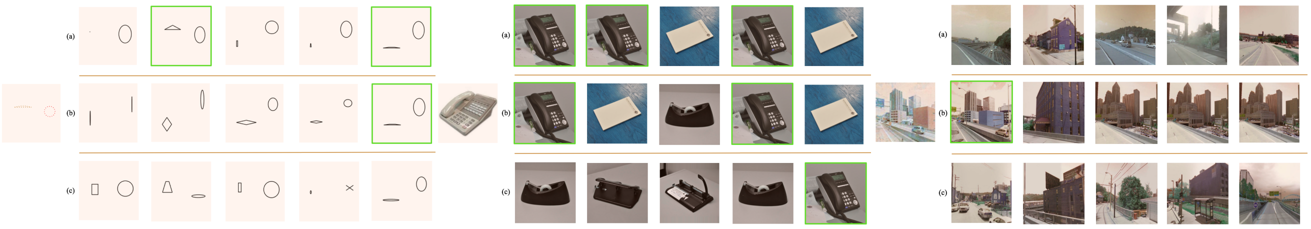


(a) Shape

(b) Office

(c) Pitts-CycleGAN

Figure 6: Comparing our method *DA+OutlierDetection* to upper-bound method *SiameseDA* and baseline method *SiameseDAOut* for the performance of unsupervised cross domain image matching with outlier detection on datasets (a) *Shape*, (b) *Office* and (c) *Pitts-CycleGAN*. The curves represent interpolated average precision-recall.



(a) Shape

(b) Office

(c) Pitts-CycleGAN

Figure 7: Retrieval results for three datasets. For each dataset, the left column shows a query image, the top row shows the top-5 results for *SiameseDA* method, the middle row shows the top-5 results for our *DA+OutlierDetection* method, and the bottom row shows the top-5 results for *SiameseDAOut* method. Green boxes indicate the corresponding correct test impression.

crease in the number of outliers, our method operates consistently robust and successfully detects the outliers. However, it is important to notice the downside of this method, which wrongly recognizes some inliers as outliers.

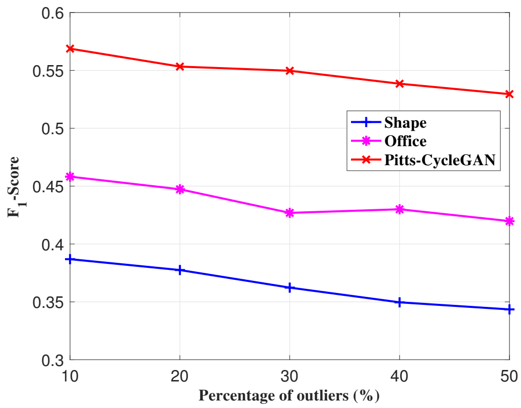


Figure 8: Comparisons of F_1-scores on three datasets for different percentages of outlier samples involved in the target domain.

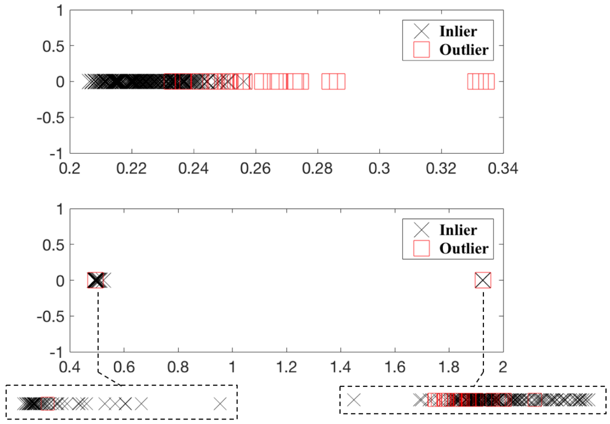


Figure 9: The average distance change of target samples to source samples before and after training on our method. The upper subplot is the target sample distribution by distance before training, the bottom one is that after training. The x-axis is the average target sample distance to source data. This is measured on *Shape* dataset.

Data Distribution Change Since the target training data are the ”inlier” reference in our system, we measure the average distance between every testing target sample and source reference data to show the distinguishing ability of our method. Figure 9 shows the distance change of target samples consisting of inliers and outliers before (upper subplot) and after training (bottom subplot) the network on our method with *Shape* dataset. The portion of outliers in this experiment is 10% during training. It is obvious that the inliers and outliers in the target domain are still hard to distinguish before training. After training the network on our method, we can see that in the bottom subplot, the outliers and inliers are well separated even though it sacrifices some inliers (false negative).

5.6. Limitations

From the experiment results, it is important to notice that our outlier detection method categorizes some inlier samples of the target domain as outliers in the training. This is mainly caused by the way of initializing the probabilities of the target domain training data. Since it is unsupervised learning, we have to initialize with the assumption of the worst case: the portion of outliers is 50%. And the initial categorizing of inliers and outliers is measured by average Euclidean distance between every target sample and the source domain data, which is not accurate (e.g. outliers may be closer to the source than inliers). These above lead some inliers to be classified as outliers in the end.

6. Conclusion

We have proposed a triplet network that is trained for unsupervised 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 learning with contrastive loss and weighted MK-MMD loss, (ii) outlier detection with entropy loss by training the network in an iterative way. The results on several datasets demonstrate that the proposed method is capable of detecting outlier samples in the target domain and achieving unsupervised cross domain image matching at the same time. But our method still needs improvement to overcome the wrongly categorizing inliers to outliers problem. In addition, it is worth further exploration to see if domain adaptation alone can produce a good performance for unsupervised cross domain image matching, even without the paired-image information from the source domain.

References

[1] R. Chalapathy, A. K. Menon, and S. Chawla. Anomaly detection using one-class neural networks. *arXiv: 1802.06360*, 2018. 2, 3

[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. 2, 3, 8

[3] 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. *arXiv preprint arXiv:1310.1531*, 2013. 2

[4] B. Fernando, T. Tommasi, and T. Tuytelaars. Location recognition over large time lags. *Computer Vision and Image Understanding*, 139:21–28, 2015. 1

[5] 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, 3, 4

[6] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv: 1311.2524*, 2013. 2

[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 2672–2680, 2014. 3

[8] 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. 3

[9] 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 F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1205–1213. 2012. 2, 3, 4, 5

[10] 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. 4

[11] X. Ji, W. Wang, M. Zhang, and Y. Yang. Cross-domain image retrieval with attention modeling. *2017 ACM Multimedia Conference*, 2017. 1, 2

[12] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv: 1703.05192*, 2017. 2

[13] B. Kong, J. S. S. III, D. Ramanan, and C. C. Fowlkes. Cross-domain image matching with deep feature maps. *arXiv preprint arXiv:1804.02367*, 2018. 2, 3

[14] 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. 2, 8

[15] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays. Learning deep representations for ground-to-aerial geolocalization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 3

[16] T. Y. Lin, Y. Cui, S. Belongie, and J. Hays. Learning deep representations for ground-to-aerial geolocalization. In *2015*

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5007–5015, 2015. 3

[17] 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. 3

[18] 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. 2, 4

[19] 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. 4, 5

[20] 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, 3, 8

[21] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1717–1724, 2014. 2

[22] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015. 3

[23] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid. Local convolutional features with unsupervised training for image retrieval. pages 91–99, 2015. 2

[24] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. pages 143–156, 2010. 2

[25] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli. Adversarially learned one-class classifier for novelty detection. *arXiv: 1802.09088*, 2018. 2, 3

[26] 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, 7, 8

[27] 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. 2

[28] 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. 8

[29] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. *arXiv: 1704.01705*, 2017. 2, 3

[30] B. Schlkopf and A. J. Smola. Support vector machines, regularization, optimization, and beyond. In *MIT Press*, 2002. 2

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv: 1409.1556*, 2014. 2

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv: 1409.4842*, 2014. 2

[33] R. Tao, E. Gavves, and A. W. M. Smeulders. Siamese instance search for tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3

[34] D. M. Tax and R. P. Duin. Support vector data description. *Machine Learning*, 54:45–66, Jan 2004. 2

[35] Y. Tian, C. Chen, and M. Shah. Cross-view image matching for geo-localization in urban environments. *In CVPR*, 2017. 1, 2, 3

[36] 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. 2, 7

[37] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. *CoRR*, abs/1510.02192, 2015. 4

[38] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. *arXiv: 1702.05464*, 2017. 2

[39] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. *arXiv: 1702.05464*, 2017. 3

[40] H. Venkateswara, S. Chakraborty, and S. Panchanathan. Deep-learning systems for domain adaptation in computer vision: Learning transferable feature representations. *IEEE Signal Processing Magazine*, 34:117–129, 2017. 2

[41] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. *arXiv: 2017*, 2017. 2, 3, 4, 8

[42] F. Wang, L. Kang, and Y. Li. Sketch-based 3d shape retrieval using convolutional neural networks. *arXiv: 1504.03504*, 2015. 3

