Geometric and Textural Augmentation for Domain Gap Reduction

Xiao-Chang Liu Yong-Liang Yang Peter Hall
University of Bath
{XL2546,yy753,maspmh}@bath.ac.uk

Abstract

Research has shown that convolutional neural networks for object recognition are vulnerable to changes in depiction because learning is biased towards the low-level statistics of texture patches. Recent works concentrate on improving robustness by applying style transfer to training examples to mitigate against over-fitting to one depiction style. These new approaches improve performance, but they ignore the geometric variations in object shape that real art exhibits: artists deform and warp objects for artistic effect. Motivated by this observation, we propose a method to reduce bias by jointly increasing the texture and geometry diversities of the training data. In effect, we extend the visual object class to include examples with shape changes that artists use. Specifically, we learn the distribution of warps that cover each given object class. Together with augmenting textures based on a broad distribution of styles, we show by experiments that our method improves performance on several cross-domain benchmarks.

1. Introduction

Recognising objects is a well-researched area of visual computing, with performance rates exceeding 90%. More exactly, state-of-the-art performance is attained for photographic style inputs, performance falls significantly when artwork is input. Consequently, object recognition regardless of depiction remains a significant open problem. Not only is this an interesting in-principle problem, but is one which if solved would support many applications that currently are out of reach, indexing digital collections in art galleries is an obvious example. Less obvious applications we have come across include curating large multi-media databases, building interactive interfaces for professional artists, and IP protection.

The so-called "cross depiction" problem is that *all* classifiers (trained on photographs) show a fall in performance when given art work – this has been observed for some time [24]. One explanation is that most learning algorithms assume that the training (*e.g.*, photographs) and test sets (*e.g.*, art work) have very different low-level statistics. Furthermore, various types of built-in bias (*e.g.*, selection bias, capture bias, and negative set bias) are ubiquitous in existing datasets [34,71]. Several studies [7,39,82] have investigated the similarities and differences between neural networks and human perception. Researchers [19, 62] found the human visual system generalizes robustly across depictions, whereas neural networks are vulnerable to depiction shifts.

Some research considers the problem as one of domain generalization (DG), with each depiction style being a different domain [23,41,52]. However, it has been established that the distance between the two images in the same object class but different styles tends to be larger than two images in the same style but different objects [24] – sufficiently large to frustrate many general DG approaches. Additionally, photographs and artworks of objects do not appear in equal abundance. As a result, moving from training data comprising almost exclusively of photographs to test sets containing artwork is a significant challenge.

In response to such problems, recent literature has made use of style transfer to widen visual object classes so that different depiction domains are included [18]. The underlying idea is that because the input texture varies, the network is forced to learn object-class models that are not biased towards the low-level statistics of any texture class (*e.g.*, photo, line drawing, *etc.*), rather they should rely more on characteristics such as shape [27,46].

Using style transfer as a way to avoid depiction bias has met with some success – performance levels are raised [18,46,55]. However, texture is not the only aspect of object appearance that artists change: artists also warp and deform the objects they render. This so-called "geometric style" has started to influence the style transfer literature [37,49], and has been shown to have a significant impact on subjective judgments regarding style similarity [49]. In other words, people notice that the geometry of objects in artwork differs from the geometry of objects in photographs.

Contributions: First, we bridge the depiction-domain gap in terms of both geometric and texture style, rather than texture alone. The underlying idea is inspired from the literature (see Related Work), which is to robustify classification by including random samples into the training data, a process called "augmentation". **Second**, our augmentation process differs from state of the art. Current literature augments by processing photographs into artwork using a set of (textural) style exemplars. In

contrast, we build independent distributions of texture and geometry descriptors and sample from them to augment the training data. Our experiments show that the geometric and texture augmentation improve classification generalization across several common cross-domain benchmarks. Our code is available at: https://github.com/xch-liu/geom-tex-dg.

2. Related Work

The problem of object recognition regardless of depiction is considered as one of domain generalization (DG), with each style being a different domain. DG considers how to take knowledge from a group of related domains, and apply it to previously unknown domains. The concept was first introduced by Blanchard *et al*. [5] and later popularized by Muandet *et al*. [52]. DG methods can be classified based on the motivations behind them. In the following, we discuss the most relevant work following the taxonomies proposed in [75, 87]. Interested readers can refer to comprehensive reviews [75, 87] for DG on other tasks (*e.g*., segmentation [12, 22, 38, 83], person reID [33, 70], and face attack detection [32, 66]).

