Multi-component Image Translation for Deep Domain Generalization

Mohammad Mahfujur Rahman Clinton Fookes Mahsa Baktashmotlagh Sridha Sridharan
Image and Video Laboratory, Queensland University of Technology (QUT), Brisbane, QLD, Australia
Email: {m27.rahman, c.fookes, m.baktashmotlagh, s.sridharan}@qut.edu.au

Abstract

Domain adaption (DA) and domain generalization (DG) are two closely related methods which are both concerned with the task of assigning labels to an unlabeled data set. The only dissimilarity between these approaches is that DA can access the target data during the training phase, while the target data is totally unseen during the training phase in DG. The task of DG is challenging as we have no earlier knowledge of the target samples. If DA methods are applied directly to DG by a simple exclusion of the target data from training, poor performance will result for a given task. In this paper, we tackle the domain generalization challenge in two ways. In our first approach, we propose a novel deep domain generalization architecture utilizing synthetic data generated by a Generative Adversarial Network (GAN). The discrepancy between the generated images and synthetic images is minimized using existing domain discrepancy metrics such as maximum mean discrepancy or correlation alignment. In our second approach, we introduce a protocol for applying DA methods to a DG scenario by excluding the target data from the training phase, splitting the source data to training and validation parts, and treating the validation data as target data for DA. We conduct extensive experiments on four cross-domain benchmark datasets. Experimental results signify our proposed model outperforms the current state-of-the-art methods for DG.

1. Introduction

The success of Deep Neural Networks (DNNs) or Deep Learning (DL) largely depends on the availability of large sets of labeled data. When the training and test (source and target) images come from different distributions (domain shift), DL cannot perform well. Domain adaption (DA) and domain generalization (DG) methods have been proposed to address the poor performance due to such domain shift. DA and DG are similar frameworks, but the key difference between DA and DG is the availability of the target data in training phase. DA methods access the target data during training while DG approaches do not have access to the target data.

DA is an attractive area of research in computer vision and pattern recognition fields because it handles the task properly with limited target samples. Domain adaptation can be classified into three groups: Unsupervised Domain Adaptation (UDA), Semi-Supervised Domain Adaptation (SSDA) and Supervised Domain Adaptation (SDA). UDA does not require any labeled target data whereas SDA needs all labeled target samples. Unsupervised deep domain adaptation (UDDA) methods require large sets of target data in order to be more successful in producing a desired output.

To reduce the discrepancy between the source and target data, traditional domain adaptation methods consider seeking domain-invariant information of the data, adapting classifiers, or both. In the conventional domain adaptation task, we still access the target data in training phase. However, in real world scenarios, the target data are out of reach in the training phase. Consequently, ordinary domain adaptation methods do not perform well without having target data in the training phase.

When target data is inaccessible, the issue of DG arises. Without the target data, DG makes full use of and derives benefit from all available source domains. It then tries to acquire a domain independent model by merging data sources which can be used for unknown target data. For an example, we have the images from ImageNet [6] and Caltech-256 [15] datasets and we want to classify images in the LabelMe dataset [35]. We have source data from multiple sources for training without having any knowledge about the target data. That is, the target data is completely unseen during the training phase. DG models utilize the captured information from various source domains and generalize to unseen target data, which is used only in the test phase. Although DA and DG frameworks are very close and both have similar goals (producing a strong classifier on target data), the existing DA techniques do not perform well when directly applied in DG.

In this paper, we propose a novel deep domain generalization framework by utilizing synthetic data that are generated by a GAN during the training stage where the discrep-

ancy between the real and synthetic data is minimized using the existing domain adaptation metrics such as maximum mean discrepancy or correlation alignment. The motivation behind this research is that if we can generate different images from a single image whose styles are similar to the available source domains, the framework will learn a more agnostic model that can be applied for unseen data. Our approach takes advantage of the natural image correspondence built by CycleGAN [45] and ComboGAN [1]. The contributions of this paper are two-fold:

- We implement a novel deep domain generalization framework utilizing synthetic data that is generated by a GAN. We generate images from one source domain into the distributions of other available source domains which are used in the training phase as validation data. The discrepancy between the real data and synthetic data is decreased using existing domain discrepancy metrics in DG settings.

- We introduce a protocol for applying DA methods on DG scenarios where the source datasets are split into training and validation sets, and the validation data acts as target data for DG.

We conduct extensive experiments to evaluate the image classification accuracy of our proposed method across a large set of alternatives in DG settings. Our method performs significantly better than previous state-of-the-art approaches across a range of visual image classification tasks with domain mismatch between the source and target data.

2. Related Research

This section reviews existing research on DA and DG, especially in the avenue of object classification.

2.1. Domain Adaptation

To address the problem of domain shift, many DA methods [13, 23–25, 31, 37, 40–42, 44] have been introduced in recent years. All the DA techniques can be divided into two main categories: Conventional Domain Adaptation methods [13, 31, 37, 44] and Deep Domain Adaptation methods [23–25, 40–42]. The conventional domain adaptation methods have been developed into two phases, the features are extracted in the first phase and these extracted features are used to train the classifier in the second phase. However, the performance of these DA methods is not satisfactory.