[43] Z. Wang, B. Fan, and F. Wu. Local intensity order pattern for feature description. pages 603–610, 2011. 2

[44] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun. Learning discriminative reconstructions for unsupervised outlier removal. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2

[45] T. Xiao, H. Li, W. Ouyang, and X. Wang. Learning deep feature representations with domain guided dropout for person re-identification. *arXiv: 1604.07528*, 2016. 3

[46] 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. 4

[47] C. You, D. P. Robinson, and R. Vidal. Provable self-representation based outlier detection in a union of subspaces. *arXiv: 1704.03925*, 2017. 2

[48] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. *arXiv: 1504.03641*, 2015. 3

[49] J. Zbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. *arXiv: 1409.4326*, 2014. 3

[50] X. Zhang, F. X. Yu, S. Chang, and S. Wang. Deep transfer network: Unsupervised domain adaptation. *arXiv: 1503.00591*, 2015. 2

[51] X. Zhang, F. X. Yu, S. Chang, and S. Wang. Deep transfer network: Unsupervised domain adaptation. *arXiv: 1503.00591*, 2015. 3

[52] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, pages 487–495, 2014. 2

[53] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv: 1703.10593*, 2017. 2, 7

2 Introduction

Over the past decade, fueled by cheaper storage and availability of ever increasing computational resources, there has been an explosive increase in the collection of data about the same concept from multiple sources and in multiple formats. This leads to pattern matching scenarios across that necessitate the development of learning algorithms that have the ability to learn concepts from diverse source of data, much like human learning. And cross domain image matching is a research object that belongs to this topic.

There are many methods that have been implemented to deal with the cross domain image matching problem, such as using hand-crafted features for matching [1], convolutional neural network based image matching[2] [3], etc. In the convolutional neural network based image matching field, there are also two main categories as supervised cross domain image matching [3] and unsupervised cross domain image matching [4]. For supervised approach, images in both domains have labels; for unsupervised approach, one of the two domains contains only unlabeled samples. When talking about domains, one thing that can not be ignored is the outlier images in either domain. In the image datasets collected from real world, it is normal to find that the datasets contain some images that not belong to both domains. And these images are called outliers, which could compromise the learning model. Therefore, pruning the irrelevant images, i.e., the outliers, becomes necessary for the cross domain image matching task.

This work aims to investigate the possibility of detecting the outliers as well as achieving the unsupervised cross domain image matching task at the same time by machine learning techniques. Towards that end, the following literature research is carried out to present an analysis of the state of the art research in the topic of unsupervised cross domain images matching with detection of outliers at the same time.

In this introduction chapter, section 2.1 presents the motivations behind this research, section 2.2 outlines the research objectives to be addressed. The final section of this chapter shows the outlines of following chapters.

2.1. Motivation

Cross domain image matching is about searching a large-scale dataset (gallery dataset) to find the images, which are visually similar to a given query image across different domains. The cross domain here means that the marginal distributions of the data in the two domains are different, but the conditional distributions are the same. This is possible because the two domains are assumed to be correlated. This correlation is often modeled as covariate shift [5]. Cross domain image matching is still a challenging task since small perceptual differences can result in arbitrarily large differences at the raw pixel level. In addition, real world cross domain image matching usually encounters the problem that the knowledge of query specific domain is not available. It means that, for example, the paired information of images in domain A (gallery dataset) is given, but the information of a similar pair of two images (one from domain A and the other from domain B) is not available. This makes it difficult to develop a generalized solution for multiple potential visual domains.

Cross domain image matching arises in a variety of application domains. For example, matching aerial photos to GIS map data for location discovery [6] [7] [8], image retrieval from hand drawn sketches and paintings [9] [10], and matching images to 3D models [11]. Some of these research works present a semi-supervised setting of the datasets, which means that part of the label information is given for the query specific domain. But in our work, the unsupervised setting will be considered, which is closer to the real world cases.

In unsupervised cross domain image matching, the labeled dataset is usually called the source dataset, the unlabeled query dataset is called the target dataset. Figure 2.1 shows the concept of unsupervised cross domain image matching with outliers in the query target domain. To solve this challenge, there are two main approaches, one is the traditional way of extracting feature descriptors (SIFT [12], LIOP [13], Patch-CKN [14]) or representations (Fisher Vectors [15], DeCAF [16]) of images and matching with different metrics. These methods focused on designing feature extractor for each domain which yield domain invariant descriptors which can then be directly compared. However, these approaches would fail when matching in high oblique views [3]. The other main approach is utilizing the power of deep learning with proper learning objectives [3][17][18]. Deep convolutional neural net (CNN) features hierarchies have proven incredibly effective at a wide range of recognition tasks. In this work, we chose to investigate deep learning approaches for unsupervised cross domain image matching problem.

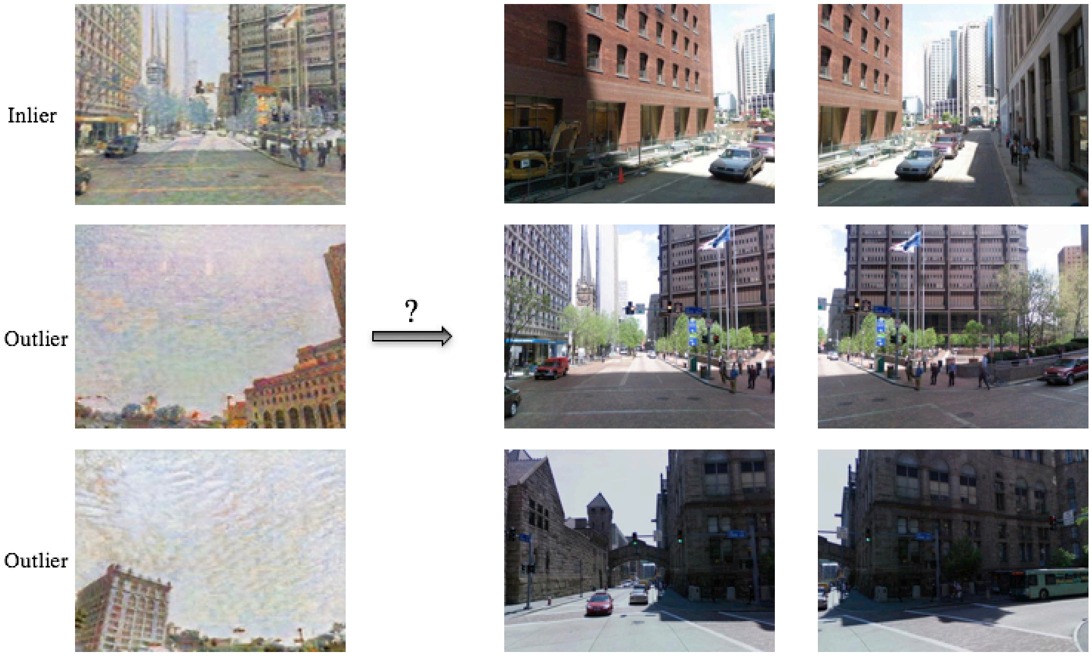


Figure 2.1: An example of unsupervised cross domain image matching. The left column is the query images from a painting style target dataset containing sky images as outliers; the right two columns are photographs of city from the source dataset. The query images and photographs are in different domains. The paired-image information in the source domain is given, which is that the two photographs in each row are a positive pair. We need to detect the outliers in the query target set to avoid the impact of outliers on the matching performance.

Even though many deep learning frameworks have been used in cross domain image matching, the settings of those research are in a supervised manner. Those work have information of what is a matching pair of images from different domains during training [18][17][3]. In our work, we would like to investigate the possibility of only using the labeled source domain images and unlabeled target domain image to achieve the cross domain matching task.

When matching cross domain images, another aspect that can not be ignored is the noise images existing in the query image domain. For example, in Figure 2.1, both domains' images should be city views, but the query dataset may also contain sky only images (e.g. outliers in Figure 2.1) or pedestrian only images that do not belong to both of the two domains. This appears a lot when collecting data from various sources. We call these images as outliers. Previous work treated the query images as they are all inliers, which actually not and may affect the performance of training the network. In this work, we investigate the possibility of rejecting the outliers and matching the cross domain images at the same time. To our knowledge, no previous work focused on this problem setting.

2.2. Research objectives

The aim of this research work is to investigate the unsupervised cross domain image matching in a case where the query target domain has some other image categories that do not belong to both of the two domains, e.g, photos that are not depicting city views. From the research objective, the following research questions are derived,

- To define an empirical loss for the network that can encourage domain adaptation matching as well as perform outlier detection.
- Is it possible to use the labels in the source domain to learn a feature space that the outliers and inliers of the target domain are more distinct?

