Introducing shape priors in Siamese networks for image classification

Hiba Alqasir, Damien Muselet, Christophe Ducottet

Introducing shape priors in Siamese networks for image classification

Hiba Alqasir, Damien Muselet, Christophe Ducottet*

Université Jean Monnet Saint-Etienne, CNRS, Institut d'Optique Graduate School, Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France.

Abstract

The efficiency of deep neural networks is increasing, and so is the amount of annotated data required for training them. We propose a solution improving the learning process of a classification network with less labeled data. Our approach is to inform the classifier of the elements it should focus on to make its decision by supplying it with some shape priors. These shape priors are expressed as binary masks, giving a rough idea of the shape of the relevant elements for a given class. We resort to Siamese architecture and feed it with image/mask pairs. By inserting shape priors, only the relevant features are retained. This provides the network with significant generalization power without requiring a specific domain adaptation step. This solution is tested on some standard cross-domain digit classification tasks and on a real-world video surveillance application. Extensive tests show that our approach outperforms the classical classifier by generating a good latent space with less training data. Code is available at `https://github.com/halqasir/MG-Siamese`

Keywords: Deep learning, Siamese networks, Shape prior, Domain generalization, Proxy-based learning

1. Introduction

Recently, there has been a growing interest in using Siamese networks on visual tasks in the context of few-shot learning or self-supervised learning [1, 2]. The principle of Siamese networks is to learn a high dimensional latent space where similar images are projected near each other and dissimilar ones are projected far from each other. They have the structure of two twin networks that accept distinct inputs and are joined by a contrastive loss to compare the two outputs. Once the Siamese network has been trained, the class prediction can accurately be done with a nearest

*Corresponding author
Email address: ducottet@univ-st-etienne.fr (Christophe Ducottet)

neighbor classifier operating in the latent space [3]. Siamese networks can also be used in the context of unsupervised training, where the two inputs are one image and a transformed version of this image. The resulting latent space gets interesting invariance properties and provides powerful image features that standard classifiers can exploit.

In this paper, we explore the potential of Siamese networks in an original image classification setting where, in addition to the training images, we have a shape prior (used as a proxy) given by a particular binary mask for each class. This is the case when each class can be represented by an object having a discriminant shape easily described by a binary mask. The purpose of the shape prior is to highlight the most important parts of the images for solving the classification task. Typically, the binary mask can be drawn by a human using image editing software by delineating the object in a selected training image characteristic of the class. In numerous applications, this kind of prior is easy to obtain because it corresponds to the general shape of the class. For example, in digit recognition it is the basic shape of the digits. In sign language recognition, it is the prototype of each sign to recognize. In this paper, we also consider a real-life application in the field of video surveillance where the goal is to determine whether people boarding a chairlift have properly set their safety bar[1]. In this case, the mask corresponds to the shape of the safety bar both in open or closed positions. Figure 1 shows a few examples of binary masks and the respective image instances of the corresponding class. In such applications, it is clear that the mask is only a coarse representation of the geometry of the important element. Its shape, location and orientation are approximated. Furthermore, it is worth mentioning that a single binary mask is expected for each class, limiting the effort of the user to introduce shape priors.

Figure 1: Examples where binary masks can help for classification. Note that, for our approach, one mask is expected for each category, not for each image.

In this context, we propose a Mask-Guided Siamese approach (referred to as MG-Siamese below) built upon the combination of Siamese network architecture and specific training and testing configurations exploiting the binary masks and the ability of Siamese networks to learn relevant latent spaces. More precisely, each binary mask (from one domain and one category) is used as a proxy in the metric learning process so that all the images from the same category and the same domain concentrate around this mask in the embedding space. This is a smart solution

[1] in collaboration with Bluecime company as part of Mivao project

to remove the irrelevant background from the final features. We show that this approach provides a simple yet effective solution for two important problems in image classification: (i) inserting shape priors in a deep architecture in order to guide the network towards the relevant features and so reducing the amount of training data; (ii) generalizing well on a new domain with only some shape priors or even without any information from the target domain.

The most important finding of this paper is to show that using binary masks as simple shape priors in Siamese networks helps the model generate good latent space with less training data and that this space is robust to slight distribution discrepancy between the source and target data. In case of more considerable discrepancy, the user just needs to provide the shape priors of the new data, with a single binary image, in order to get good results. These properties are studied experimentally using three different experiments in the context of digit classification and chairlift safety.

In a recent study, we have shown that shape priors can be introduced in Siamese architectures in order to boost the performances on large-scale datasets in the context of intra-domain experiments [4]. This paper goes a step further by assessing the quality of such an approach trained with small datasets from different domains. Furthermore, we run more experiments and provide a deep analysis of the results. Indeed, we experimentally assess the impact of each element of our framework and answer important questions:

1. What kind of prior is it better to introduce in the network?
2. Should we learn domain invariant features or domain-specific features to improve the generalization property of the solution?
3. Should we train our Siamese network with a pair-based or a proxy-based approach?
4. What is a good proxy for proxy-based learning?

Our contributions are multiple:

- We challenge the generalization properties of a Siamese network learned on a small source dataset over a new unseen target dataset, *i.e.*, domain generalization.
- We propose a solution to exploit shape priors at test time when the target masks are available.
- We run extensive tests and show that our solutions always outperform the classical classification models.

The rest of the paper is organized as follows. Section 2 is a review of the related works, section 3 presents our MG-Siamese approach to introduce shape prior and explains different generalization scenarios. Section 4 describes experiments and results for three different contexts to evaluate the domain generalization capability of MG-Siamese. Finally, section 5 concludes with a summary of the method and main results.

2. Related Works

The works related to our paper deal with the insertion of priors in the deep architecture and domain generalization are detailed below, emphasizing the approaches based on Siamese architectures.

2.1. Siamese networks

The general purpose of Siamese networks is to provide a suitable embedding space to compare two inputs [5]. They have thus naturally been used in various contexts as object tracking [6], face or signature verification [7, 5], or few-shot learning [3]. They recently became a crucial component in the fields of unsupervised/self-supervised visual representation learning [8, 9, 1, 2]. The principle relies on an unsupervised pretext task involving comparing two images to learn a representation space that can further be used for another task. A simple example of a pretext task is determining if the two inputs are augmented versions of the same image, or if two image crops come from the same original image. In this work, we propose an original setting where one of the two inputs is a proxy image used as a shape prior, and we study the generalization properties of this solution in the context of a small training set and domain generalization.