Obtaining the features using a deep neural network even without adaptation techniques outperforms conventional DA methods by a large margin. The image classification accuracy obtained with Deep Convolutional Activation Features (DeCAF) [8] even without using any adaptation algorithm is remarkably better than any conventional domain adaptation methods [13, 31, 37, 44] due to the capacity of a DNN to extract more robust features using nonlinear function. Inspired by the success of DL, researchers widely adopted DL based DA techniques.

Maximum Mean Discrepancy (MMD) is a well known metric for comparing the discrepancy between the source and target data. Eric Tzeng et al. [41] introduced the Deep Domain Confusion (DDC) domain adaptation method where the discrepancy is reduced by introducing a confusion layer. In [40], the previous work [41] is improved by adopting a soft label distribution matching loss which is used to reduce the discrepancy between the source and target domains. Long et al. proposed the Domain Adaptation Network (DAN) [23] that introduced the sum of MMDs defined between several layers which are used to mitigate the domain discrepancy problem. This idea was further boosted by the Joint Adaptation Networks [25] and Residual Transfer Networks [24]. Hemanth et al. [42] proposed a new Deep Hasing Network for UDA where hash codes are used to address the DA problem.

Correlation Alignment is another popular metric for feature adaptation where the covariances or second order statistics of the source and target data are aligned. In [26, 38, 39], UDDA approaches have been proposed where the discrepancy between the source and target data is reduced by aligning the second order statistics of the source and target data. The idea of [26, 38, 39] is similar to Deep Domain Confusion (DDC) [41] and Deep Adaptation Network (DAN) [23] except that instead of MMD, they adopted CORAL loss to minimize the discrepancy. CORAL loss is modified in [20] by introducing a Riemannian Metric which performs better than [26, 39]. The aforementioned methods utilized two streams of Convolutional Neural Networks (CNN) where the source and target networks are fused at the classifier level. Domain-Adversarial Neural Networks (DANN) [10] introduced a deep domain adaptation approach by integrating a gradient reversal layer into the traditional architecture.

Current DA approaches still suffer from DG problems when the target data is unavailable during training. In this paper, we explore how the DA approaches can be applied more efficiently on DG scenarios to improve their generalization capability.

2.2. Domain Generalization

In recent research, DG is a less explored issue than DA. The aim of DG is to acquire knowledge from the multiple source domains available to obtain a domain-independent model that can be used for a particular task, such as classification, to an unseen domain. Blanchard et al. [2] first introduced an augmented Support Vector Machine (SVM) based model that solved automatic gating of flow cytometry by encoding empirical marginal distributions into the kernel. In [28], a DG model was proposed which learned

a domain invariant transformation by reducing the discrepancy among available source domains. Xu et al. [43] proposed a domain generalization method that worked on unseen domains by using low-rank structure from various latent source domains. Ghifary et al. [12] proposed an autoencoder based technique to extract domain-invariant information through multi-task learning. These DG methods aggregate all the samples from training datasets to learn a shared invariant representation which is then used for the unseen target domain.

In [19], a domain generalization technique was introduced based on a multi-task max-margin classifier that regulates the weights of the classifier to improve the performance on unseen datasets which was inaccessible in the training phase. Fang et al. [9] proposed a domain generalization method using Unbiased Metric Learning (UML). It makes a less biased distance metric that provides better object classification accuracy for an unknown target dataset. In [43], a domain generalization approach was proposed that combined the score of the classifiers. For multi-view domain generalization, [29, 30] proposed another DG approach where multiple types of features of the source samples were used to learn a robust classifier. All the above methods exploit all the information from the training data to train a classifier or adjust weights.

The above mentioned DG methods use shallow models, hence the performance still needs to improve. Moreover, these shallow models extract the features first, which are then used to train the classifier. As a result, these are not end-to-end methods. Motiian et al. [27] proposed supervised deep domain adaptation and generalization using contrastive loss to align the source data and target data. In this method, the source domains are aggregated and the learned model from source data is used for unseen target data. Li et al. [22] proposed another DG method based on a low-rank parameterized CNN. A deep domain generalization architecture with a structured low-rank constraint to mitigate the domain generalization issue was proposed in [7] where consistent information across multiple related source domains were captured.

Although many DA techniques based on deep architectures are proposed in recent years, very few DG approaches based on deep architecture are introduced. In this paper, we explore GAN along with a deep architecture to address the DG challenges. We propose a deep domain generalization framework using synthetic data generated by a GAN where the styles are transferred from one domain into another domain.