2.3. Outline

The rest of the thesis report is organized as follows, Chapter 2 gives the theoretical background on the deep learning network. Chapter 3 introduces the domain adaptation methods. Chapter 4 explains the outlier detection approaches. Chapater 5 discusses the evaluation methods used to measure the network performance in our experiments. Chapter 6 presents some additional experiments that are not mentioned in the scientific paper.

References

[1] N. Ofir, S. Silberstein, D. Rozenbaum, and S. D. Bar, *Deep multi-spectral registration using invariant descriptor learning,* (2018), arXiv:1801.05171 .

[2] S. Chopra, R. Hadsell, and Y. LeCun, *Learning a similarity metric discriminatively, with application to face verification,* in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005) pp. 539–546.

[3] Y. Tian, C. Chen, and M. Shah, *Cross-view image matching for geo-localization in urban environments,* (2017), arXiv:1703.07815 .

[4] R. Gopalan, R. Li, and R. Chellappa, *Domain adaptation for object recognition: An unsupervised approach,* in *2011 International Conference on Computer Vision* (2011) pp. 999–1001.

[5] H. Venkateswara, S. Chakraborty, and S. Panchanathan, *Deep-learning systems for domain adaptation in computer vision: Learning transferable feature representations,* in *IEEE Signal Processing Magazine* (2017) pp. 117–129.

[6] T. Senlet, T. El-Gaaly, and A. M. Elgammal, *Hierarchical semantic hashing: Visual localization from buildings on maps,* in *22nd International Conference on Pattern Recognition* (2014) pp. 2990–2995.

[7] D. Costea and M. Leordeanu, *Aerial image geolocalization from recognition and matching of roads and intersections,* (2016), arXiv:1703.07815 .

[8] M. Divecha and S. Newsam, *Large-scale geolocalization of overhead imagery,* in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2016) pp. 32:1–32:9.

[9] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu, *Sketch2photo: Internet image montage,* in *ACM SIGGRAPH Asia 2009 Papers* (2009) pp. 124:1–124:10.

[10] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros, *Data-driven visual similarity for cross-domain image matching,* in *Proceedings of the 2011 SIGGRAPH Asia Conference* (2011) pp. 154:1–154:10.

[11] B. C. Russell, J. Sivic, J. Ponce, and H. Dessales, *Automatic alignment of paintings and photographs depicting a 3d scene,* in *ICCV Workshops* (2011) pp. 545–552.

[12] D. G. Lowe, *Object recognition from local scale-invariant features,* in *Proceedings of the International Conference on Computer Vision*, ICCV '99 (1999) pp. 1150–1157.

[13] Z. Wang, B. Fan, and F. Wu, *Local intensity order pattern for feature description,* in *Proceedings of the 2011 International Conference on Computer Vision* (2011) pp. 603–610.

[14] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid, *Local convolutional features with unsupervised training for image retrieval,* in *2015 IEEE International Conference on Computer Vision* (2015) pp. 91–99.

[15] F. Perronnin, J. Sánchez, and T. Mensink, *Improving the fisher kernel for large-scale image classification,* in *Proceedings of the 11th European Conference on Computer Vision: Part IV* (2010) pp. 143–156.

[16] 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,* (2013), arXiv:1310.1531 .

[17] X. Ji, W. Wang, M. Zhang, and Y. Yang, *Cross-domain image retrieval with attention modeling,* (2017), arXiv:1709.01784 .

[18] B. Kong, J. S. S. III, D. Ramanan, and C. C. Fowlkes, *Cross-domain image matching with deep feature maps,* (2018), arXiv:1804.02367 .

3

Background on Deep Learning

3.1. Deep learning

Deep Learning is a sub-field of machine learning methods, which layers of artificial neurons to learn data representations [1]. Deep learning architectures have gained success in computer vision, speech recognition, natural language processing, and many other domains. The basic computational unit for most deep learning frameworks are artificial neurons with trainable parameters, which are trained by backpropagation procedure [2].

Neural networks Neural networks are modeled as collections of neurons that are connected in an acyclic graph. They are often organized into distinct layers of neurons. Figure 3.1 illustrates a mathematical model of a neuron. The weights w_i are learnable and control the strength of influence and its direction of one neuron on an activation function.

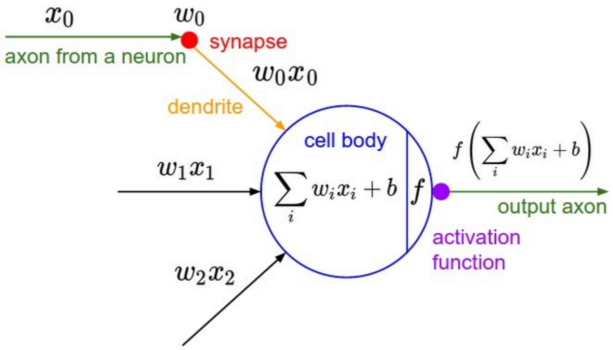


Figure 3.1: A mathematical model of a single neuron with 3 inputs, 3 + 1 learnable weights and bias parameters. After the affine transformation, an activation function f is applied.

Mathematically, each neuron applies an affine transformation of the input $\mathbf{x} = [x_1, x_2, ..., x_n]^T$:

$$u = \sum_{i=1}^{n} w_i x_i + b, \tag{3.1}$$

where w_i is the weight for input x_i and b is the bias term. A non-linear activation function f is applied after this affine transformation:

$$o = f(u). \tag{3.2}$$

, which is the final output o.

By connecting multiple neurons in different layers, a neural network is formed. For regular neural networks the most common layer type is the fully-connected layer in which neurons between two adjacent layers are fully pairwise connected, but neurons within a single layer share no connections. Figure 3.2 is an example of a 3-layer neural network.

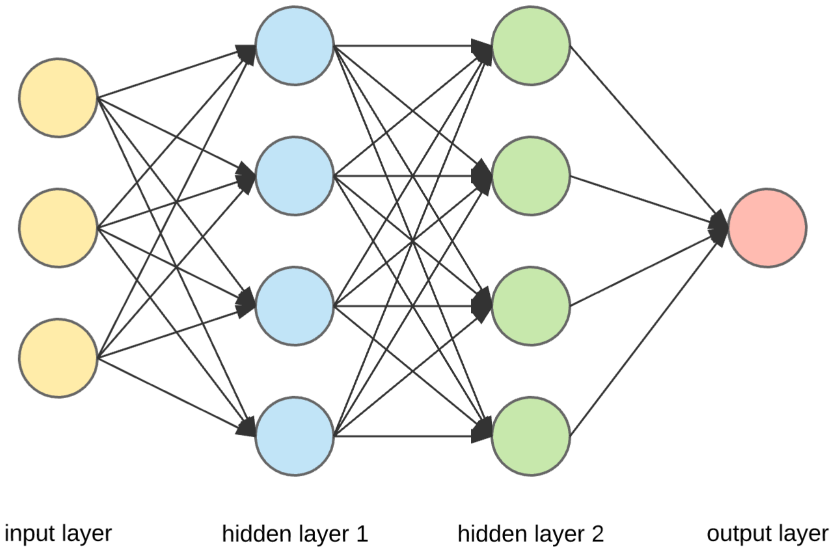


Figure 3.2: A 3-layer neural network with three inputs, two hidden layers of 4 neurons each and one output layer.

Activation functions Some common activation functions are Sigmoid, Tanh and Rectified Linear Unit (ReLU). Figure 3.3 presents these three non-linearity outputs given a range of input values.

Sigmoid non-linearity has two major drawbacks, which are Sigmoid saturates and killed gradients and its outputs are not zero-centered. Like the Sigmoid neuron, Tanh activations saturate, but its output is zero-centered. The ReLU unit has become very popular in the last few years. It has been widely used in deep learning networks compared to the other two activation functions. ReLU was found to greatly accelerate the convergence of stochastic gradient descent [3]. Compared to Tanh/Sigmoid neurons that involve expensive operations, ReLU can be implemented by simply thresholding a matrix of activations at zero. However, ReLU units can be fragile during training and can 'die'. There is an activation called Leaky ReLU attempting to fix this problem. Instead of the function being zero when $x < 0$, a leaky ReLU will instead have a small negative slope.

Learning and optimization Neural networks are usually initialized with small random weights, and these parameters are updated during training by minimizing a loss function L. The learning process is achieved by gradient descent parameter update method with gradients calculated by backpropagation [2]. A set of parameter update techniques have been developed to optimize the neural network, such as Stochastic Gradient Descent (SGD) [4], RMSprop [5], Adam [6], and etc.

3.2. Convolutional neural networks

Convolutional neural networks (ConvNet) take advantage of the fact that the inputs are images, and they constrain the network in a more reasonable way with less parameters than the regular neural networks. ConvNets are built by three main types of layers, which are convolutional layer, pooling layer, and fully-connected layer.