A part of related works concerns the use of Siamese networks for comparing multimodal images. Indeed, by providing pairs of images as input and designing specific losses, Siamese networks are smart solutions to compare patches from different modalities (color, infra-red, thermal, sketch, etc.). When the two sub-networks share their weights, the idea is to extract features that are common to the two modalities, while when the two sub-networks are different (pseudo-Siamese network), the aim is to discover the features specific to each modality. En *et al.* [10] propose to exploit the benefits of these two approaches in a single three-stream network.

2.2. Domain generalization

In real-world applications, it is not rare that a domain shift occurs between the training data and the test data. This shift can result from many factors such as variations in background, locations, viewpoints, lighting conditions, acquisition devices, modalities, or image quality. A key challenge is to bridge the gap between the domains to the extent that a system trained on a source domain will generalize well to a target domain.

When (unlabeled) target data is available while training the network, domain adaptation can be applied with interesting results [11, 12]. When only one labeled example of the target (domain or task) is given, we refer to one-shot domain adaptation. Liu *et al.* [13] proposed to address the gap between source and target tasks in one-shot learning by filtering some features that could harm the target task. The authors' filtering process is straightforward; they randomly select a binary mask and multiply it by the embedding. They make several attempts and finally choose the mask that yields the best accuracy.

When no target data is available at learning time, people resort to zero-shot domain adaptation [14, 15, 16, 17] or domain generalization [18, 19, 20]. For zero-shot domain adaptation, Yang and Hospedales propose associating each domain (source and target) with a vector of discrete parameters called a semantic descriptor [17]. Then, they use a two-branches network whose inputs are the sample features and the domain descriptor from this sample. The output of the network (the predicted sample class) is a fusion of the outputs of the two branches. This is a way to adapt the classifier to the domain provided as input. For a new unseen target domain, given its semantic descriptor, the network is able to accurately predict the class of its samples. This approach requires the user to be able to describe all the domains with a vector of discrete parameters. Kumagai and Iwata use a similar architecture, but instead of using an attribute vector for each domain, they propose to extract a latent domain vector from the set of features of each domain [15]. This approach requires the knowledge of the whole target features in order to start predicting the classes of the target data.

Zhou *et al.* propose to solve the problem of domain generalization by mapping source learning data into synthesized data from unseen domains [19], using a domain transformation

network (DoTNet). This is accomplished with a learning objective formulated to minimize label classification loss while maximizing domain classification loss using domain adversarial training. Ghifary *et al.* present a robust feature learning technique using auto-encoders that improve cross-domain generalization properties [20]. Their multi-task strategy suggests that auto-encoders are not only a robust framework for unsupervised feature learning, but also for invariant feature learning. Motiian *et al.* [21] seek to align marginal distributions across domains using a Siamese architecture, making the distance (in the embedding space) of samples from different domains with the same labels closer while those with different labels are further apart. This approach is similar to ours since it is based on a Siamese architecture, but they consider pairs of images from different domains, while we use image/mask pairs from the same domain. Many studies have compared the proxy-based solutions and the pair-based solutions for metric learning, and it appears that the main weakness of the latter approaches is that they suffer from a high training complexity because of the huge number of image pairs [22]. Our image/mask pairs are a good solution to cope with this problem. Second, training a network to be invariant to any eventual domain shift requires large-scale datasets with many domains. In our context, where only a small set of labeled data is available, we notice that it is much better to learn specific features for each domain instead of invariant features for all the domains. This is the solution we are proposing by using different masks for different domains. The advantages of our approach over [21] are experimentally shown in the experiments section.

Unlike other solutions that use domain features to predict labels, Shankar *et al.* propose to learn a domain invariant classifier [16] using multi-domain training data to generalize to unseen domains. Their original solution consists of artificially transferring data between domains to learn invariant features, and they show that this augmented data helps to generalize to unseen domains better. The new domain-guided samples are created by augmenting the original samples with their gradient from a domain classifier.

In our case, the problem is slightly different from all the previous approaches since we have additional geometric features that are domain-dependent and that can be inserted into the model. For this reason, our solution is complementary to all these solutions and could help to improve their results if geometric priors are available.

2.3. Inserting priors

The insertion of geometric features in a deep neural network has not been much studied in the literature. Conditional networks could be interesting solutions to exploit such available data [23, 24]. For example, Zhao and Snoek propose modulating the RGB features of a video with optical flow features in order to improve the action detection accuracy. The proposed motion condition and motion modulation layers incorporate motion and modulate the contribution of the RGB features. Such conditional networks require the different features (optical flow and RGB) to be well spatially registered, which is not the case for our images and binary masks.

One solution to guide a classification network towards the relevant elements of a category is to transform it into a detection network, such as Faster R-CNN [25], CornerNet [26], or DetectoRS [27]. Indeed, providing bounding boxes around the relevant elements at training time is a good solution to help solve the classification task [28]. Given all the elements detected in one test image, a decision rule can be used to deduce the category of this image. In our experiments, we will show that detecting objects can indeed boost the classification results. Unfortunately, inserting bounding boxes priors requires a considerable effort since several boxes have to be provided for each training image. Our solution reduces this effort significantly since only one binary mask is required for each category and not for each image.

The attention mechanism also gives the network the ability to focus on a subset of its inputs or features rather than processing the whole image at once, allowing regions of interest to be localized and the scene to be analyzed selectively. These mechanisms have been applied to a wide variety of tasks; one approach that could be related to ours deals with a person re-identification task [29], where the idea is to help the network to extract features only from the body of the person in the image and not from the cluttered background. In this aim, the authors propose to use a binary mask of the person to create three images: the full image, the body image, and the background image. Then a triplet loss is used to bring the features of the full image closer to those of the body alone, and to move away the features of full images from those of the background image. Thus, the network is trained to automatically extract the most important features (*i.e.*, from the body only) from the full image. This approach requires designing a triplet loss to extract features from the body but also a Siamese network in order to bring closer images from the same person and move away images from different persons. This complex architecture is not adapted to our problem with a small set of labeled images and requires a perfect match between the binary mask and each image.

In this paper, we propose to use Siamese networks to enforce the model to concentrate on specific features in the images that are essential for the classifier to make its decision. We show that these geometrical constraints are also very interesting to generalize to new unseen domains without any specific adaptation step. The approach is presented in the next section.

3. Mask-Guided Siamese approach

To insert shape priors in our model, we resort to a Siamese architecture that learns a mapping that projects the images and the masks into a feature embedding space, each mask corresponding to the specific class to be tested [30]. Euclidean distances are evaluated in the feature space at test time to decide to which class the input image belongs.