3. Methodology

In this section, we will introduce our proposed Deep Domain Generalization approach by using synthetic data during training to produce an agnostic classifier that can be applied for unseen target data. We build a model to generate images with the distributions of one source domain into other available source domains. We follow the definitions and notation of [5, 11, 32]. Let, X and Y stand for the input (data) and the corresponding output (label) random variables, and $P(X, Y)$ represents the joint probability distribution of the input and output. We can define a domain as the probability distribution $P(X, Y)$ on $X \times Y$. We formalize domain adaptation and domain generalization as follows:

Domain Adaptation

Let us consider that the source domain data samples are $S_D = \{X_i^s\}$ with available labels $L_s = \{Y_i\}$ and the target data samples are $T_D = \{X_i^t\}$ without labels. The data distribution of the source and target data are different, i.e., $P_s(X_i^s, Y_s) \neq P_t(X_i^t, Y_t)$ where Y_t is the label of target samples. The task of domain adaptation is to learn a classifying function $f : X \rightarrow Y$ able to classify $\{X_i^t\}$ to the corresponding $\{Y_i^t\}$ given S_D and T_D during training as the input.

Domain Generalization

Let us consider that we have a set of n number of source domains such as $\Delta = S_D^1; S_D^2; S_D^3; \ldots; S_D^n$ and target domain T_D^n where $T_D^n \notin \Delta$. The aim of domain generalization is to gain a classifying function $f : X \rightarrow Y$ able to classify $\{X_i^t\}$ to the corresponding $\{Y_i^t\}$ given $S_D^1; S_D^2; S_D^3; \ldots; S_D^n$ during training as the input, but T_D^n is unavailable during training.

3.1. Our Approach

Generative Adversarial Networks (GANs) have accomplished great outcomes in different applications, for example, image generation [14], text2image [34], image-to-image translation [17, 45], person re-identification [18] and image editing [33]. The strong success of GANs is influenced by the concept of an adversarial loss that drives the generated images to be identical to the real images. Zhu et al. [46] proposed a conditional multimodal image-to-image translation method where the training procedure requires paired data which is a supervised solution of image translation from one domain to other domains. As we do not have paired data on domain adaptation or generalization settings, we do not consider this method for synthetic image generation. Zhu et al. [45] proposed CycleGAN that learned a mapping between an input image and an output image using both adversarial and cycle consistency loss where unpaired data were used for image generation. Anoosheh et al. [1] modified CycleGAN by reducing the number of generators and discriminators while there are more than two domains. Choi et al. [4] proposed Stargan for image translation from one domain to another domain. Huang et al. [16]

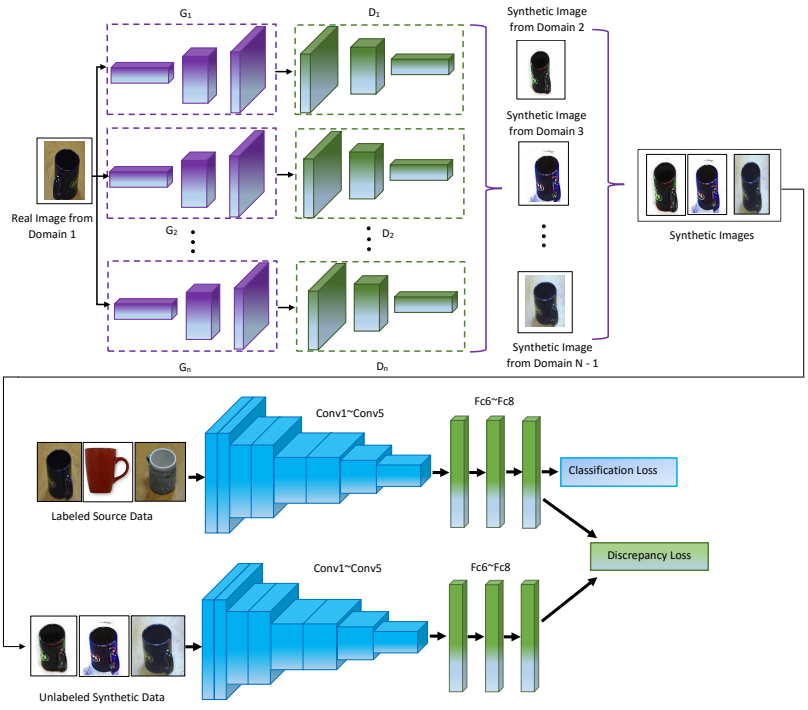

Figure 1: Our proposed methodology of Deep Domain Generalization. At first, synthetic images are generated using ComboGAN [1] which are used during the training phase and a discrepancy loss is used to mitigate the distribution discrepancy between the real source domain images and generated synthetic images. For an example, in Office-Caltech dataset [13], 4 different domains are available: Amazon, Webcam, DSLR and Caltech. When Caltech domain is unseen, we generate the images from the other three domains: Amazon, Webcam and DSLR. These generated images are used to train the network while the images from Caltech domain are classified during the test phase.

proposed another multi-modal image translation technique known as MUNIT which can translate image from one domain to other domains. However, the experiments so far have been tailored by [4, 16] to merely two domains at a time whereas ComboGAN [1] is capable to translate images at a time when more than two domains exist in the dataset. Our approach takes advantage of the natural image correspondence built by [1]. Figure 1 shows the overall architecture of our proposed method for DG based on synthetic image. We use ComboGAN [1] to translate the samples in the source domain D_s^n to those other domains. Having 4 domains of Office-Home dataset [42], we take the image from one domain i.e., Art and translate it to 4 different image domains which are similar to Art, $Clipart$, $Product$ and $Real-world$ domains.