Convolutional layer To have a better understanding of convolutional layer, the convolution operation is first discussed. A convolution is an integral function that computes the amount of overlap between two

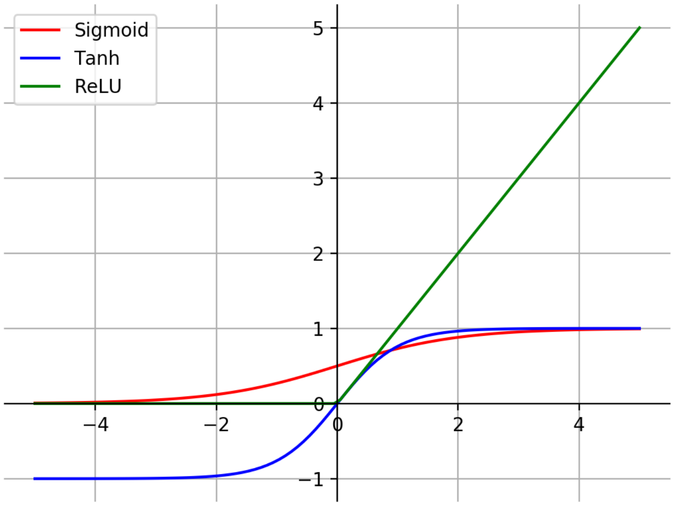


Figure 3.3: Three common activation functions. Sigmoid maps real numbers to range between $[0,1]$, Tanh squashes real numbers to range between $[-1,1]$, and ReLU function is zero when input < 0 and then linear with slope 1 when input > 0.

functions [7]. Its mathematical form is as the following equation:

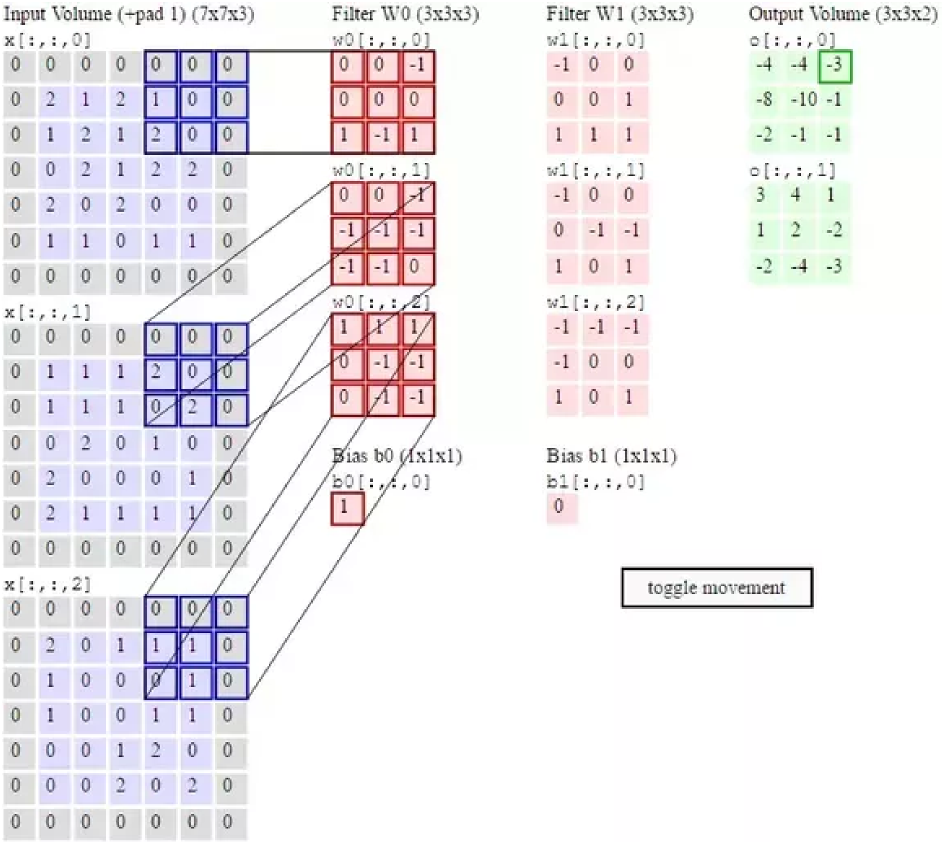
$$f * g = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau, \tag{3.3}$$

The integral calculates the product of function f and g as g is shifted over f by τ. Convolution is widely used as filter in image processing [7].

The basic idea of convolutional layer is to connect each neuron to only a local region of the input volume. The spatial extent of this connectivity is a receptive field of the neuron (or the filter size). The extent of the connectivity along the depth axis is always equal to the depth of the input volume. Then the output volume of the conv layer depends on the depth, stride and zero-padding. One of the purpose for introducing the convolutional layer is to decrease the parameters in the network. So one important character of convolutional layer is parameter sharing of all neurons in a single depth slice. Then the forward pass of the convolutional layer can be computed as a convolution of the neuron's weights with the input volume since the filers 'slides' across the input volume. A visualization of convolutional layer is shown in Figure 3.4

Pooling layer Pooling layer is usually insert between successive convolutional layers. It can progressively reduce the spatial size of the representations, which is helpful for reducing the amount of parameters and computation of the network, and controlling overfitting. The pooling layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation. Thus, the depth of the output remains unchanged. A visual expression of max pooling with $F = 2, S = 2$ is shown in Figure 3.5.

AlexNet In this work, we used AlexNet [3] as sub-network, so it is necessary to introduce AlexNet briefly. AlexNet was the first work that popularized convolutional networks in computer vision. AlexNet architecture is illustrated in Figure 3.6. It consists of five convolutional layers and three fully connected layers, and the output is 1000-class softmax for classification. AlexNet has been widely used as a pre-trained model in research. Here we also use AlexNet as our sub-network for our task with finetuning it.



Input Volume (+pad 1) (7x7x3)

x[:,:,0]

0	0	0	0	0	0	0
0	2	1	2	1	0	0
0	1	2	1	2	0	0
0	0	2	1	2	2	0
0	2	0	2	0	0	0
0	1	1	0	1	1	0
0	0	0	0	0	0	0

x[:,:,1]

0	0	0	0	0	0	0
0	1	1	1	2	0	0
0	1	1	1	0	2	0
0	0	2	0	1	0	0
0	2	0	0	0	1	0
0	2	1	1	1	1	0
0	0	0	0	0	0	0

x[:,:,2]

0	0	0	0	0	0	0
0	2	0	1	1	1	0
0	1	0	0	0	1	0
0	1	0	0	1	1	0
0	0	0	1	2	0	0
0	0	0	2	0	2	0
0	0	0	0	0	0	0

Filter W0 (3x3x3)

w0[:,:,0]

0	0	-1
0	0	0
1	-1	1

w0[:,:,1]

0	0	-1
-1	-1	-1
-1	-1	0

w0[:,:,2]

1	1	1
0	-1	-1
0	-1	-1

Bias b0 (1x1x1)

b0[:,:,0]

1

Filter W1 (3x3x3)

w1[:,:,0]

-1	0	0
0	0	1
1	1	1

w1[:,:,1]

-1	0	0
0	-1	-1
1	0	1

w1[:,:,2]

-1	-1	-1
-1	0	0
1	0	1

Bias b1 (1x1x1)

b1[:,:,0]

0

Output Volume (3x3x2)

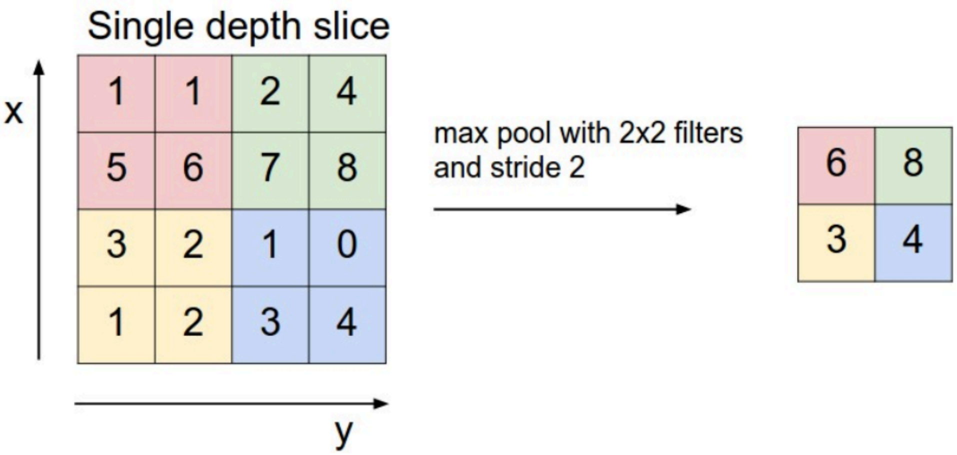
o[:,:,0]

-4	-4	-3
-8	-10	-1
-2	-1	-1

o[:,:,1]