In this work we address a classification problem with a significant shape prior in the classes. This is the case when each class can be represented by an object with a discriminant shape easily described by a binary mask. Furthermore, we assume that the general shape does not change from one instance of the class to the next so that the binary mask remains specific to the corresponding class. Moreover, we assume a reasonable number of classes (several dozen at most) to enable an expert to choose a binary mask for each class. In practice, the binary mask can be drawn by the expert using an image editing software by roughly delineating the object in a selected training image characteristic of the class.

Formally, we consider a set of domains D split in source and target domains denoted D^S and D^T, respectively, such as $D = D^S \cup D^T$. All these domains share the same set of classes Y. Each source domain $d \in D^S$ comes with a set of masks $M_d^S = \{m_{d,y}^S\}_{y \in Y}$ where $m_{d,y}^S$ is the mask associated with class y in domain d. Similarly, each target domain $d \in D^T$ comes with a set of masks $M_d^T = \{m_{d,y}^T\}_{y \in Y}$ where $m_{d,y}^T$ is the mask associated with class y in domain d. Finally, we have two sets of source and target images denoted X^S and X^T respectively, with $X = X^S \cup X^T$, and each image is associated with a class $y \in Y$ and a domain $d \in D$. Following our previous conventions, the set of images of the source domain $d \in D^S$ is denoted X_d^S and the set of images of the target domain $d \in D^T$ is denoted X_d^T.

Figure 2 shows a simplified view of our classification problem, in which we have, for each source domain $d \in D^S$, a set of training images X_d^S and a set of binary masks M_d^S, such that each class is represented by one mask in its specific domain. For the sake of clarity, we illustrate the

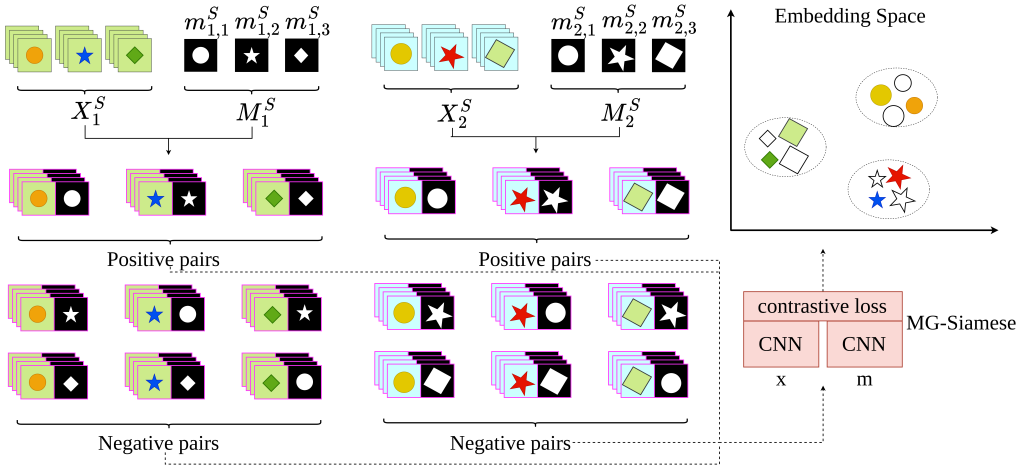


Figure 2: The proposed training process on source domains with image/mask pairs. The training set comprises two source domains (green and blue background respectively), with $|Y| = 3$ categories (circle, star or diamond). Each domain i comes with a set of images X^S_i and a set of masks M^S_i which are used to generate positive and negative pairs. Each pair feeds the MG-Siamese. Thanks to the contrastive loss, the output of the MG-Siamese ($F()$) for each input (image or mask) is a class-specific discriminative feature vector. It is preferable to use color for this figure in print.

process in this figure with three different classes and two different source domains, although our model can be trained with any number of classes or domains.

3.1. Training the model

A Siamese network, composed of two sister CNNs sharing their weights, is used at training time. Training pairs from $\{X^S_d \times M^S_d\}_{d \in D^S}$ are fed to the Siamese network, each pair is composed of an image x and a mask m of the same source domain $d \in D^S$. Each pair is labeled positive or negative, such that a positive pair is composed of an image and a mask of the same class and a negative pair is composed of an image and a mask of different classes. Each sister CNN is learned so that the two inputs x and m are transformed into two vectors that will be similar if they are from the same class, and different if they are from two different classes. Denoting F the function mapping the input image (or mask) to the corresponding output vector, the two outputs $F(x)$ and $F(m)$ are compared through a contrastive loss function $\mathcal{L}$ defined by [31]:

$$\mathcal{L}(F(x), F(m)) = \alpha \|F(m) - F(x)\|^2 + (1 - \alpha) \max(1 - \|F(m) - F(x)\|, 0)^2 \qquad (1)$$

where $\|.\|$ denotes the L_2 norm, $\alpha = 1$ for a positive pair, and $\alpha = 0$ for a negative one.

Minimizing this loss forces the CNN to extract features from the images that are similar to the features from the corresponding mask of the same class. This is a smart way to inform the network of the element it needs to focus on in order to classify the image. By learning this model over multiple domains and projecting all source images and masks into the same embedding space, the model learns to extract automatically class-specific discriminative features of the images. The following sections present different inference processes depending on the domain of the tested data (source or target), and on the available masks.

3.2. Testing on source data

In this section, we consider that test images come from a given source domain $d \in D^S$ for which the binary masks representing the shapes priors are available. Considering one test image $x \in X_d^S$, we can use one branch of the trained Siamese network and feed it with each mask $m_{d,y}^S$ from M_d^S. In total $|Y|$ masks (where $|Y|$ denotes the number of elements of Y) are available as illustrated on Figure 3. The corresponding outputs are denoted $F(m_{d,y}^S)$.

Given a test image x, its inferred class $\hat{y}$ is the class of the nearest mask, in terms of Euclidean distance, in the embedding space:

$$\hat{y} = \arg\min_{y \in Y} \|F(m_{d,y}^S) - F(x)\|^2. \tag{2}$$

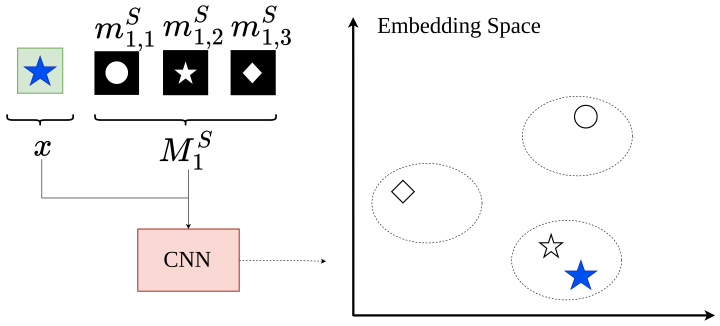


Figure 3: Test on a source image when the corresponding source masks are available. The image x and the set of masks M_1^S, which all belong to the same source domain $d_1^S \in D^S$ are fed to one branch of the model. Then the output in the embedding space indicates the class of the image.