GANs comprise of a generator G and a discriminator D. The input of the generator is random numbers and it generates an image. This generated image is fed into the discriminator with a stream of images taken from real datasets and the discriminator returns the probabilities representing the probability of a real or fake prediction. The training process is called adversarial training. Training itself is executed in two alternating steps; first the discriminator D is trained to distinguish between one or more pairs of real and generated samples, and the generator is trained to fool the discriminator with generated samples. At beginning, the discriminator should be too powerful so that it can identify the real and fake images. After that the generator would be more powerful than the discriminator so that the discriminator cannot distinguish the real and generated images. The training procedure of GAN is a min-max game between a generator and a discriminator. The optimization problem of GAN can be expressed as,

$$\min_G \max_D E_x[logD(x)] + E_z[log(1 - D(G(z))]. \quad (1)$$

Let the available source domains be S_D^1; S_D^2; S_D^3;...;S_D^n. The training samples from the domains are $\{X_i^1\}, \{X_i^2\}, \{X_i^3\}, \ldots \{X_i^n\}$ respectively. The corresponding data distribution of the domains are $P_s^1, P_s^2, P_s^3, \ldots, P_s^n$. The aim of our model is to learn mapping functions among the available source domains S_D^1; S_D^2; S_D^3;...;S_D^n. Our model includes n mappings G_1: S_D^1

$\rightarrow S_D^2$; G_2: $S_D^1 \rightarrow S_D^3$;...; G_n: $S_D^1 \rightarrow S_D^n$. In our model, we have n number of discriminators where the discriminators distinguish between the real images and translated images. For transferring one domain's images into the style of other domain's images, we use adversarial losses [14] and cycle consistency losses [45]. To compare the distribution of the synthetic images to the distribution in the target domain, adversarial losses are used. On the other hand, to preclude the learned mappings G_1, G_2, G_3 and G_n to be in conflict with each other, cycle consistency losses are used. The adversarial losses, for an example, mapping function $G_1 : S_D^1 \rightarrow S_D^2$ and its discriminator D_1 can be expressed as,

$$L_{GAN}(G_1, D_1, S_D^1, S_D^2) = E_{y \sim P_{S2}}[logD_1(y)] + E_{x \sim P_{S1}}[log(1 - D_1(G_1(x))], \quad (2)$$

where the generator attempts to generate images with the distribution of a new domain (S_D^2) and the discriminator D_1 tries to differentiate the generated images from the real images.

We need to map an individual input x_i to an output y_i, however, adversarial losses cannot ensure that it can map the input to the output properly. In addition to decrease the discrepancy among mapping functions, we use cycle consistency loss,

$$L_{cyc}(G_1, G_2) = E_{x \sim P_{s1}}[||G_2(G_1(x)) - x||] + E_{y \sim P_{s2}}[||G_1(G_2(y)) - y||]. \quad (3)$$

To transfer the style of one domain into another domain, for an example, to generate images in the style in domain S_D^2 from the image of domain S_D^1, the full objective of the GAN is,

$$L(G_1, G_2, D_1, D_2) = L_{GAN}(G_1, D_2, S_D^2, S_D^1) + L_{GAN}(G_2, D_1, S_D^2, S_D^1) + \lambda L_{cyc}(G_1, G_2). \quad (4)$$

3.2. Domain Generalization using Synthetic Data

The previous domain generalization method [11] divided the source domain datasets into a training set and a test set by random selection from the source datasets. It minimized the discrepancy between these training and test samples during training. In our method, we decrease the discrepancy between the real images and generated images. If there is no domain mismatch between the source and target data, the classifying function f is trained by minimizing the classification loss,

$$L_C(f) = E[l(f(X^s), Y], \quad (5)$$

where $E[.]$ and l represent mathematical expectation and any loss functions (such as, categorical cross-entropy for multi-class classification) respectively. On the other hand, if the distribution of the source and target data are different, a DA algorithm is used to minimize the discrepancy between the source and target distributions. As in UDA techniques, there is no labeled data, thus the discrepancy is minimized using the following Equation,

$$L_D(g) = d(p(g(X^s)), p(g(X^t))). \quad (6)$$

The motivation behind Equation (6) is to adjust the distributions of the features within the embedding space, mapped from the source and the target data. Overall, in most of the deep domain adaptation algorithms, the objective is,

$$L_T(f) = L_C + L_D. \quad (7)$$

We use GAN to generate images from one source domain into the distributions of other available source domains. The discrepancy between the synthetic and real images is minimized by domain discrepancy loss during training. Thus, the learned model can be applied to an unseen domain more effectively. We propose the new loss for training the model as follows,

$$L_D(g) = d(p(g(X^s)), p(g(X^G))), \quad (8)$$

where X^s are the real images and X^G are the synthetic images, d can be any domain discrepancy metric such as coral loss or maximum mean discrepancy.