3	4	1
1	2	-2
-2	-4	-3

Figure 3.4: A visualization of convolutional layer. The input volume is of size $W = 5, H = 5, D = 3$, and the CONV layer parameters are $K = 2, F = 3, S = 2, P = 1$. The green box is the output activations, in which each element is computed by elementwise multiplying the hightlighted input (blue) with the filter (red), summing it up, and then offsetting the result by the bias. [8]



Single depth slice

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

Figure 3.5: A visual expression of max pooling with a stride of 2. Each max is taken over 4 numbers (little 2x2 square). [8]

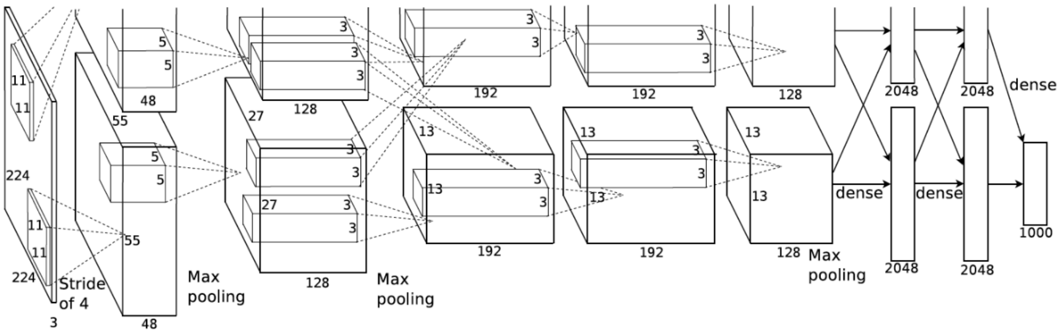


Figure 3.6: The network architecture of AlexNet. It was separated into two parallel parts since the GPU calculation was not strong enough at that time [3].

3.3. Siamese network

For paired comparison tasks, the Siamese architectures have been widely implemented. It integrates feature extraction and comparison in a single model that can be optimized end-to-end [9]. The SiameseNet architecture is presented in Figure 3.7.

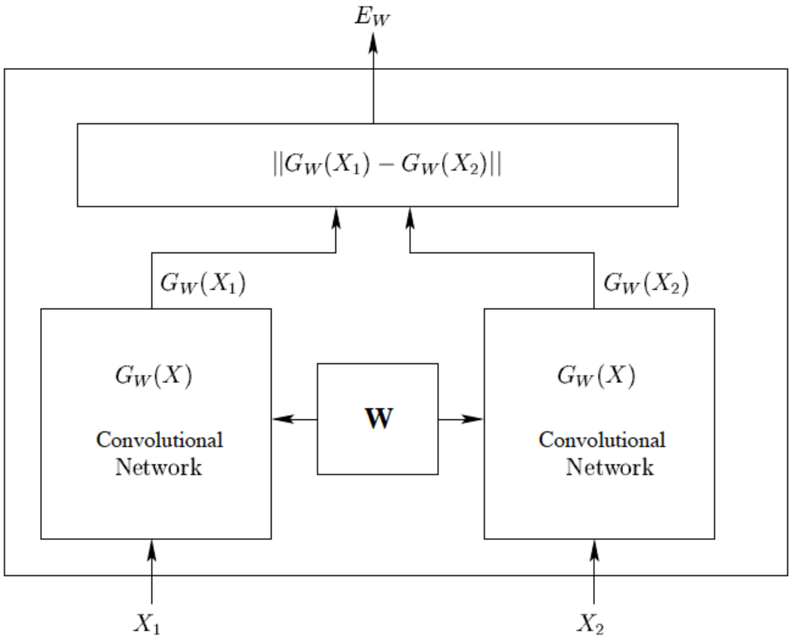


Figure 3.7: The network architecture of Siamese. X_1 and X_2 are a pair of images as input to the two identity sub-networks $G_w(X)$ that share weights. Output is E_w is a scalar energy that measures the compatibility between X_1 and X_2 [10].

It has a pair of input images to two sub-networks that are exactly the same. The output is a scalar energy E_w that measures the similarity between the input pair. For training, a contrastive loss is developed, which depends on the input and the parameters only indirectly through the energy [10]. The loss function is of the form:

$$L(W) = \sum_{i}^{P} L(W,(Y,X_1,X_2)^i) \tag{3.4}$$

$$L(W,(Y,X_1,X_2)^i) = (1-Y)L_G(E_W(X_1,X_2)^i) + YL_I(E_W(X_1,X_2)^i), \tag{3.5}$$

where $(Y,(X_1,X_2)^i)$ is the i-sample, which is composed of a pair of images and a label (genuine $Y = 1$ or impostor $Y = 0$), L_G is the partial loss function for a genuine pair, L_I is the partial loss function for an impostor pair, and P is the number of training samples. L_G and L_I are designed in such a way that minimization of L will decrease the energy of genuine pairs and increase the energy of impostor pairs [10]. Here we employ the contrastive loss [11]:

$$L(W,(Y,I_1,I_2)^i) = (1-Y)D(I_1,I_2)_i^2 + Y(max(0,m - D(I_1,I_2)_i))^2. \tag{3.6}$$

where I_1 and I_2 are the output representations of X_1 and X_2 from SiameseNet, and $D(I_1,I_2)$ is the distance of the two representations. In our work, we adopt Euclidean distance metric.

In fact, SiameseNet is originally developed for face verification task [10]. But it shows the ability for image matching, image retrieval tasks in works of recent years. Past works have demonstrated that Siamese networks learn good features for person re-identication, face recognition, and stereo matching [12][13][14]. We also employ this network structure as part of our network to help learn deep representations to distinguish matched and unmatched pairs in cross domain images of our datasets.

References

[1] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, Vol. 1 (MIT press Cambridge, 2016).

[2] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, *Backpropagation applied to handwritten zip code recognition,* Neural computation **1**, 541 (1989).

[3] 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* (2012) pp. 1097–1105.

[4] L. Bottou, *Large-scale machine learning with stochastic gradient descent,* in *Proceedings of COMPSTAT'2010* (Springer, 2010) pp. 177–186.

[5] T. Tieleman and G. Hinton, *Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,* COURSERA: Neural networks for machine learning **4**, 26 (2012).

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

[7] R. N. Bracewell and R. N. Bracewell, *The fourier transform and its applications,* (McGraw-Hill, 1986).

[8] *Lecture note of stanford cs231n convolutional neural networks for visual recognition,* (2018).

[9] B. Kong, J. S. S. III, D. Ramanan, and C. C. Fowlkes, *Cross-domain image matching with deep feature maps,* (2018), arXiv:1804.02367 .

[10] S. Chopra, R. Hadsell, and Y. LeCun, *Learning a similarity metric discriminatively, with application to face verification,* in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005) pp. 539–546.

[11] 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* (2006) pp. 1735–1742.

[12] O. M. Parkhi, A. Vedaldi, and A. Zisserman, *Deep face recognition,* in *British Machine Vision Conference* (2015).

[13] T. Xiao, H. Li, W. Ouyang, and X. Wang, *Learning deep feature representations with domain guided dropout for person re-identification,* (2016), arXiv:1604.07528 .

[14] J. Zbontar and Y. LeCun, *Computing the stereo matching cost with a convolutional neural network,* (2014), arXiv:1409.4326 .

4

Domain Adaptation

In this chapter, we introduce the domain adaptation definition. Since part of our research goal is on unsupervised domain adaptive image matching, we will mainly discuss about two unsupervised domain adaptation methods that have been investigated in computer vision research.

4.1. Introduction to domain adaptation

Traditional machine learning paradigms train statistical models to make predictions or more on unseen data in the future. These models do not guarantee optimal performance if the test data are vastly different from the training data. To reduce the effort involved in recollecting labeled data and retraining a new model, knowledge transfer between tasks or domains is desirable [1].

In a standard supervised learning setting, test data are sampled from the same distribution as the training data. Therefore, trained models can guarantee a level of performance. When test data come from a distribution very different from training data, transfer of knowledge from the training domain is necessary to build robust models. Domain adaptation is one of the multiple types paradigms in transfer learning. For the introduce here, the definitions of *domain* and *task* are outlined in line with [1]. A domain D is said to consist of two components, a feature space $\mathcal{X}$ and a marginal probability distribution $P(\mathbf{X})$ that governs the feature space, where $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\} \in \mathcal{X}$ is the set of samples from the feature space. Two domains are considered different if their feature spaces are different, or their probability distributions are different. If $D = \{\mathcal{X}, P(\mathbf{X})\}$ is a domain, then a task T consists of two components, $T = \{y, f(\cdot)\}$, where y is the label space and $f(\cdot)$ is the function $f : \mathcal{X} \rightarrow y$.