2.1. Learning-based DG

***Representation-Learning-based DG*:** Methods in this category are characterized by learning domain-invariant representations by minimizing the difference between source domains. Some works [21, 44] align the marginal distributions of source domains. Motiian *et al*. [51] proposed to reduce distribution mismatch by minimizing the cross-domain contrastive loss. Moreover, some work learns domain-invariant representations through feature distribution alignment [58, 64], feature normalization [54, 57], or attribute regularization [15].

***Learning-Strategy-based DG*:** Methods in this category are characterized by exploiting the general learning strategy to enhance the generalization capability. Some works [3, 13, 42, 45] adopt meta-learning to gain general knowledge by constructing meta-learning tasks to simulate domain shift. Huang *et al*. [30] proposed a self-challenging training strategy that aims to learn general representations by masking out over-dominant features with large gradients. Besides, other learning strategies such as self-supervised learning [8], random forest [63], episodic training [43], and flat minima seeking [9] can also be used for DG.

2.2. Data Augmentation-based DG

Data augmentation [68] is one of the most effective strategies to avoid overfitting in deep learning models. This type of DG method focuses on exploring techniques to enhance the size and quality of training datasets, such that domain-shift-robust models can be built. Our method also belongs to this category.

The basic idea of this strategy is to augment the original data x with new $A(x)$ where $A(\cdot)$ indicates a transformation and is used to simulate domain shift. Hence the design of $A(\cdot)$ is critical to performance. Wong *et al*. [79] found that it is better to perform data augmentation in data-space, if label preserving transformations are known. Liu *et al*. [48] leveraged disentangled domain information to perform cross-domain image translation and manipulation. Shankar *et al*. [65] leveraged domain-adversarial gradients to synthesize domain-agnostic images. Qiao *et al*. [60] exploited task-adversarial gradients to perturb the input images. Xu *et al*. [81] used randomly initialized convolutional network to transform the input images. On the other hand, there are many mix-based methods such as Mixup [85], Manifold-Mixup [73], CutMix [84], Mixmatch [4], PuzzleMix [35], AugMix [26], and StyleMix [28].

***Use of Neural Style Transfer*:** Neural style transfer (NST) [17] is best known for its artistic applications, but it also serves as a tool to improve the generalization ability of simulated datasets, via augmentation, as it can change the color and texture distributions of an image whilst preserving the content. Currently NST methods used in domain generalization researches are: AdaIN [29] used in [6, 11, 53, 55, 69, 88, 92], Style-Swap [10] used in [36], ComboGAN [1] used in [61], and style-complement modules used in [31, 77].

Style transfer methods are also used in our approach. But our use differs from most methods that interpolate the feature statistics between different samples in a mini-batch. We are inspired by a recent study [47] on the impact of style transfer on model robustness which attributed the effectiveness of texture augmentation to the diversity of images. Unlike others in this literature category, we construct both geometric and texture style distributions, which we use to augment training data and so robustify classification. Details are provided in Section 4.

***Summary*:** The problem of recognition in art as well as photographs is a stubborn one, but "robustification" via style transfer has to date offered the greatest uplift in performance. This is possible because the distance between style domains for any single class of object tends to be larger than the distance between classes in a given style [24]: DG methods must shift the domain a long way, style transfer has proven useful in that regard. But, only recently has the style transfer literature considered geometric style [37, 49] – work we take advantage of.

3. Idea Overview

Art historians recognized that style can be decomposed into two parts called "denotation" and "projection" [78]. Within style transfer, these terms are approximated by "texture style" and "geometric style", which are broadly explained below. Understanding the nature of each of them is fundamental to the design of our augmentation strategy. We use the term depiction style to refer to the combination of texture style and geometric style.

***Texture Style*:** There is a considerable texture difference between different image domains (*e.g*., photo vs. painting), which is caused by a range of factors including but not limited to: the medium used, the way it was applied, the substrate it was applied to. Such factors are included within Willet's "denotation" component of style [78]. Within NST these factors are approximated via texture.