3.3. Protocol for Applying DA Methods on DG Scenario

In this section, we will discuss our second approach for domain generalization where we randomly split each source domain data into a training (70%) and a validation set (30%). Most of the DA methods use Siamese based architecture where two stream of CNN is used. This 70% (from all source domain) is fed into one stream of CNN and the other 30% (from all source domain) data is fed into the second stream of CNN during the training phase. The discrepancy is minimized between these data in the training phase. In the test phase, we use unlabeled target data which is completely unseen during the training stage to evaluate our method for domain generalization.

4. Experiments

In this section, we conduct substantial experiments to evaluate the proposed method and compare with state-of-the-art UDDA and DG techniques.

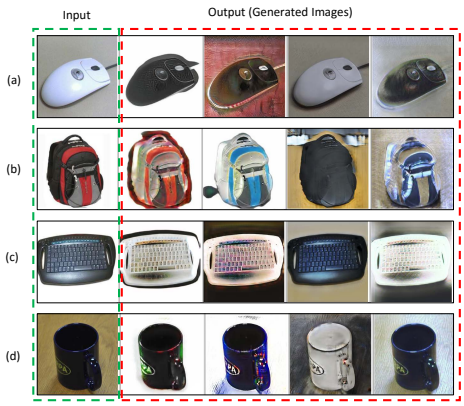

Figure 2: Sample synthetic images from the **Office-Caltech** dataset [13] which comprises 4 domains. (a) Four images are generated from one image of Webcam domain, (b) four images are generated from one image of Amazon domain, (c) four images are generated from one image of Caltech domain and (d) four images are generated from one image of DSLR domain.

4.1. Datasets

We evaluate all the methods on four standard domain adaptation and generalization benchmark datasets: Office-31 [36], Office-Caltech [13], Office-Home [42] and PACS [22].

Office-31 [36] is the most prominent benchmark dataset for domain adaptation. It has 3 domains: **Amazon (A)** domain is formed with the downloaded images from amazon.com, **DSLR (D)**, containing images captured by Digital SLR camera, and **Webcam (W)** contains images that captured by web camera with different photo graphical settings. For all experiments, we use labeled source data and unlabeled target data for unsupervised domain generalization where the target data is totally unseen during training. We conduct experiments on 3 transfer tasks ($A, W \rightarrow D$; $A, D \rightarrow W$ and $D, W \rightarrow A$) where two domains are used as source domains and other is as target domain. For an example, the transfer task of $A, W \rightarrow D$, $Amazon(A)$ and $Webcam(W)$ are used as source domains whereas $DSLR(D)$ is used as target domain. We also calculate the average performance of all transfer tasks.

Office-Caltech [13] dataset is formed by taking the 10 common classes of two datasets: Office-31 and Caltech-256. It has 4 domains: **Amazon (A)**, **Webcam (W)**, **DSLR (D)** and **Caltech (C)**. We conduct experiments on 4 transfer tasks where 3 domains or 2 domains are used as source domains ($W, D, C \rightarrow A$; $A, W, D \rightarrow C$; $A, C \rightarrow D, W$ and $D, W \rightarrow A, C$).

Office-Home [42] is a benchmark dataset for domain adaptation which contains 4 domains where each domain consists of 65 categories. The four domains are: **Art (Ar)**: artistic images in the form of sketches, paintings, ornamentation, etc.; **Clipart (Cl)**: collection of clipart images; **Product (Pr)**: images of objects without a background and **Real-World (Rw)**: images of objects captured with a regular camera. It contains 15,500 images, with an average of around 70 images per class and a maximum of 99 images in a class. We conduct experiments on 4 transfer tasks where 3 domains are used as source domains ($Pr, Rw, Cl \rightarrow Ar$; $Pr, Rw, Ar \rightarrow Cl$; $Pr, Ar, Cl \rightarrow Rw$ and $Ar, Cl, Rw \rightarrow Pr$).

PACS [22] is also a recently released benchmark dataset for domain generalization which is created by considering the common classes among Caltech256 , Sketchy, TU-Berlin and Google Images. It has 4 domains, each domain consists of 7 categories. It contains total 9991 images. We conduct experiments on 4 transfer tasks where 3 domains are used as source domains ($P, C, S \rightarrow A$; $P, S, A \rightarrow C$; $S, A, C \rightarrow P$ and $P, C, A \rightarrow S$).

4.2. Network Architecture

We adopt the network model for generative architecture as [1, 45] where impressive results for image translation in different domains are shown. The GAN network consists of two convolution layers with two stride-2, several residual blocks and two fractionally strided convolution layers with stride 0.5. We use 9 blocks for 256×256 resolution images. We use 70×70 PatchGANs for the discriminator networks. For domain generalization, we use DAN [23], D-CORAL [39], RTN [24] and JAN [25] architecture where Alexnet [21] is used, comprising of five convolution layers and three fully connected layers.

4.3. Experimental Setup

For synthetic image generation, we use ComboGAN [1] where we set the value of λ as 10 in Equation 4. In this model, Adam solver is used with a batch size of 1. We trained all the networks from scratch and we set the learning rate to 0.0002. We set total 200 epochs, and for the first 100 epochs, the learning rate does not change whereas it linearly decays the rate towards zero over the next 100 epochs.