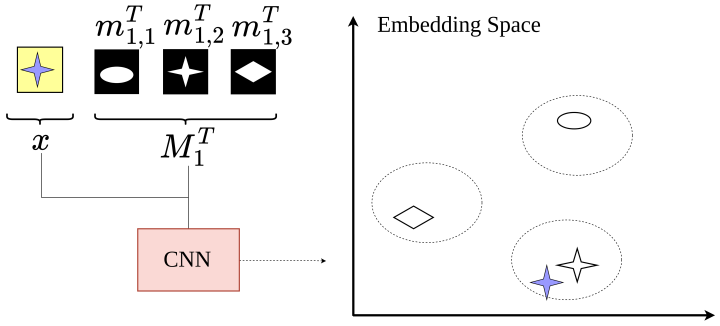


Figure 4: Test on a target image when the corresponding target masks are available. The image x and the set of masks M_1^T, which all belong to the same target domain $d_1^T \in D^T$ are fed to one branch of the model. Then the output in the embedding space indicates the class of the image.

3.3. Generalizations scenarios

In this section, we consider the case where the test images come from a target domain, *i.e.*, a domain that has not been used to train the model. Two different cases are considered: (i) the corresponding target masks are available; (ii) the corresponding target masks are not available. It is worth mentioning that in both cases the target data distribution is not available and is not used at training time, unlike classical domain adaptation approaches.

Testing on target data with target masks

In this section, we assume that for each target domain, the masks are available. In this case, the solution is straightforward. Considering the given target domain $d \in D^T$, we feed the network with the masks $M_d^T = \{m_{d,y}^T\}_{y \in Y}$ as well as the test image $x \in X_d^T$. Then, the Euclidean distances between the image feature vector and the masks feature vectors allow us to infer the class of this test image (see Figure 4):

$$\hat{y} = \arg\min_{y \in Y} \|F(m_{d,y}^T) - F(x)\|^2. \tag{3}$$

Where $F(x)$ and $F(m)$ are the output vectors of the image and the mask, respectively. This solution fully exploits the advantages of our learning process that allows learning domain-dependent discriminators by comparing each image only with its corresponding masks while projecting all the feature vectors in the same embedding space so that the features of different domains can help to improve each other.

Testing on target domain without target masks

In this section, we consider the most challenging case, where we would like to infer the class of images from a target domain for which we do not have any information *i.e.*, neither the image distribution in the embedding space, nor the binary masks informing about the shape priors.

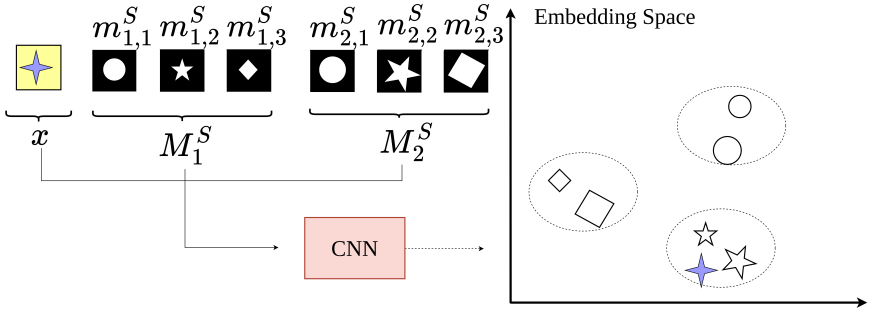


Figure 5: Test on a target image when target masks are not available. The image which belongs to the target domain $d_1^T \in D^T$ and the two sets of masks M_1^S and M_2^S, which belong to the source domains d_1^S and d_2^S respectively, are fed to one branch of the model. Then the output in the embedding space indicates the class of the image.

In this case, we propose exploiting the diversity of the source data and assuming that each (unknown) target mask is similar to a source mask. Since the embedding space has been learned on several source domains, this assumption is highly likely to be true in practice. Thus, we propose to project the target image $x \in X^T$ and all the source masks M_d^S into the embedding space. Then, by looking at the nearest neighbors, we derive the class of the target image as (see Figure 5):

$$\hat{y} = \arg\min_{(d,y) \in D^S \times Y} \|F(m_{d,y}^S) - F(x)\|^2. \tag{4}$$

The experiments confirmed our assumption that the feature vectors of the source masks could provide an approximation of the location of the feature vector of any new unseen target example, even in case of a high source/target distribution discrepancy. We will show that, although this is not a constraint in our learning process, the network automatically discovered that masks in one class share geometric properties that help to discriminate them from masks in other classes.

4. Experiments

We carried out experiments on three different contexts to evaluate the domain generalization capability of our proposed system. After presenting the datasets used in these contexts and the associated experimental settings, we will answer four questions about the best priors, the best features for generalization, and proxy-based metric learning. Then some geometric interpretations will be conducted before concluding this work.

4.1. Datasets and settings

The performances of our approach are assessed in three different contexts with different datasets. The experimental setting is adapted to each context for a fair comparison. The experiments range in their degree of difficulty and nature from the authentic chairlift safety problem to the traditional digit classification problem, with the intent of evaluating the proposed approach in different settings and number of classes, however in all cases we address the problem of domain generalization for datasets with non-deformable objects where shape priors could be obtained easily with almost no cost.

4.1.1. Context 1: chairlifts safety problem

The safety of chairlifts is a major concern for ski resort operators. To prevent possible accidents, it is necessary to detect dangerous situations on chairlifts as early (after boarding) as possible. As part of an authentic project to develop a system to provide a thorough analysis of the boarding scene, one but not the only aspect is to detect users who have not properly closed the safety bar before leaving the boarding station. A major challenge of the project is that the system must learn to configure itself automatically, taking into account new situations or types of chairlifts not previously recognized. Without the need for a costly domain adaptation step or any additional annotated data.