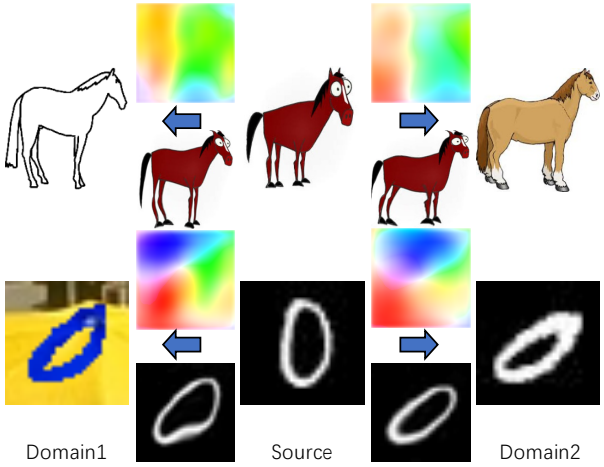

Figure 1. Style comprises both textural and geometric components. Texture style is independent of object class: knowing a picture is a water-color or clip-art is not predictive of its object class. Geometric style, though, is contingent on object class. We warp source images that show the same object class but differ in texture style using [49] (color-coded warp fields above the arrow and the corresponding deformed results below). It can be seen that warp fields within object classes are similar, but vary significantly between object classes.

We assume that any object can be successfully depicted in any denotation, *i.e*., texture style and object class are independent. This assumption is made (implicitly) by NST algorithms and hence by methods that use NST for augmentation.

***Geometric Style*:** Artistic geometric variations extend the possible range of shapes in an object class beyond the boundaries observed in natural images. As an example, Dali painted melting watches that could never be photographed. Artists routinely change the relative shape and size of an object's parts for artistic effect, caricature provides the most obvious examples – but all artists practice geometric deformation of some kind. Unlike texture style, the geometric style depends on object class as illustrated by the following examples.

To make people look stronger, painters increase body mass compared to the head and limb size. The inverse of this gives the person relatively larger hands, feet, and especially eyes, cues that make them look baby-like and cute. Facial caricatures tend to enlarge some features and diminish, again the warp fields form a related set. Figure 1 provides evidence that these observations extend to objects more generally – that there are similarities between warp fields within a given object class, but significant differences in the warp fields between object classes.

4. Method

Augmentation expands training data by processing a training input x to make a new training input $A(x)$. We divide our augmentation pipeline into two steps. Geometric augmentation is first applied, which warps the image; then texture augmentation is applied in which the warped image is re-textured.

These two steps are performed by independently sampling two distributions: one for geometric style and one for texture style. The distributions were constructed by leveraging the pre-trained feature extractors, as shown in Figure 2.

To better describe the method, we introduce the notations used. Let $I^{j,k}$ be an image from depiction style domain j with object class label k, a training image set S which consists of N images from M depiction styles with C object classes can be denoted as:

$$S=\{I_i^{j,k}\in\mathbb{R}^{W\times H\times 3}|\exists!j\in\mathbb{Z}^+_{\leqslant M},\exists!k\in\mathbb{Z}^+_{\leqslant C}\}_{i=1}^{N}; \quad (1)$$

A simplified S of 9 images from 3 depictions with 3 object classes is shown in the left of Figure 2.

Our goal is to use S to learn a predictive model $f:T\mapsto\mathbb{Z}^+_{\leqslant C}$ that generalizes well to an unseen style domain T. That is, use S to build a classifier that performs well when given images in an unseen style.

4.1. Augmenting Geometric Style

Geometric augmentation is applied by randomly deforming training images. There are several requirements for every random deformation: (1) The deformation should be fast enough to be performed online during training; (2) The degree of deformation should be constrained to avoid over-distortions; (3) Most importantly, the type of deformation should be abundant enough to ensure variability, and reasonable enough to avoid meaningless and misleading distortions – geometric augmentation is class-specific. Some widely used geometric augmentations (*e.g*., rotation, translation, flipping, *etc*.) may be useful for size, orientation, *etc*. but do not fully satisfy the class-specific augmentations that we address.

We accomplish the above by constructing object-class specific geometric distributions utilizing the training data. We follow WarpST [49], a general method for estimating intra-class and cross-domain warp fields that leverages feature correlations encoded by pre-trained CNNs. It is the fastest algorithm we know of that warps an image in one style to an image in another style (so satisfies constraint (1), above). The geometric warping module $\mathcal{D}$ accepts an image pair $\langle I_c, I_g\rangle \in \mathbb{R}^{W\times H\times 3}$ and computes a non-parametric vector field $w\in\mathbb{R}^{W\times H\times 2}$:

$$w=\mathcal{D}(I_c,I_g), \quad (2)$$

used to warp I_c onto I_g. It should be noted that we only leverage the warping unit of [49] in the following process hence texture transfer is not included in our geometric augmentation.