For domain generalization, we use two streams of CNN. In each stream, we extend AlexNet [21] model which is pre-trained on the ImageNet [6] dataset. We set the dimension of the last fully connected layer (fc8) to the number of categories (for instance, 65 for Office-Home dataset). We set the learning rate to 0.0001 to optimize the network. Moreover, we set the batch size to 128, weight decay to 5×10^{-4} and momentum to 0.9 during the training phase.

4.4. Results and Discussion

In this section, we will discuss our experimental results in detail. For fair comparison, we use the same net-

work architecture (AlexNet [21]) that are used in the existing domain adaptation methods such as DAN [23], D-CORAL [39], JAN [25] and RTN [24] in domain generalization settings. DAN [23] is a deep domain adaptation model where the discrepancy between the source and target data is minimized using MMD. JAN [25] is also a deep domain adaptation method where the discrepancy between the source and target data is mitigated using Joint Maximum Mean Discrepancy (JMMD) criterion. RTN [24] is used for minimizing domain discrepancy between different distributions data using residual transfer network and MMD. On the other hand, D-CORAL [39] is a DDA method where the discrepancy between two domains is minimized by the second order statistics (covariances) which is known as correlation alignment.

Sources → Target	**Ours** (DAN)	**Ours** $(D-CORAL)$	**Ours** (JAN)	**Ours** (RTN)
$Pr, Rw, Cl \rightarrow$**Ar**	44.16	44.08	**45.29**	44.58
$Pr, Rw, Ar \rightarrow$**Cl**	39.99	40.28	**40.70**	40.61
$Pr, Ar, Cl \rightarrow$**Rw**	64.17	64.10	**64.89**	64.33
$Ar, Cl, Rw \rightarrow$**Pr**	61.99	62.82	**63.78**	63.05
$Avg.$	52.58	52.82	**53.67**	53.14

Table 1: Recognition accuracies for domain generalization on the Office-Home dataset [42]. Here, we split the source datasets into 70-30% training-validation samples.

Sources → Target	**Ours** (DAN_S)	**Ours** $(D-CORAL_S)$	**Ours** (JAN_S)	**Ours** (RTN_S)
$Pr, Rw, Cl \rightarrow$**Ar**	47.85	47.98	**48.09**	47.90
$Pr, Rw, Ar \rightarrow$**Cl**	43.85	44.73	45.20	**45.30**
$Pr, Ar, Cl \rightarrow$**Rw**	67.27	67.58	**68.35**	68.09
$Ar, Cl, Rw \rightarrow$**Pr**	64.29	64.78	**66.52**	66.21
$Avg.$	55.82	56.27	**57.04**	56.88

Table 2: Recognition accuracies for domain generalization on the Office-Home dataset [42] with synthetic images that are generated using ComboGAN. The subscript **S** represents synthetic data.

Sources → Target	**Ours** (DAN_S)	**Ours** $(D-CORAL_S)$	**Ours** (JAN_S)	**Ours** (RTN_S)
$Pr, Rw, Cl \rightarrow$**Ar**	45.03	45.27	**46.24**	45.71
$Pr, Rw, Ar \rightarrow$**Cl**	41.92	42.60	**42.89**	42.85
$Pr, Ar, Cl \rightarrow$**Rw**	65.39	65.72	**66.08**	65.41
$Ar, Cl, Rw \rightarrow$**Pr**	62.56	63.43	**64.90**	64.37
$Avg.$	53.73	54.26	**54.78**	54.59

Table 3: Recognition accuracies for domain generalization on the Office-Home dataset [42] with synthetic images that are generated using MUNIT. The subscript **S** represents synthetic data.

We evaluate our DG model on different datasets. At first, we follow the same experimental setting as in [11, 12] for Office-Home dataset, and randomly split each source domain into a training set (70%) and a validation set (30%).

Sources → Target	**Ours** (DAN_S)	**Ours** $(D-CORAL_S)$	**Ours** (JAN_S)	**Ours** (RTN_S)
$Pr, Rw, Cl \rightarrow$**Ar**	44.86	45.09	**45.95**	45.26
$Pr, Rw, Ar \rightarrow$**Cl**	40.57	42.19	42.38	**42.41**
$Pr, Ar, Cl \rightarrow$**Rw**	64.80	65.38	**65.94**	65.05
$Ar, Cl, Rw \rightarrow$**Pr**	62.07	63.10	**64.14**	63.51
$Avg.$	53.08	53.69	**54.60**	54.06

Table 4: Recognition accuracies for domain generalization on the Office-Home dataset [42] with synthetic images that are generated using Stargan. The subscript **S** represents synthetic data.

Then we conduct experiments of having one domain totally unseen during the training stage. The 70% data is fed into one stream of CNN and 30% data is fed into the second stream of CNN. In the test phase, we use unlabeled target data which is completely unseen during the training stage. It is noted that the target data is not splitting. In the above setting, we evaluate 4 existing domain adaptation (DAN [23], D-CORAL [39], JAN [25] and RTN [24]) methods for domain generalization settings. We report these comparative results in Table 1.