Our chairlift dataset[2] consists of images from 21 different chairlifts acquired using the following process. For a given chairlift, several video recordings are first made in the ski resort in real conditions. Then, each video is pre-processed to extract a set of frames containing the passage of a single chairlift carrier. Then, each image is registered and cropped according to a reference image to have the chairlift carrier roughly at the same 2D position, scale and orientation. As we can see in the example images in Figure 6, there is a great diversity between chairlifts: 3D geometry of the carrier, number of seats, viewpoints, weather conditions, background, ... The images are labeled "open" or "close" according to the position of the safety bar of the chairlift carrier.

Data setup. As we want to test the ability of our model to generalize, we randomly chose 6 chairlifts to be the target domains $D^T = \{d_1^T, ..., d_6^T\}$ and the remaining 15 chairlifts to be the source domains $D^S = \{d_1^S, ..., d_{15}^S\}$. Given that the annotation step is very time-consuming, we chose a challenging setting with very little labeled data. Specifically, only 20 random images from each of the 15 source chairlifts are labeled as open or closed, resulting in 300 training images in total. The number of test images is different depending on the target chairlift, as shown in Table 1.

[2]provided by Bluecime company

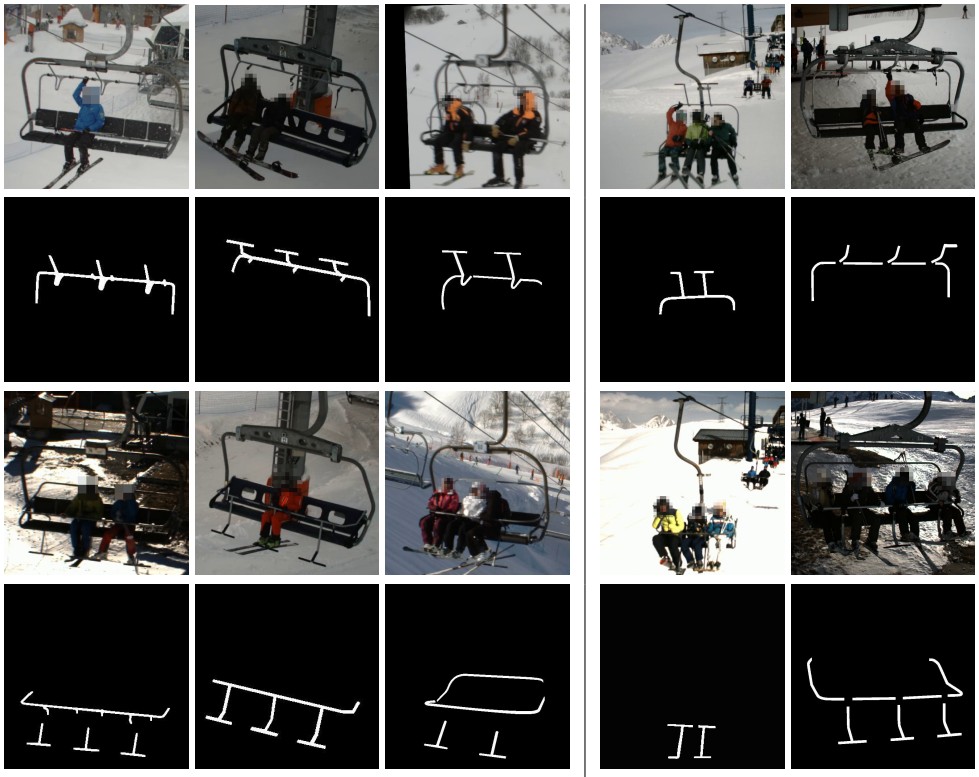
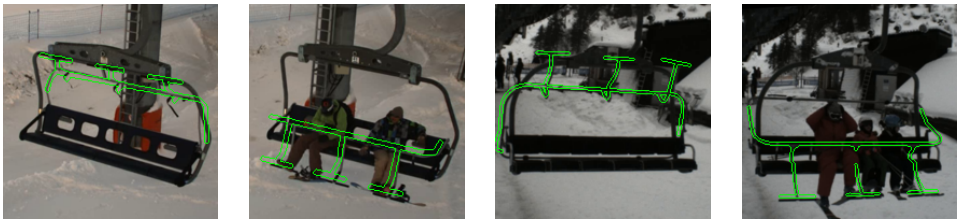


Figure 6: Some source and target images and the corresponding chairlift masks (experimental context 1). Each column contains images and masks from the same chairlift (top:open and bottom:closed). Note that each chairlift stands for one domain, where there are many images and only one mask per class. One may notice the diversity between chairlifts (domains): point of view, background, 3D geometry of the carrier, number of seats, etc.

Masks generation. Each chairlift (domain) comes with two binary masks (open and closed) that were easily drawn from one image of each category. Since the safety bar is a non-deformable object which is always observed with the same viewpoint for a given chairlift, we can create two binary masks that represent the chairlift carrier shape when it is open (open mask) and when it is closed (closed mask). Each time a new chairlift is installed, the operator can easily create these two mask images by acquiring one image of each class (open and closed) and by drawing two binary masks representing the shape of the safety bar.

The purpose of the masks is to have a rough idea of the geometric constraints of the object of interest, it is not needed that they are perfect or superimposed on the real object in the RGB image, and this is a strength of the proposed method, as it does not require expensive perfectly segmented masks, instead it is the network's job to make use of the features extracted from the masks and those extracted from the image and to match them in the embedding space. Figure 7 shows that the open (resp. closed) mask of a chairlift carrier is not perfectly superimposed with all the open (resp. closed) images of this chairlift carrier. But it gives a coarse idea about the shape of the safety bar and its relative location in the image.

The open mask of chairlift d_2^T superposition with 3 different open images of d_2^T.



Open/closed masks superposition with open/closed images from chairlifts d_{11}^S (on the left) and d_{14}^S (on the right).

Figure 7: Superposition of masks and the corresponding images.

Network and training settings. Our Siamese architecture has two identical networks with shared weights, which are composed of the convolutional part of VGG16 [32] pre-trained on Imagenet [33] and two randomly initialized fully connected (FC) layers. The first FC layer has 4096 outputs and the second one has 1024. The networks are trained using back-propagation and the stochastic gradient descent algorithm with a learning rate equal to 10^{-5}, a decay equal to 10^{-8}, and a momentum equal to 0.9. All tests are averaged over 10 runs, each of them consisting of 1000 epochs. As explained in the previous section, the input of the Siamese model is an image/mask pair, both belonging to the same chairlift. A positive pair is composed of an image and a mask of the same class (open or closed), and a negative pair is composed of an image and a mask of different classes. At test time, we apply a K-Nearest Neighbors (KNN) classifier on the source or target masks, depending on the experiment. All our tests are compared to a classical binary classifier that uses the same architecture (augmented with a two-neuron classification layer) for a fair comparison. Since the weights are shared between the two sister networks in our Siamese architecture, the number of parameters of our solution is almost the same as that of the binary classifier (the latter having 2048 more weights).