In domain adaptation, the source domain D_S and the target domain D_T are not the same, and the goal is to solve a common task $T = \{y, f(\cdot)\}$. For example, in an image-recognition task, the source domain could contain labeled images of objects against a white background, and the target domain could consist of unlabeled images of objects against a noisy background. Both domains inherently have the same set of images categories. The difference between the domains is modeled as the variation in their joint probability distributions $P_S(\mathbf{X,Y}) \neq P_T(\mathbf{X, Y})$ [2]. Standard domain adaptation assumes that there are plenty of labels data in the source domain, while there is no or few labeled data in the target domain. Particularly, in unsupervised domain adaptation, there is no labeled data in the target domain. Then the key task of domain adaptation is to get a good estimation of $P_T(\mathbf{X,Y})$ using the source data distribution estimation $P_S(\mathbf{X,Y})$. This is possible since the two domains are assumed to be correlated. This correlation is often modeled as covariate shift, where $P_S \neq P_T$ and $P_S(\mathbf{Y|X}) \approx P_T(\mathbf{Y|X})$.

4.2. Deep learning domain adaptation with statistic methods

Using deep networks as feature extractors, the performance of naive statistic domain adaptation methods can be boosted [3, 4]. Some of the naive statistic domain adaptation methods are, for example, maximum mean discrepancy (MMD) [5], moment alignment [6], or a loss function that drives the source and target classifiers to be indistinguishable. Reducing domain disparity through nonlinear alignment of data has been made possible with MMD, which is a nonparametric distance kernel Hilbert space (RKHS). The data are mapped to a high-dimensional space defined by $\Phi = [\phi(\mathbf{x}_1), ..., \phi(\mathbf{x}_n)]$, $\phi : R^d \Rightarrow \boldsymbol{H}$ defines a mapping function, and $\boldsymbol{H}$ is a RKHS. Gretton *et al.* in [5] introduced the MMD to estimate the distance between the source and the target data sets, which is given by

$$MMD = |\frac{1}{n_s}\sum_{i=1}^{n_s}\phi(x_i^s) - \frac{1}{n_t}\sum_{j=1}^{n_t}\phi(x_j^t)|_H^2. \tag{4.1}$$

The distance between two distributions is the distance between their means in an RKHS. When two data sets belong to the same distribution, their MMD is zero.

Long *et al.* proposed the deep adaptation networks (DAN) model [7], which incorporates an MMD loss for all the fully connected layers of AlexNet [8]. The MMD loss is estimated for the feature representations over every minibatch during training. Based on the network architecture of the DAN [7], Venkateswara *et al.* [9] developed a hashing algorithm for domain adaptation. The architecture of the domain adaptive hash (DAH) network is based on the VGG-F, and domain alignment is achieved using MMD. The residual transfer network (RTN) in [10], which also achieved feature adaptation with MMD loss. In all of these deep domain adaptation approaches, the weights are shared between the source and the target network to ensure domain invariant features.

4.3. Domain adversarial learning methods

The introduce of generative adversarial networks (GANs) by Goodfellow *et al.* [11] helps the research in domain adaptation. GANs are networks that generate data (text, images, audio, etc.) such that the data follow a predetermined distribution $P(X)$. A vanilla GAN implementation has two deep networks, generator $g(\cdot)$ and discriminator $f(\cdot)$, competing against each other. The generator network tries to fool the discriminator network by generating data that appear to belong to $P(X)$, and the discriminator tries to distinguish between real images and fake images. The core concept of the GAN is applied to achieve domain adaptation. The pixel-GAN in [12] is a straightforward extension of the GAN for unsupervised domain adaptation. The domain adversarial neural network (DANN) [13] trained in a domain adversarial manner for image classification problem involving domain adaptation. In DANN, the features from the bottom layers are fed into two branches of the network. The first branch is a softmax classifier trained with the labeled source data. The second branch is a domain classifier trained to distinguish between the features of the source and the target. The key in DANN is the gradient reversal layer connecting the bottom feature extraction layers and the domain classifier. During back propagation, the gradient from the domain classifier is reversed when learning the feature extractor weights. In this way, the feature extractor is trained to extract domain invariant features.

Adversarial methods have shown remarkable performance in domain adaptation. However, in our problem setting, we will utilize statistic domain adaptation to matching cross domain images, and recognize the outliers in the training data of the target domain at the same time. Statistic method compares the sample distributions in a latent space. With this, we can form a loss function that could leverage the sample distributions for the matching of non-fully overlapping domains. Therefore, we convert the MMD loss to a weighted form to control that only the inliers can contribute to the domain adaptation. The weight is the probability of a target training sample to be an inlier. The probability is calculated by entropy loss of an outlier-inlier classifier.

References

[1] S. J. Pan and Q. Yang, *A survey on transfer learning,* IEEE Transactions on Knowledge and Data Engineering **22**, 1345 (2010).

[2] L. Bruzzone and M. Marconcini, *Domain adaptation problems: A dasvm classification technique and a circular validation strategy,* IEEE Trans. Pattern Anal. Mach. Intell. **32**, 770 (2010).

[3] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, *Learning and transferring mid-level image representations using convolutional neural networks,* in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014) pp. 1717–1724.

[4] 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,* (2013), arXiv:1310.1531 .

[5] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, *A kernel method for the two-sample-problem,* in *Advances in Neural Information Processing Systems 19* (2007) pp. 513–520.

[6] B. Sun and K. Saenko, *Deep CORAL: correlation alignment for deep domain adaptation,* (2016), arXiv:1607.01719 .

[7] M. Long, Y. Cao, J. Wang, and M. Jordan, *Learning transferable features with deep adaptation networks,* in *Proceedings of the 32nd International Conference on Machine Learning* (2015) pp. 97–105.

[8] 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* (2012) pp. 1097–1105.

[9] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, *Deep hashing network for unsupervised domain adaptation,* (2017), arXiv:2017 .

[10] M. Long, H. Zhu, J. Wang, and M. I. Jordan, *Unsupervised domain adaptation with residual transfer networks,* in *Advances in Neural Information Processing Systems 29* (2016) pp. 136–144.

[11] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets,* in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14 (2014) pp. 2672–2680.

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

[13] 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 , 1 (2016).

5 Outlier Detection

A lot of recent vision research works have exploited a massive number of images from the Internet as a source of training data for different learning tasks. A problem exists that images gathered via various sources are often noisy, which could compromise the learning model. Therefore, pruning the irrelevant images, *i.e.,* the *outliers*, becomes necessary.

5.1. Introduction to outlier detection

Anomalies or outliers can be caused by errors in the data but sometimes are indicative of a new, previously unknown, underlying process; in fact Hawkins [1] defines an outlier as an observation that deviates so significantly from other observations as to arouse suspicion that it was generated by a different mechanism.

Outliers detection [2, 3] is the process of identifying the new or unexplained set of data to determine if they are within the norm (i.e., inliers) or outside of it (i.e., outliers). Figure 5.1 shows the examples of the outliers in our datasets. For the *Shape* dataset, the outliers are images with single digits or alphabets. For the *Pitts-CycleGAN* dataset, the outliers are images of sky views or random meaningless city views of Pittsburgh. For the *Office* dataset, the outliers are two categories ('scissors' and 'speaker') chosen from the 31 categories of *Office* [4] dataset.

5.2. Unsupervised outlier detection methods

For our case, we train our network in an unsupervised manner. The target training data here have no label information. Thus, we need to consider unsupervised outlier detection for our problem.

Outliers detection can be portrayed in the context of one-class classification, which aims to build classification models. For example, one-class support vector machines [5, 6] are widely used, effective unsupervised techniques to identify outliers. In addition, there are many other existing works addressing this problem, such as Liu *etal.* [7] proposed a kernel-based method jointly learning a large margin one-class classifier and a soft label assignment for inliers and outliers. Chalapathy *etal.* [8] proposed an one-class neural network (OC-NN) model to detect anomalies in complex data sets. The model is an encoder-decoder architecture. Sabokrou *etal.* [9] investigated the adversarially learned one-class classifier for novelty detection, which also applied encoder-decoder architecture as part of their network.

Considering both cross domain image matching and outlier detection, it is hard to change our architecture into the encoder-decoder manner. Instead we inspire by the soft label assignment technique, jointly implement outlier detection and unsupervised cross domain image matching in an iteratively sample-reweighted way. We build an entropy loss to act as the inlier-outlier classifier for outlier detection and providing the sample weights.