WarpST [49] imposes no constraint on the input images. We impose a constraint: both inputs come from the same object class. They can be in any texture style, provided they depict the same kind of thing. Specifically, as shown in the middle of Figure 2, given the training image set S (Eq. 1), we first build new sets P_k of image pairs that disregard the depiction style label but is specific to the object class, k:

$$P_k=\{\langle I_i^{*,k},I_{i'}^{*,k}\rangle|i\neq i'\in\mathbb{Z}^+_{\leqslant N}\}, \quad (3)$$

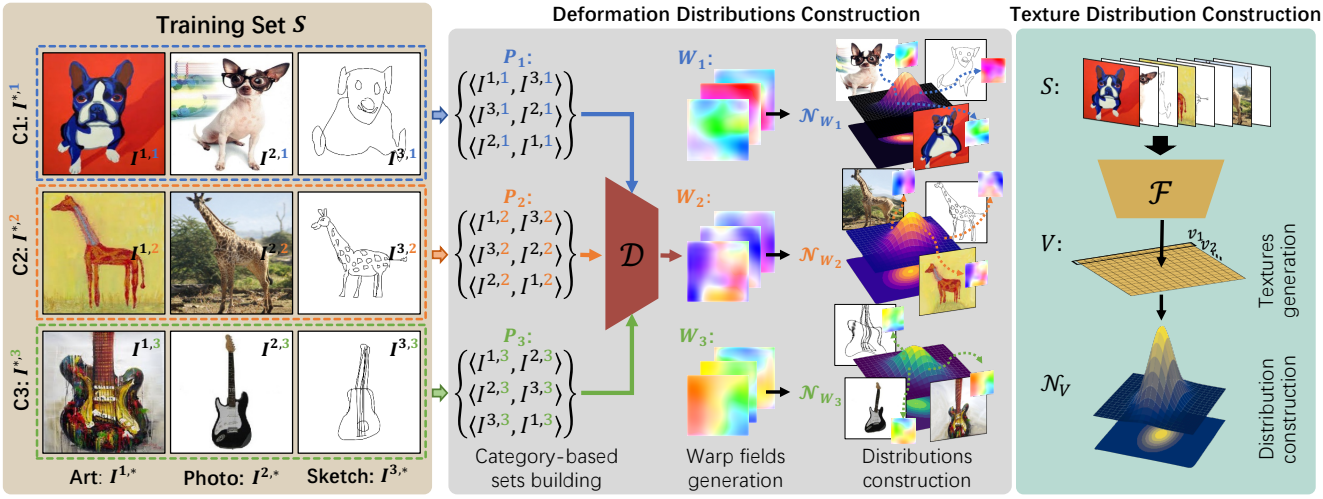

Figure 2. Diagram of the proposed framework for the construction of style distributions. We leverage the training data to construct geometric and texture distributions for data augmentation during the training. **Left:** A simplified training set S which includes three objects (C1-3: dog, giraffe, guitar) from three depictions (art, photo, and sketch). **Middle:** Construction of the geometric distributions. Object-based image pair sets P_* are built and then fed into a geometric warping module $\mathcal{D}$ to generate the warp field sets W_*. Deformation distributions $\mathcal{N}_{W_*}$ are constructed based on W_* per category, and further sampled to achieve the corresponding augmentation effects. **Right:** Construction of the texture distribution. Training set S is fed into a texture style prediction network $\mathcal{F}$ to generate the texture representation set V. Texture distribution $\mathcal{N}_V$ is constructed based on V. For better presentation, warp fields are color-coded, $\mathcal{N}_{W_*}$ and $\mathcal{N}_V$ are visualized in low-dimensional spaces.

and $|P_k| = \binom{N^k}{2}$ where N^k represents the number of images in S with object label k.