Table 1: The number of test images in each class in the target chairlifts (experimental context 1).

class	d_1^T	d_2^T	d_3^T	d_4^T	d_5^T	d_6^T
open	12	438	408	283	389	125
closed	385	462	277	302	329	62

4.1.2. Context 2: cross-domain digit recognition

Several digit datasets exist today with some important differences between them; in this experimental context, we consider four digit datasets: MNIST, MNISTM, USPS, and SVHN [34]. Since they represent images of the same categories, i.e., digits between 0 and 9, they can be used for cross-domain classification. Even though this problem seems to be very basic, it is important to employ such a classical task to validate the effectiveness of our proposed method on more widely known databases.

Data setup. As shown in Figure 8, MNIST and USPS are grayscale handwritten digits datasets. MNISTM is a colored version of MNIST created artificially using image patches from the colored BSD500 dataset [35]. SVHN dataset contains colored images of street numbers; in its cropped digits version, only one digit is labeled and it is usually located in the center. The number of training and test images in each dataset is shown in Table 2.

Table 2: The number of training and test images of the four digits datasets (experimental context 2).

dataset	MNIST/MNISTM	USPS	SVHN
training images	60000	7291	73257
test images	10000	2007	26032

Figure 8: Some images from the four digits datasets (experimental context 2).

Masks generation. From MNIST dataset, we randomly selected one image per class and considered these 10 images as the masks for this experimental context. Again, these masks do not have to be perfectly segmented and they do not have to superimpose all objects of the same category, as it is the job of the network to match the features of the RGB images and the binary masks in the embedding space.

Network and training settings. For this experiment, we follow the settings of [34]. All the images are rescaled to $32 \times 32 \times 3$. The backbone architecture is composed of 9 convolutional layers with several dropout, Maxpooling, and global pooling layers, and one fully connected layer serves as a classifier.

In this experiment, we consider 5 tasks, by learning the model on a single dataset (source domain) and testing it on a different dataset (target domain). This setting is challenging for domain generalization since only one domain is used at training time.

4.1.3. Context 3: cross-domain digit recognition with Rotated MNIST

Another classical task in the field of domain generalization is digit recognition with Rotated MNIST, which we will elaborate in the following.

Data setup. Different domains can be artificially created by applying geometric transformations, such as rotations. This is the case of the rotated MNIST dataset used in [21]. There, a set M_{0° of 100 images per class (1000 in total) was randomly sampled from MNIST dataset. Then five new domains M_{15°, M_{30°, M_{45°, M_{60°, and M_{75° were generated by rotating the original images by 15, 30, 45, 60, and 75 degrees, respectively. Sample images are depicted in Figure 9.

Masks generation. From MNIST dataset, we randomly selected one image per class and considered those 10 images as the masks for M_{0°. And then, for each rotated set, we apply a rotation with the same angle to get the masks of each domain. For example to obtain the masks of the rotated set M_{15°, we apply a 15° rotation on the 10 masks of M_{0°.

M M_{15° M_{30° M_{45° M_{60° M_{75°

Figure 9: Some image examples from the Rotated MNIST (experimental context 3).

Network and training settings. We carried out 6 cross-domain experiments. In each, we omit one domain, considered the target domain, and train the model using the remaining 5 sets. The input of the Siamese model is an image/mask pair, both belonging to the same set (same rotation degree). A positive pair is composed of an image and a mask of the same class, and a negative pair is composed of an image and a mask of different classes. For this experimental context, each sister network in our Siamese architecture is similar to the one proposed in [21], which is composed of 4 convolutional layers and one fully connected layer.

4.2. Tested approaches

The experiments aim to measure the impact of the different choices we made in our final solution. Since the heart of our work is the insertion of shape priors in a deep model, we cannot compare with the domain generalization state-of-the-art approaches that do not make use of these priors. We just keep in mind that our solution is complementary to most of the other domain generalization approaches. In this section, we rather propose to test many alternatives on the three experimental contexts, for which we run our own code for a fair comparison. These alternatives use different architectures, priors, or metric learning approaches. Here is the list of the tested methods with their detailed features, as summarized in Table 3:

- The **Baseline** is a branch of the two twin networks used in our **MG-Siamese** solution. Its architecture is based on different experimental contexts, as mentioned before. Since the weights are shared between the twin networks in our **MG-Siamese**, the number of learned parameters in the **Baseline** is slightly higher because it is augmented with a two-neuron classification layer (see above).

- Our mask-guided Siamese solution is called **MG-Siamese_S** or **MG-Siamese_ST**, depending on whether it exploits only source masks or source and target masks. It is worth mentioning that, for our method **MG-Siamese_ST**, the target masks are used only at test time and not at training time. This means we need to train a single network for all the domains and exploit the target masks during the inference.

- The same architecture as ours is employed for the approaches **Siam_Intra** and **Siam_Inter**. The difference is that these latter are trained with pairs of images without exploiting the binary masks, whereas our approach is trained only with image/mask pairs. Thus **Siam_Intra** and **Siam_Inter** learn their embedding space through a pair-based metric learning approach, while **MG-Siamese_S** and **MG-Siamese_ST** use proxy-based training [22]. The difference between **Siam_Intra** and **Siam_Inter** is that the former constitute pairs with images from the same domain only, while the latter use also images from different domains in each pair. The idea is to improve the invariance property of the embedding space by mixing the domains [21].

- In order to disentangle the impacts of the proxy-based approach and this of the shape priors insertion, we propose to test the approach **Siam_Intra_Proxy**, which is similar to **Siam_Intra** but use a proxy-based training where the proxies are images instead of masks. Thus, for this approach, one image is randomly selected in each domain and each category and used as a proxy in the training. Since the random selection of the proxy can impact the results, we have run many tests for this approach and averaged the results.

- One important question raised in this work is to know if some other priors can be introduced in the network to guide the classification. When some small elements in an image are very relevant for the final task, one solution could consist in detecting these elements with classical object detection network such as Faster R-CNN [25]. For our experiment on the chairlift dataset, we have trained such a network to detect "open" and "close" safety bars. At inference time, the object detected with the highest score in one image provides the whole image category. This solution is called **Faster R-CNN**. Of course, it requires many bounding boxes at training time, unlike our shape priors that requires one binary image for each category.