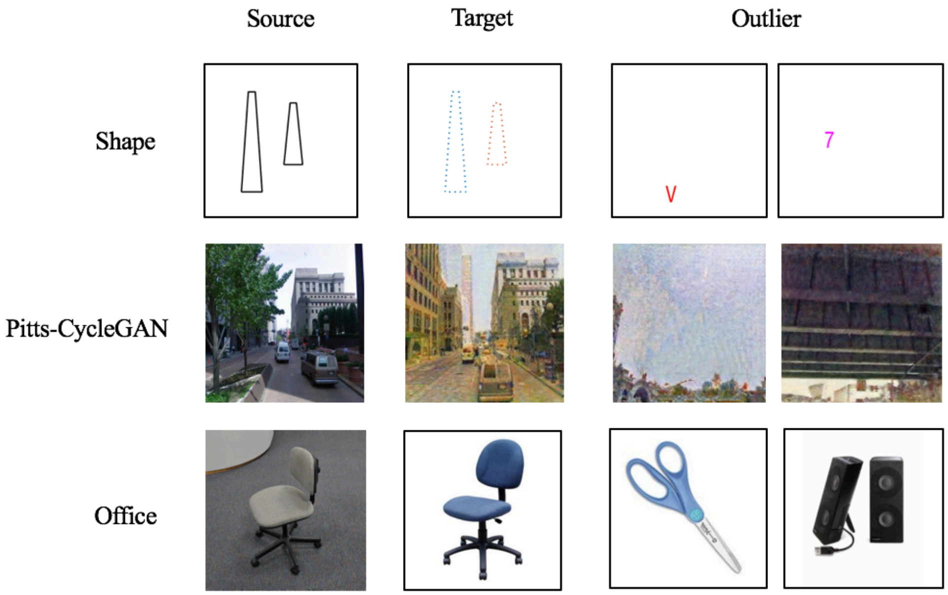


Figure 5.1: Sample images from *Shape, Pitts-CycleGAN and Office datasets*. The samples display the source domain, target domain and outlier images in our three datasets.

References

[1] D. Hawkins., *Identification of outliers,* (1980).

[2] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun, *Learning discriminative reconstructions for unsupervised outlier removal,* in *The IEEE International Conference on Computer Vision (ICCV)* (2015).

[3] C. You, D. P. Robinson, and R. Vidal, *Provable self-representation based outlier detection in a union of subspaces,* (2017), arXiv:1704.03925 .

[4] 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 (2010) pp. 213–226.

[5] B. Schölkopf and A. J. Smola, *Support vector machines, regularization, optimization, and beyond,* in *MIT Press* (2002).

[6] D. M. Tax and R. P. Duin, *Support vector data description,* Machine Learning **54**, 45 (2004).

[7] 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* (2014) pp. 3826–3833.

[8] R. Chalapathy, A. K. Menon, and S. Chawla, *Anomaly detection using one-class neural networks,* (2018), arXiv:1802.06360 .

[9] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, *Adversarially learned one-class classifier for novelty detection,* (2018), arXiv:1802.09088 .

6

Performance Evaluation Method

Our proposed method is evaluated by the ranked retrieval results of the query testing set. The most standard measure for evaluating ranked retrieval results is *Mean Average Precision* (MAP), which provides a single-figure measure of quality across recall levels. Among evaluation measures, MAP has been shown to have especially good discrimination and stability [1].

6.1. The interpolated precision

In a ranked retrieval context, appropriate sets of retrieved items are naturally given by the top ***k*** retrieved items. For each such set, precision and recall values can be plotted to give a *precision-recall* curve, such as the one shown in Figure 6.1. Precision-recall curves have a distinctive saw-tooth shape. If the $(k+1)^{th}$ item retrieved is nonrelevant then recall is the same as for the top k items, but precision has dropped. If it is relevant, then both precision and recall increase, and the curve jags up and to the right. It is often useful to remove these jiggles by an *interpolated precision*. The *interpolated precision* p_{interp} at a certain recall level r is defined as the highest precision found for any recall level $r^{'} \geq r$:

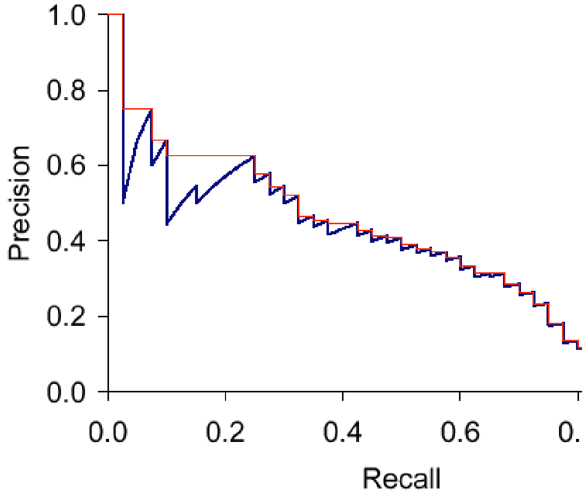


Figure 6.1: Precision-recall curve [2]

$$p_{interp}(r) = max_{r^{'} \geq r} p(r^{'}) \tag{6.1}$$

Interpolated precision is shown by a thinner dark blue line in Figure 6.1. With this definition, the interpolated precision at a recall of 0 is well-defined as 1.

Another benefit to use the *interpolated precision* in our experiments is that, we can use the mean of the interpolated precision over images in the testing set. In our evaluation, we need to evaluate our method on a set of query images, not only one. Thus, we need to calculate the arithmetic mean of the interpolated precision at that recall level for images in the test set.

6.2. Mean average precision

Mean average precision (MAP) provides a single-figure measure of quality across recall levels. For a single item need, Average Precision is the average of the precision value obtained for the set of top k items existing after each relevant items is retrieved, and this value is then averaged over the ground true items that should be retrieved. If the set of relevant items for an query $q_i \in Q$ is $\{I_1, ...I_{m_i}\}$ and R_{ik} is the set of ranked retrieval results from the top result until you get to item I_k, thus,

$$MAP(Q) = \frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{m_i}\sum_{k=1}^{m_i} Precision(R_{ik}). \tag{6.2}$$

When a relevant item is not retrieved at all, the precision value in the above equation is taken to be 0. For MAP, fixed recall levels are not chosen, and there is no interpolation.

When the relevant item expected to be retrieved is one, using MAP for such system evaluation is problematic. Calculated MAP scores normally vary widely across information needs when measured within a single system. This means that MAP is more suitable to the case, where a set of test information needs is large and diverse enough to be representative of the system effectiveness across different queries.

References

[1] J. Davis and M. Goadrich, *The relationship between precision-recall and roc curves,* in *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06 (2006) pp. 233–240.

[2] S. N. Group, *Evaluation of ranked retrieval results,* .

7

Additional Experiments

In addition to the experiments presented in the scientific paper, some additional experiments results are shared here.

7.1. Performance on noisy toy dataset

Similar to the *Shape* toy dataset mentioned in the scientific paper part, we also create a *Noisy Shape* toy dataset. Some samples from this dataset are shown in Figure 7.1. Instead of only drawing shapes with colored dot lines to represent the target domain, the *Noisy Shape* dataset also adds Gaussian white noise to the background. This makes the target domain more difficult to learn.

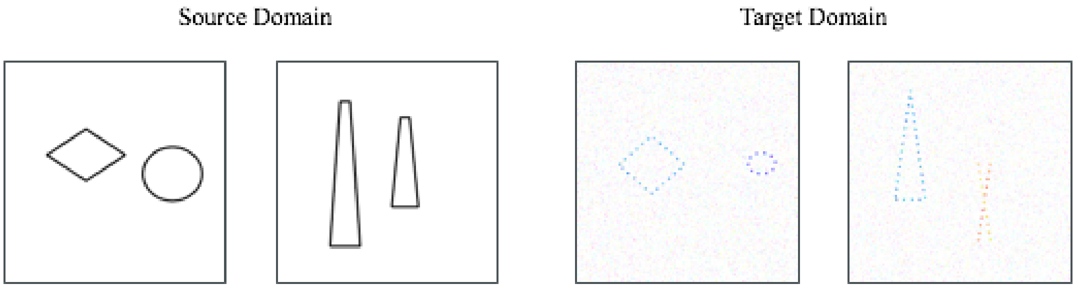


Figure 7.1: Samples from the source domain and the target domain of *Noisy Shape* toy dataset.

Domain difficulty We train the Siamese network on image pairs of the source domain and the target domain respectively to test the domain difficulty. Results in Table 7.1 show that the target domain is much more difficult than the source domain.

Dataset	*Source*	*Target*
Noisy Shape	0.950 ± 0.002	0.306 ± 0.001

Table 7.1: Source and target domain difficulty validation on Siamese network for *Noisy Shape* dataset. Mean average precision (MAP) for matching source domain query to source domain database $S \to S$, and target domain query to target domain database $T \to T$.

Unsupervised cross domain matching To investigate if our unsupervised cross domain image matching method also works well for the *Noisy Shape* dataset, we conduct experiments same as that in section 5.4 in the scientific paper. The learning objective again is *contrastive loss* $L(u_s)$ and *MK-MMD loss* $M(u_s, u_t)$,

$$min_u J = L(u_s) + \gamma M(u_s, u_t), \tag{7.1}$$

where, the u_s, u_t are feature representations of the source domain images and target domain images respectively.

The matching results are shown in Figure 7.2. It is easy to find that, even with a large γ value, the cross domain matching ($T \rightarrow S$) does not work better than the Siamese network only trained on the source domain data ($\gamma = 0$). It indicates that the domain adaptation method Mk-MMD loss may not be able to improve the performance of cross domain image matching when the dataset is too complex. This also illustrates that MK-MMD for domain adaptation is not robust enough for datasets in different complexity.