Next, we feed the P_k into the warping module $\mathcal{D}$ (Eq. 2) and get class-specific warp fields

$$W_k = \mathcal{D}(P_k) = \{\mathcal{D}(I_i^{*,k}, I_{i'}^{*,k}) | i \neq i' \in \mathbb{Z}^+_{\leqslant N}\}. \tag{4}$$

And W_k can be simply denoted as:

$$W_k = \{w_i^k\}_{i=1}^{|P_k|}, \tag{5}$$

where $w_i^k \in \mathbb{R}^{W \times H \times 2}$ is the i^{th} warp field in W_k. This guarantees that every warp field obeys constraints (2), above. Because images are paired regardless of style labels (Eq. 3), warp fields for both intra-domain and cross-domain are taken into account, meaning the variety and generalization are also ensured: constraint (3) is satisfied.

A straightforward way for geometric augmentation is sampling an element from W_k, then use it to deform the training image. However, this means the diversity of geometric style is limited by the cardinality of W_k. To enable our method to support as broad a range of geometric styles as possible, we construct a geometric distribution based on W_k and sample new warp fields directly from it. That is, we generate a warp field from the class distribution.

We model the deformation distribution with a multivariate normal distribution: $w \sim \mathcal{N}_{W_k}(\mu_k, \Sigma_k)$, where μ_k and Σ_k are the mean vector and covariance matrix of W_k. These statistics are computed by representing W_k as a 2-D matrix in which each column is a "vectorised" warp field, then:

$$\mu_k = \mathbb{E}[W_k], \tag{6}$$

$$\Sigma_k = \mathbb{E}[(W_k - \mu_k)^\intercal (W_k - \mu_k)]. \tag{7}$$

As shown in Figure 2, the constructed distributions can well describe the intra-class warping fields.

The sampling warp fields and the corresponding deformed results are shown in Figure 2. Such a strategy can speed up the processing and significantly increase the variety of geometric styles. In practice, to reduce the amount of computation in Equations 6-7, we downsample W_k before reshaping. Please see Section 5.1 for more details.

4.2. Augmenting Texture Style

A common strategy for texture augmentation is applying style transfer on pairs of images. Currently, there is a great variety of choice of style transfer methods, such as AdaIN [29] used in [55, 88, 92], NST [17] in STADA [86], CycleGAN [93] in [83], *etc*. Most of them follow the same paradigm: sample an image from the training data or a mini-batch as the style exemplar, and use it to transfer the texture of another one. Such sampling process means that the diversity of texture representations are limited to the cardinality of the image set.

A recent study [47] on the impact of style transfer on model robustness attributed the effectiveness of texture augmentation to the diversity of images. Motivated by this observation, we adopt a similar strategy as described in Section 4.1 to amplify the diversity of texture style representations from limited images to the utmost extent. To balance flexibility with speed, similar to [31], we choose a model-based method [20] rather than optimization-based methods [17, 93] as the texture style representation extrac-

tor. The approach of Ghiasi *et al.* [20] predicts an embedding vector $v \in \mathbb{R}^{100}$ from a style image I_s through a texture style prediction network $\mathcal{F}$ which was trained on the PBN dataset[1]:

$$v = \mathcal{F}(I_s), \tag{8}$$

and transforms the texture style of content image I_c through a style transfer network $\mathcal{T}$ leveraging conditional instance normalization [14]:

$$I_o = \mathcal{T}(I_c, v). \tag{9}$$

We leverage the texture style prediction network $\mathcal{F}$ in our method with post-processing. Specifically, as shown in the right of Figure 2, given a training image set S, we first feed S into $\mathcal{F}$ (Eq. 8) and get a set V which consists of texture style vectors:

$$V = \{v_i | v_i \in \mathbb{R}^{100}\}_{i=1}^{N}, \tag{10}$$

where V can be regarded as a 2D matrix comprising texture-style vectors. It should be noted that because texture style and object class as assumed to be independent, we fuse both category and domain labels to treat S as a whole (compare with category-specific warp distributions used for geometric augmentation Section 4.1).

Then we construct the texture style distribution based on V as a multivariate normal distribution with mean μ and covaraiance Σ given by:

$$\mu = \mathbb{E}[V], \tag{11}$$

$$\Sigma = \mathbb{E}[(V-\mu)^\intercal (V-\mu)]. \tag{12}$$

In the training stage, for every training image I, we apply texture augmentation by randomly sampling a texture style embedding v_r from $\mathcal{N}_V$, and transferring the texture style of I based on v_r through the style transfer network $\mathcal{T}$ (Eq. 9). To control the strength of the style augmentation, we linearly interpolate between v_r and the identity transformation achieved by feeding I into the style prediction network $\mathcal{F}$:

$$v_s = \alpha v_r + (1-\alpha)\mathcal{F}(I), \tag{13}$$

where α is the balancing weight to control the extent of augmentation.