- The solution **Siam_Inter** is very similar to the approach proposed in [21], called **CCSA**, but this latter is trained with two losses, namely the classical contrastive loss for image pairs (positive and negative) and a classification loss. We did not run experiments ourselves with this approach but just reported the results from [21] as it is also based on Siamese architecture for generalization.

4.3. Results

Tables 4, 5, and 6 display the results obtained in the three experimental contexts. We concentrate the main analysis on the context of chairlifts safety problem, because the masks are very diverse across domains, whereas the variations are lower for the two other contexts. Thus we do not need to insert target prior at test time for these latter.

From these three tables, many comments can be made, and we propose to organize them by answering the following questions.

1. Are the provided binary masks good priors?

Method	Domain features	Training					Test
		Input	Prior	Metric learning	Proxy	Used masks	Used masks
Baseline	Invariant	Image	Ø	Ø	Ø	Ø	Ø
Siam_Intra	Specific	Image/Image pair Same domain	Ø	Pair-based	Ø	Ø	Ø
Siam_Inter	Invariant	Image/Image pair Different domains	Ø	Pair-based	Ø	Ø	Ø
Siam_Intra_Proxy	Specific	Image/Image pair Same domain	Ø	Proxy-based	Image	Ø	Ø
Faster R-CNN	Invariant	Image	Bounding boxes	Ø	Ø	Ø	Ø
MG-Siamese_S	Specific	Image/Mask pair Same domain	Masks	Proxy-based	Mask	Source	Ø
MG-Siamese_ST	Specific	Image/Mask pair Same domain	Masks	Proxy-based	Mask	Source	Target

Table 3: The tested approaches.

Table 4: Accuracy on chairlifts safety problem (experimental context 1).

Method	Accuracy (± std)
Baseline	85.71
Siam_Intra	90.57
Siam_Inter	86.91
Siam_Intra_Proxy	89.35 ± 3.90
Faster R-CNN	87.12
MG-Siamese_S	**94.09**
MG-Siamese_ST	**95.23**

Table 5: Accuracy on cross-domain digit recognition with Rotated MNIST (experimental context 3).

Method	Accuracy
Baseline	88.0
CCSA [21]	89.1
MG-Siamese_S	**89.5**

Table 6: Accuracy on cross-domain digit recognition (experimental context 2).

Source Target	MNIST SVHN	SVHN MNIST	MNIST MNISTM	MNIST USPS	USPS MNIST
Baseline	35.2	77.6	58.0	83.1	68.9
MG-Siamese_S	**42.5**	**81.0**	**64.6**	**87.1**	**78.4**

One crucial point of our approach is to insert shape priors in the network by using binary images. As discussed earlier, one way to introduce priors about important elements in an image is to apply object detection to these elements. Using Faster R-CNN led to an accuracy of (87.12%) which surpasses the Baseline accuracy (85.71%). However, the insertion of shape priors with binary masks using MG-Siamese_ST clearly outperforms these two approaches with an accuracy of (95.23%), see Table 4. In the same table, we notice that the addition of masks surpasses the performance of Siam_Intra (90.57%), showing that these binary images inform the network about the most relevant part of the images for each category.

2. Should we learn domain invariant features or domain-specific features?

One way to reach generalization across domains is to mix all the data from the different domains at training time and group them together in the feature space. This is the solution selected by the Baseline, Siam_inter, CCSA, and Faster R-CNN. The other approaches rather propose learning specific and accurate features for each domain, and using the adapted feature subspace at test time. It seems that mixing the domains forces the network to spend a lot of energy learning invariant features at the expense of the main task. Indeed, we can see that Siam_Inter (86.91%) does not reach the same accuracy as Siam_Intra (90.57%) while being trained with the same data, see Table 4.

3. Is it better to learn proxy-based or pair-based metrics?

There is a current discussion about the pros and cons of pair-based and proxy-based metric learning solutions [22]. Our experiments show that randomly selecting a proxy for each category per domain among the training data is irrelevant. Indeed, we can see in Table 4 that Siam_Intra_Proxy (89.35%) does not improve over Siam_Intra (90.57%), which is a pair-based training solution. Furthermore, the random selection of proxy makes the approach unstable. One solution in the current approaches consists in learning the proxy with a specific network instead of randomly selecting it. Obviously, this additional learning step requires much more training data than in our settings.

4. Should we choose images or masks as a proxy?

On the other hand, when a shape prior is available, it should be used as a proxy for each

category per domain, as done in our MG-Siamese networks. We can see that using the masks as a proxy instead of images boosts the performance from (89.35%) to (95.23%), see Table 4.

Tables 5, and 6, also show that inserting shape priors through Siamese architecture is accurate since it provides the best results in the other experimental contexts too, by concentrating the data from the same category within each domain around a single point our MG-Siamese_S approach outperforms the other solutions *i.e.*, Baseline and CCSA.

4.4. Observing the embedding space

Since our approach consists in projecting the data into an embedding space, we propose to look at their distribution in 2D. In order to observe the distribution of the data in the embedding space, we applied tSNE (t-distributed Stochastic Neighbor Embedding) [36] on the output feature vectors of our MG-Siamese_S model. We also applied tSNE to the feature vectors of the last fully connected layer of the Baseline classifier in order to compare the two distributions.

Fig. 10 shows the embedding space of both the Baseline and MG-Siamese model in the context of the chairlifts safety problem. On the two plots, we can see samples from the source data that have been used to learn the network depicted with dot markers, and samples from the target data (not used during training) depicted with crosses. The colors indicate the class of each point. In these plots, we can clearly see that the binary classifier separates well the two classes when considering the source data, but we can also see that the target data is sometimes not on the correct side of the classifier. On the other hand, we can see that the Siamese architecture is pushing far away the points from different classes thanks to the contrastive loss and that the target points are also mainly following the source distribution. Also, we notice that the open (resp. closed) masks are near to each other and far from the closed (resp. open) masks. We do not use any constraint to enforce this behavior, but since we are learning on a set of different chairlifts and projecting them in the same embedding space, the model automatically learns the features that help to discriminate the open masks from the closed masks.