After that, we generate synthetic images using ComboGAN [1] in the training phase. Figure 2 shows some generated images. The real images are fed into one stream of CNN and synthetic images are fed into another stream of CNN. In Table 2, we report the comparative results. From Table 1 and Table 2, we can observe that our method improves on an average 3% higher than 70%-30% settings in each transfer task on Office-Home dataset.

To further increase the justification of adopting ComboGAN in our framework, we also experimentally evaluate our domain generalization model using synthetic images that are generated by MUNIT [16] and Stargan [4] on Office-Home dataset. The domain generalization on different tasks are reported in Table 3 and Table 4 using MUNIT and Stargan respectively. From the results (Tables 1, 2, 3 and 4), we make two important observations: (1) The multi-component synthetic data generation using GAN boosts the domain generalization performance; and (2) As ComboGAN can handle more than two domains at a time compared to MUNIT and Stargan, it can generate more multi-component images which are more effective for domain generalization.

We further evaluate and compare our proposed approach with both shallow and deep domain generalization state-of-the-art methods: Undoing the Damage of Dataset Bias (Undo-Bias) [19], Unbiased Metric Learning (UML) [9], Low-Rank Structure from Latent Domains for Domain Generalization (LRE-SVM) [43], Multi-Task Autoencoders (MTAE) [12], Domain Separation Network (DSN) [3], Deep Domain Generalization with Structured Low-Rank Constraint(DGLRC) [7], Domain Generalization via Invari-

Sources → Target	Undo-Bias	UML	LRE-SVM	MTAE	DSN	DGLRC	**Ours** (DAN_S)	**Ours** ($D-CORAL_S$)	**Ours** (JAN_S)	**Ours** (RTN_S)
$A, W \rightarrow$**D**	98.45	98.76	98.84	98.97	99.02	**99.44**	97.18	96.18	97.87	97.23
$A, D \rightarrow$**W**	93.38	93.76	93.98	94.21	94.45	**95.28**	92.70	93.20	94.03	94.53
$D, W \rightarrow$**A**	42.43	41.65	44.14	43.67	43.98	45.36	49.30	51.61	**51.73**	50.86
Avg.	78.08	78.05	78.99	78.95	79.15	80.02	79.72	80.33	**81.21**	80.87

Table 5: Recognition accuracies for domain generalization on the Office31 dataset [36] using synthetic images that are generated by ComboGAN. The subscript **S** represents synthetic data.

Sources → Target	Undo-Bias	UML	LRE-SVM	MTAE	DSN	DGLRC	**Ours** (DAN_S)	**Ours** ($D-CORAL_S$)	**Ours** (JAN_S)	**Ours** (RTN_S)
$W, D, C \rightarrow$**A**	90.98	91.02	91.87	93.13	-	**94.21**	92.27	92.79	93.31	93.06
$A, W, D \rightarrow$**C**	85.95	84.59	86.38	86.15	-	**87.63**	84.41	86.74	86.28	85.31
$A, C \rightarrow$**D, W**	80.49	82.29	84.59	85.35	85.76	**86.32**	85.17	82.07	85.17	84.27
$D, W \rightarrow$**A, C**	69.98	79.54	81.17	80.52	81.22	82.24	80.24	79.56	**82.74**	81.53
$Avg.$	81.85	84.36	86.00	86.28	-	**87.60**	85.52	85.29	86.87	86.04

Table 6: Recognition accuracies for domain generalization on the Office-Caltech dataset [13] using synthetic images that are generated by ComboGAN. The subscript **S** represents synthetic data.

Sources → Target	uDICA	LRE-SVM	MTAE	DSN	DBADG	**Ours** (DAN_S)	**Ours** ($D-CORAL_S$)	**Ours** (JAN_S)	**Ours** (RTN_S)
$P, C, S \rightarrow$**A**	64.57	59.74	60.27	61.13	62.86	**64.69**	61.37	62.64	62.64
$P, S, A \rightarrow$**C**	64.54	52.81	58.65	66.54	**66.97**	63.60	66.16	65.98	66.52
$S, A, C \rightarrow$**P**	**91.78**	85.53	91.12	83.25	89.50	90.11	89.16	90.44	89.86
$P, C, A \rightarrow$**S**	51.12	37.89	47.86	58.58	57.51	58.08	**58.92**	58.76	57.68
$Avg.$	68.00	58.99	64.48	63.38	69.21	69.12	68.90	**69.45**	69.18

Table 7: Recognition accuracies for domain generalization on the PACS dataset [22] using synthetic images that are generated by ComboGAN. The subscript **S** represents synthetic data.

ant Feature Representation (uDICA) [28], Deeper, Broader and Artier Domain Generalization (DBADG) [22].