In addition to boosting the texture style representations, an additional benefit of the above method is computational efficiency. Batch process images to construct the texture distribution followed by direct representation sampling can largely reduce time cost during the training.

[1]https://www.kaggle.com/c/painter-by-numbers

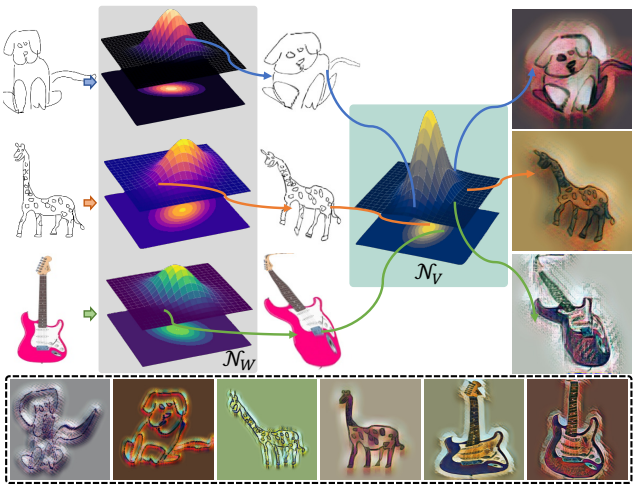

Figure 3. Usage of geometric and texture distributions in the training process. During training, every image is possibly deformed by a warp field sampled from the class-specific warp distribution, then possibly re-textured using a sample from the class-irrelevant texture distribution. Some potential augmentations with mixed geometric and texture styles are shown in the dashed box below.

4.3. Using Deformation and Texture Distributions

During training, deformation distributions ($\mathcal{N}_W$) and texture distributions ($\mathcal{N}_V$) are employed consecutively to augment training images. As shown in Figure 3, training images are first deformed according to their object class labels using the warping fields randomly sampled from the geometric distribution $\mathcal{N}_W$. Then the deformed images are stylised using randomly sampled style representations from the texture distribution $\mathcal{N}_V$.

5. Experiments

In this section, we evaluate the performance of our approach on several benchmarks, and compare it with recent state-of-the-art methods. We also carry out ablation studies to verify the significance of each component in our method for recognition regardless of depiction style. In every case, the hypothesis under test is that augmentation robustifies object classification performance by widening the visual object class (VOC) to include unseen examples. Therefore we apply no augmentation of any kind to any *test* image: our hypothesis is the VOC is wide enough to include the new image.

5.1. Datasets and Implementation Details

We adopt three commonly used multi-domain (depiction style) datasets: **PACS** [41] consists of four domains (Art Painting, Cartoon, Photo, and Sketch) with a total of 9,991 images of 7 classes. **Office-Home** [72] consists of four domains (Art, Clipart, Product, and Real World) with 15,500 images of 65 classes for home and office object recognition. **Digits-DG** [89] contains four digit datasets (MNIST [40],

MNIST-M [16], SVHN [56], and SYN [16]). The digit images in each dataset vary significantly in font style and background.

Our framework can be prepositioned in any CNN architecture. For PACS and Office-Home, we use ResNet [25] as our backbone. All images are resized to 224^2. We train the network using SGD with batch size 64 and a consistent learning rate of 0.001 for 50 epochs. For Digits-DG, we use the same backbone as [89]. Images are resized to 64^2. We train the network using SGD with batch size 128 and a consistent learning rate of 0.05 for 50 epochs. During the construction of geometric distributions (Section 4.1), warp fields w_i (Eq. 5) are implemented based on elastic deformations and resized to improve the later computational efficiency. During the texture augmentation, the balancing weight α (Eq. 13) is set to 0.5. Our framework only applies to the training. For all experiments, we use two probabilities to separately decide if geometric and texture augmentations are performed for any given training image (we further discuss this in Section 5.3). As stated above, at test time no augmentations are applied. We follow the training and evaluation protocol as used in [80, 91, 92]. All the results are reported in terms of accuracy and each performance is an average of three runs.