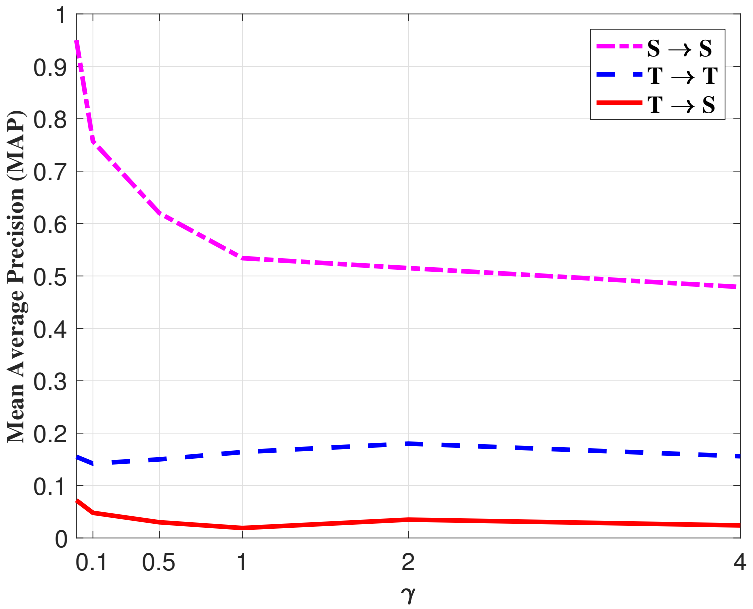


Figure 7.2: Matching performance of MAP under different γ parameters for MK-MMD loss term. $T \rightarrow S$ means matching target domain query image to source domain database.

References

[1] N. Ofir, S. Silberstein, D. Rozenbaum, and S. D. Bar, *Deep multi-spectral registration using invariant descriptor learning,* (2018), arXiv:1801.05171 .

[2] S. Chopra, R. Hadsell, and Y. LeCun, *Learning a similarity metric discriminatively, with application to face verification,* in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005) pp. 539–546.

[3] Y. Tian, C. Chen, and M. Shah, *Cross-view image matching for geo-localization in urban environments,* (2017), arXiv:1703.07815 .

[4] R. Gopalan, R. Li, and R. Chellappa, *Domain adaptation for object recognition: An unsupervised approach,* in *2011 International Conference on Computer Vision* (2011) pp. 999–1001.

[5] H. Venkateswara, S. Chakraborty, and S. Panchanathan, *Deep-learning systems for domain adaptation in computer vision: Learning transferable feature representations,* in *IEEE Signal Processing Magazine* (2017) pp. 117–129.

[6] T. Senlet, T. El-Gaaly, and A. M. Elgammal, *Hierarchical semantic hashing: Visual localization from buildings on maps,* in *22nd International Conference on Pattern Recognition* (2014) pp. 2990–2995.

[7] D. Costea and M. Leordeanu, *Aerial image geolocalization from recognition and matching of roads and intersections,* (2016), arXiv:1703.07815 .

[8] M. Divecha and S. Newsam, *Large-scale geolocalization of overhead imagery,* in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2016) pp. 32:1–32:9.

[9] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu, *Sketch2photo: Internet image montage,* in *ACM SIGGRAPH Asia 2009 Papers* (2009) pp. 124:1–124:10.

[10] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros, *Data-driven visual similarity for cross-domain image matching,* in *Proceedings of the 2011 SIGGRAPH Asia Conference* (2011) pp. 154:1–154:10.

[11] B. C. Russell, J. Sivic, J. Ponce, and H. Dessales, *Automatic alignment of paintings and photographs depicting a 3d scene,* in *ICCV Workshops* (2011) pp. 545–552.

[12] D. G. Lowe, *Object recognition from local scale-invariant features,* in *Proceedings of the International Conference on Computer Vision*, ICCV '99 (1999) pp. 1150–1157.

[13] Z. Wang, B. Fan, and F. Wu, *Local intensity order pattern for feature description,* in *Proceedings of the 2011 International Conference on Computer Vision* (2011) pp. 603–610.

[14] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid, *Local convolutional features with unsupervised training for image retrieval,* in *2015 IEEE International Conference on Computer Vision* (2015) pp. 91–99.

[15] F. Perronnin, J. Sánchez, and T. Mensink, *Improving the fisher kernel for large-scale image classification,* in *Proceedings of the 11th European Conference on Computer Vision: Part IV* (2010) pp. 143–156.

[16] 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,* (2013), arXiv:1310.1531 .

[17] X. Ji, W. Wang, M. Zhang, and Y. Yang, *Cross-domain image retrieval with attention modeling,* (2017), arXiv:1709.01784 .

[18] B. Kong, J. S. S. III, D. Ramanan, and C. C. Fowlkes, *Cross-domain image matching with deep feature maps,* (2018), arXiv:1804.02367 .

[19] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, Vol. 1 (MIT press Cambridge, 2016).

[20] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, *Backpropagation applied to handwritten zip code recognition,* Neural computation **1**, 541 (1989).

[21] 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* (2012) pp. 1097–1105.

[22] L. Bottou, *Large-scale machine learning with stochastic gradient descent,* in *Proceedings of COMPSTAT'2010* (Springer, 2010) pp. 177–186.

[23] T. Tieleman and G. Hinton, *Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,* COURSERA: Neural networks for machine learning **4**, 26 (2012).

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

[25] R. N. Bracewell and R. N. Bracewell, *The fourier transform and its applications,* (McGraw-Hill, 1986).

[26] *Lecture note of stanford cs231n convolutional neural networks for visual recognition,* (2018).

[27] 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* (2006) pp. 1735–1742.

[28] O. M. Parkhi, A. Vedaldi, and A. Zisserman, *Deep face recognition,* in *British Machine Vision Conference* (2015).

[29] T. Xiao, H. Li, W. Ouyang, and X. Wang, *Learning deep feature representations with domain guided dropout for person re-identification,* (2016), arXiv:1604.07528 .

[30] J. Zbontar and Y. LeCun, *Computing the stereo matching cost with a convolutional neural network,* (2014), arXiv:1409.4326 .

[31] S. J. Pan and Q. Yang, *A survey on transfer learning,* IEEE Transactions on Knowledge and Data Engineering **22**, 1345 (2010).

[32] L. Bruzzone and M. Marconcini, *Domain adaptation problems: A dasvm classification technique and a circular validation strategy,* IEEE Trans. Pattern Anal. Mach. Intell. **32**, 770 (2010).

[33] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, *Learning and transferring mid-level image representations using convolutional neural networks,* in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014) pp. 1717–1724.

[34] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, *A kernel method for the two-sample-problem,* in *Advances in Neural Information Processing Systems 19* (2007) pp. 513–520.

[35] B. Sun and K. Saenko, *Deep CORAL: correlation alignment for deep domain adaptation,* (2016), arXiv:1607.01719 .

[36] M. Long, Y. Cao, J. Wang, and M. Jordan, *Learning transferable features with deep adaptation networks,* in *Proceedings of the 32nd International Conference on Machine Learning* (2015) pp. 97–105.

[37] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, *Deep hashing network for unsupervised domain adaptation,* (2017), arXiv:2017 .

[38] M. Long, H. Zhu, J. Wang, and M. I. Jordan, *Unsupervised domain adaptation with residual transfer networks,* in *Advances in Neural Information Processing Systems 29* (2016) pp. 136–144.

[39] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets,* in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14 (2014) pp. 2672–2680.

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

[41] 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 , 1 (2016).

[42] D. Hawkins., *Identification of outliers,* (1980).

[43] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun, *Learning discriminative reconstructions for unsupervised outlier removal,* in *The IEEE International Conference on Computer Vision (ICCV)* (2015).

[44] C. You, D. P. Robinson, and R. Vidal, *Provable self-representation based outlier detection in a union of subspaces,* (2017), arXiv:1704.03925 .

[45] 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 (2010) pp. 213–226.

[46] B. Schölkopf and A. J. Smola, *Support vector machines, regularization, optimization, and beyond,* in *MIT Press* (2002).

[47] D. M. Tax and R. P. Duin, *Support vector data description,* Machine Learning **54**, 45 (2004).

[48] 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* (2014) pp. 3826–3833.

[49] R. Chalapathy, A. K. Menon, and S. Chawla, *Anomaly detection using one-class neural networks,* (2018), arXiv:1802.06360 .

[50] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, *Adversarially learned one-class classifier for novelty detection,* (2018), arXiv:1802.09088 .

[51] J. Davis and M. Goadrich, *The relationship between precision-recall and roc curves,* in *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06 (2006) pp. 233–240.

[52] S. N. Group, *Evaluation of ranked retrieval results,* .