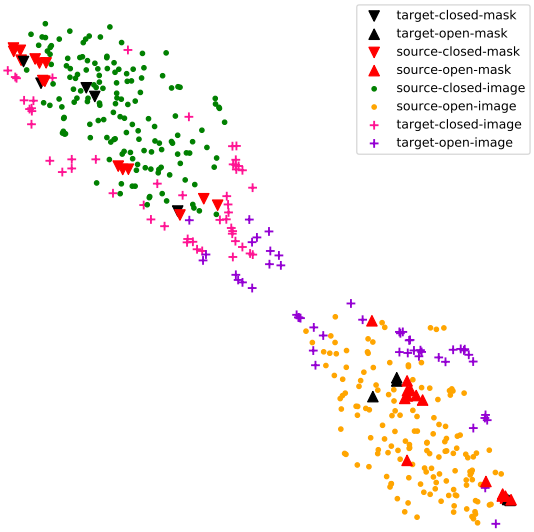
5. Conclusion

In this paper we have addressed a challenging case of image classification where a small set of labeled images is available and generalization over unseen target domains is needed. Our proposition rely on very intuitive ideas implemented in the framework of a specific embedding space with Euclidean properties. Supposing each class comes with a binary mask focusing on relevant elements to detect, the embedding space is first learned over a set of known source domains with a Siamese network uses image/mask pairs as input. A contrastive loss is used to provide Euclidean properties to the space. In the context of video-surveillance of chairlifts, we shown that this approach performs better than a generic image classifier in the target domain. Furthermore, we run extensive tests on many alternatives, measuring the impact of each choice.

Acknowledgements

This work has been supported by the French Public Investment Bank (BPI) as part of MIVAO project - FUI-AAP23, in collaboration with Bluecime company.

(a)



(b)

Figure 10: Source and target images and masks visualized using TSNE in the context of the chairlifts safety problem. (a) Embedding space of the Baseline, (b) Embedding space of the MG-Siamese_S model with the corresponding open and closed masks. It is preferable to use color for this figure in print.

References

[1] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, P. Isola, Rethinking few-shot image classification: a good embedding is all you need?, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16, Springer, 2020, pp. 266–282.

[2] X. Chen, K. He, Exploring simple siamese representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15750–15758.

[3] G. Koch, R. Zemel, R. Salakhutdinov, Siamese neural networks for one-shot image recognition, in: ICML deep learning workshop, Vol. 2, Lille, 2015.

[4] H. Alqasir, D. Muselet, C. Ducottet, Mask-guided image classification with siamese networks, in: International Conference on Computer Vision Theory and Applications, 2020.

[5] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, R. Shah, Signature verification using a "siamese" time delay neural network, International Journal of Pattern Recognition and Artificial Intelligence 7 (04) (1993) 669–688.

[6] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, P. H. Torr, Fully-convolutional siamese networks for object tracking, in: European conference on computer vision, Springer, 2016, pp. 850–865.

[7] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: Closing the gap to human-level performance in face verification, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 1701–1708.

[8] Z. Wu, Y. Xiong, S. X. Yu, D. Lin, Unsupervised feature learning via non-parametric instance discrimination, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3733–3742.

[9] A. v. d. Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, arXiv preprint arXiv:1807.03748.

[10] S. En, A. Lechervy, F. Jurie, Ts-net: combining modality specific and common features for multimodal patch matching, in: 2018 IEEE International Conference on Image Processing (ICIP), Ieee, 2018.

[11] M. Long, Z. Cao, J. Wang, M. I. Jordan, Conditional adversarial domain adaptation, in: Advances in Neural Information Processing Systems, 2018, pp. 1640–1650.

[12] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7167–7176.

[13] C. Liu, C. Xu, Y. Wang, L. Zhang, Y. Fu, An embarrassingly simple baseline to one-shot learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 922–923.

[14] K.-C. Peng, Z. Wu, J. Ernst, Zero-shot deep domain adaptation, in: The European Conference on Computer Vision (ECCV), 2018.

[15] A. Kumagai, T. Iwata, Zero-shot domain adaptation without domain semantic descriptors, ArXiv abs/1807.02927.

[16] S. Shankar, V. Piratla, S. Chakrabarti, S. Chaudhuri, P. Jyothi, S. Sarawagi, Generalizing across domains via cross-gradient training, in: International Conference on Learning Representations, 2018.

[17] Y. Yang, T. Hospedales, A unified perspective on multi-domain and multi-task learning, in: 3rd International Conference on Learning Representations (ICLR), 2015.

[18] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (10) (2009) 1345–1359.

[19] K. Zhou, Y. Yang, T. M. Hospedales, T. Xiang, Deep domain-adversarial image generation for domain generalisation., in: AAAI, 2020, pp. 13025–13032.

[20] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, Domain generalization for object recognition with multi-task autoencoders, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 2551–2559.

[21] S. Motiian, M. Piccirilli, D. A. Adjeroh, G. Doretto, Unified deep supervised domain adaptation and generalization, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 5715–5725.

[22] S. Kim, D. Kim, M. Cho, S. Kwak, Proxy anchor loss for deep metric learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[23] K. Rakelly, E. Shelhamer, T. Darrell, A. A. Efros, S. Levine, Conditional networks for few-shot semantic segmentation, in: International Conference on Learning Representations, 2018.

[24] J. Zhao, C. G. M. Snoek, Dance with flow: Two-in-one stream action detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019.

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

[26] H. Law, J. Deng, Cornernet: Detecting objects as paired keypoints, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018.

[27] S. Qiao, L.-C. Chen, A. Yuille, Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 10213–10224.

[28] Q. Chen, Z. Song, J. Dong, Z. Huang, Y. Hua, S. Yan, Contextualizing object detection and classification, IEEE transactions on pattern analysis and machine intelligence 37 (1) (2014) 13–27.
[29] C. Song, Y. Huang, W. Ouyang, L. Wang, Mask-guided contrastive attention model for person re-identification, in: 2018 IEEE conference on computer vision and pattern recognition (CVPR), Ieee, 2018.
[30] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, F. Moreno-Noguer, Discriminative learning of deep convolutional feature point descriptors, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 118–126.
[31] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: 2006 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06), Vol. 2, IEEE, 2006, pp. 1735–1742.
[32] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, 2015.
[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.
[34] S. Dai, Y. Cheng, Y. Zhang, Z. Gan, J. Liu, L. Carin, Contrastively smoothed class alignment for unsupervised domain adaptation, in: Proceedings of the Asian Conference on Computer Vision, 2020.
[35] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proc. 8th Int'l Conf. Computer Vision, Vol. 2, 2001, pp. 416–423.
[36] L. v. d. Maaten, G. Hinton, Visualizing data using t-sne, Journal of machine learning research 9 (Nov) (2008) 2579–2605.