We report comparative results in Table 5, 6 and 7 on Office 31, Office-Caltech, and PACS datasets respectively. From Table 5, it can be seen that, the previous best method [7] achieved 80.02% average accuracy where the domain generalization issue is solved by using a structured low rank constraint. In contrast, our domain generalization method using synthetic data with D-CORAL [39], JAN [25], and RTN [24] network architectures achieves 80.33%, 81.21% and 80.87% average accuracies respectively which outperforms the state-of-the-art methods. For Office-Caltech dataset (see Table 6), although DGLRC [7] achieved best performance, our proposed method is different in network architecture as we use generative adversarial network to generate synthetic data. Using synthetic data with DAN [23], D-CORAL [39], JAN [25] and RTN [24] network architecture, we achieve 85.52%, 85.29%, 86.87% and 86.04% average accuracies respectively. For PACS dataset, our proposed method achieves state-of-the-art performance using synthetic data and JAN architecture [25]. It is worth noting that for PACS dataset, our result is 69.45% while the previous state-of-the-art method achieved 69.21%.

Our image translation model is unsupervised, and capable of translating images from one domain to another in open-set domain adaptation/generalization scenario. However, to minimize discrepancy among different domains, we considered close set domain generalization settings where we assume that every source domains share similar classes as the target domain. In the future work, we will explore our method on outdoor scene dataset where the source domains may not share the same class labels with the target domain.

5. Conclusion

In this paper, we developed a novel deep domain generalization architecture using synthetic images which were generated by a GAN and existing domain discrepancy minimizing metrics which aims to learn a domain agnostic model from the real and synthetic data that can be applied for unseen datasets. More specifically, we built an architecture to transfer the style from one domain image to another domain. After generating synthetic data, we used maximum mean discrepancy and correlation alignment metrics to minimize the discrepancy between the synthetic data and real data. Extensive experimental results on several benchmark datasets demonstrate our proposed method achieves state-of-the-art performance.

Acknowledgement

The research presented in this paper was supported by Australian Research Council (ARC) Discovery Project Grants DP170100632 and DP140100793.

References

[1] A. Anoosheh, E. Agustsson, R. Timofte, and L. V. Gool. Combogan: Unrestrained scalability for image domain translation. *CoRR*, abs/1712.06909, 2017.

[2] G. Blanchard, G. Lee, and C. Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in Neural Information Processing Systems (NIPS)*. 2011.

[3] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *Proceedings of the International Conference on Neural Information Processing Systems (NIPS)*, 2016.

[4] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] G. Csurka. *Domain Adaptation in Computer Vision Applications*. Advances in Computer Vision and Pattern Recognition. Springer, 2017.

[6] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR)*, 2009.

[7] Z. Ding and Y. Fu. Deep domain generalization with structured low-rank constraint. *IEEE Transactions on Image Processing*, 27(1):304–313, 2018.

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

[9] C. Fang, Y. Xu, and D. N. Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[10] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[11] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(7):1414–1430, 2017.

[12] M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *IEEE International Conference on Computer Vision, (ICCV)*, 2015.

[13] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*. 2014.

[15] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007.

[16] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. In *European Conference on Computer Vision (ECCV)*, 2018.

[17] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[18] A. Khatun, S. Denman, S. Sridharan, and C. Fookes. A deep four-stream siamese convolutional neural network with joint verification and identification loss for person re-detection. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[19] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[20] P. Koniusz, Y. Tas, and F. Porikli. Domain adaptation by mixture of alignments of second-or higher-order scatter tensors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems (NIPS)*. 2012.

[22] D. Li, Y. Yang, Y. Z. Song, and T. M. Hospedales. Deeper, broader and artier domain generalization. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[23] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning (ICML)*, 2015.

[24] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Conference on Neural Information Processing Systems (NIPS)*, 2016.

[25] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning (ICML)*, 2017.

[26] P. Morerio and V. Murino. Correlation alignment by riemannian metric for domain adaptation. *CoRR*, abs/1705.08180, 2017.

[27] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. Unified deep supervised domain adaptation and generalization. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[28] K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *Proceedings of the International Conference on International Conference on Machine Learning (ICML)*, 2013.

[29] L. Niu, W. Li, and D. Xu. Multi-view domain generalization for visual recognition. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[30] L. Niu, W. Li, D. Xu, and J. Cai. An exemplar-based multi-view domain generalization framework for visual recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 29(2):259–272, 2018.

[31] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22, 2011.

[32] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE Signal Processing Magazine*, 32(3):53–69, 2015.

[33] G. Perarnau, J. van de Weijer, B. Raducanu, and J. M. Álvarez. Invertible Conditional GANs for image editing. In *Neural Information Processing Systems (NIPS) Workshop on Adversarial Training*, 2016.

[34] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning (ICML)*, 2016.

[35] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 2008.

[36] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*, 2010.

[37] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems (NIPS)*. 2008.

[38] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2016.

[39] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision (ECCV) Workshops*, 2016.

[40] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *International Conference on Computer Vision (ICCV)*, 2015.

[41] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014.

[42] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[43] Z. Xu, W. Li, L. Niu, and D. Xu. Exploiting low-rank structure from latent domains for domain generalization. In *European Conference on Computer Vision (ECCV)*, 2014.

[44] B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *International Conference on Machine Learning (ICML)*, 2004.

[45] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[46] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems (NIPS)*. 2017.