5.2. Comparison to Baseline and State-of-the-Art

We choose several methods from each of the literature categories described in Section 2 as baseline methods: *representation-learning-based* methods: CCSA [51] and MMD-AAE [44]; *learning-strategy-based* methods: MetaReg [3], JiGen [8], and Epi-FCR [43]; and *data-augmentation-based* methods: ADA [74], CrossGrad [65], L2A-OT [90], and DDAIG [89]. We also compare with recently proposed state-of-the-art algorithms: FACT [80], SagNet [55], and MixStyle [92]. Same as our method, the three state-of-the-art can be categorized into *data-augmentation-based* methods, and texture style transfer methods are also used in [55, 92].

As the learning of our framework does not require domain labels, we evaluate the performance of our approach on both multi-source and single-source domain generalization tasks. Multi-source approaches attempt to generalize to an unseen style given a training set that spans multiple styles; single-source differs in using a single training style.

5.2.1 Multi-Source Domain Generalization

We first test our approach on the multiple source domains generation tasks. For every dataset listed in Section 5.1, we train models using any three domains and evaluate their performances on the remaining one domain.

Results on PACS: The results on PACS are shown in Table 1. It can be seen that our method improves the performance over baselines and achieves results that outperform recent state-of-the-art methods. This implies that adding geometric and texture augmentations to the training procedure improves the generalization ability of models. Moreover, experiments without geometric augmentation (Ours^{-G}) and texture augmentation (Ours^{-T}) verify the effectiveness of each component.

Table 1. Multi-source domain accuracy on PACS. Each column indicates the target domain. Our approach improves the performance over baselines and achieves results that outperform recent state-of-the-art methods. Ours^{-T-G} refers to training without augmentation at all. Ours^{-T} refers to training without textural augmentation. Ours^{-G} refers to training without geometric augmentation.

Methods	Art	Cartoon	Photo	Sketch	Avg.
ResNet-18					
MetaReg [3]	83.70	77.20	95.50	70.30	81.70
JiGen [8]	79.42	75.25	96.03	71.35	80.51
Epi-FCR [43]	82.10	77.00	93.90	73.00	81.50
MMLD [50]	81.28	77.16	96.09	72.29	81.83
DDAIG [89]	84.20	78.10	95.30	74.70	83.10
CSD [59]	78.90	75.80	94.10	76.70	81.40
InfoDrop [67]	80.27	76.54	96.11	76.38	82.33
L2A-OT [90]	83.30	78.20	96.20	73.60	82.80
EISNet [76]	81.89	76.44	95.93	74.33	82.15
SagNet [55]	83.58	77.66	95.47	76.30	83.25
MixStyle [92]	84.10	78.80	96.10	75.90	83.70
FACT [80]	85.37	78.38	95.15	79.15	84.51
Ours^{-T-G}	77.36	76.04	96.08	68.56	79.51
Ours^{-T}	82.74	78.22	95.96	75.26	83.05
Ours^{-G}	86.34	80.12	96.38	81.42	86.07
Ours	**86.99**	**80.38**	**96.68**	**82.18**	**86.56**
ResNet-50					
MetaReg [3]	87.20	79.20	97.60	70.30	83.60
MASF [12]	82.89	80.49	95.01	72.29	82.67
EISNet [76]	86.64	81.53	97.11	78.07	85.84
FACT [80]	89.63	81.77	96.75	84.46	88.15
Ours^{-T-G}	83.04	74.96	97.67	72.39	82.02
Ours^{-T}	87.50	80.76	98.00	74.94	85.30
Ours^{-G}	89.24	83.49	97.79	81.91	88.11
Ours	**89.98**	**83.84**	**98.10**	**84.75**	**89.17**

Results on Office-Home: We show the performance of different algorithms on Office-Home in Table 2. On the whole, our method achieves results that are comparable with state-of-the-art methods. But the effects of deformation and texture augmentation are not as evident as that on PACS, especially the effect of geometric augmentation (Ours^{-T}). This is largely due to the category differences between these two datasets. The animal objects (*e.g.*, giraffe, dog, *etc.*) in PACS exhibit wide intra-class shape and pose variations, whereas the object categories in Office-Home (*e.g.*, bottle, pen, *etc.*) tend to vary far less in terms of geometry. On the other hand, Office-Home itself offers a different challenge to PACS due to its larger number of categories (65) and the average number of images per category (238) are not so balanced as those of PACS (9,991 images of 7 categories). This challenge difference is also reflected in the performance

Table 2. Multi-source domain accuracy on OfficeHome. Each column indicates the target domain.

Methods	Art	Clipart	Product	Real	Avg.
CCSA [51]	59.90	49.90	74.10	75.70	64.90
MMD-AAE [44]	56.50	47.30	72.10	74.80	62.70
CrossGrad [65]	58.40	49.40	73.90	75.80	64.40
DDAIG [89]	59.20	52.30	74.60	76.00	65.50
L2A-OT [90]	**60.60**	50.10	**74.80**	**77.00**	65.60
SagNet [55]	60.20	45.38	70.42	73.38	62.34
MixStyle [92]	58.70	53.40	74.20	75.90	65.50
FACT [80]	60.34	54.85	74.48	76.55	66.56
Ours^{-T-G}	57.87	48.89	73.96	75.31	64.01
Ours^{-T}	58.43	53.10	73.52	75.42	65.12
Ours^{-G}	60.28	55.55	74.67	75.90	66.60
Ours	60.40	**56.30**	74.55	75.77	**66.75**

Table 3. Multi-source domain accuracy on Digits-DG. Each column indicates the target domain.

Methods	MNIST	MNIST-M	SVHN	SYN	Avg.
CCSA [51]	95.2	58.2	65.5	79.1	74.5
MMD-AAE [44]	96.5	58.4	65.0	78.4	74.6
CrossGrad [65]	96.7	61.1	65.3	80.2	75.8
DDAIG [89]	96.6	64.1	68.6	81.0	77.6
Jigen [8]	96.5	61.4	63.7	74.0	73.9
L2A-OT [90]	96.7	63.9	68.6	83.2	78.1
MixStyle [92]	96.5	63.5	64.7	81.2	76.5
FACT [80]	**97.9**	65.6	72.4	**90.3**	81.5
Ours^{-T-G}	96.1	60.9	64.7	78.2	75.0
Ours^{-T}	**97.9**	64.8	64.9	86.7	78.6
Ours^{-G}	96.8	69.2	73.3	82.1	80.3
Ours	97.8	**70.0**	**75.1**	87.3	**82.6**

of other approaches. We further discuss this in Section 6.

Results on Digits-DG: Table 3 lists the results on Digits-DG. Although the image categories in Digits-DG differ drastically from those of PACS and Office-Home, there are still clear improvements in the accuracy. Our approach attains clear improvement over state-of-the-art methods in terms of average performance, only the accuracy on SYN is 3% lower than that of FACT [80].

5.2.2 Single-Source Domain Generalization

Unlike some methods [3, 43, 50, 59, 89, 92] which require style labels for learning, our construction of augmentation VOCs does not. This means that our method is applicable to single-source domain generalization tasks. We train models on each domain of PACS and evaluate the performance using the remaining domains, resulting in 12 domain-to-domain transitions in total. As shown in Table 4, our approach performs well on this task, exceeding the second best average performance (SagNet [55]) by around 3%.

5.3. Ablation Study

We examine the effect of adding deformation and texture augmentation separately. To isolate the effect of each component, we use the same parameters in all tests to train the model and only change the augmentation spaces involved. The results are shown in Tables 1-3 (Ours$^{-T-G,-T,-G}$). As can be seen, there are decreases in performance when omitting either augmentation; best results are achieved when both augmentations are used. Overall, texture augmentation increases models' robustness to wider texture styles by reducing the dependency between visual perception and texture patches. Geometric augmentation further enhances the performance by overcoming variations in geometric shapes.

As described in Section 5.1, we use a probability threshold to control whether to use geometric augmentation, and likewise for texture augmentation. In Figure 4, we show the influence of different probability thresholds on the results. We observe that the threshold value makes little difference in performance, but there is a weak peak around 0.5.

To better illustrate the effect of each augmentation on the model, we visualize the feature statistics on PACS using the T-SNE method, as shown in Figure 5, after employing texture and geometric augmentations during the training, boundaries among inter-class objects become cleaner while the intra-class objects cluster closer together.

6. Discussion

The role of geometry and texture in robustness. From the results and analysis above, we can conclude that geometry and texture augmentation improve cross-domain robustness. In terms of their individual influence, texture plays a greater role, but we cannot neglect the role of geometry style – it too is an important contributing factor. Our dual augmentations will encourage the classifier to cope well with both forms of domain shift. Our results also reflect an important property of neural net-

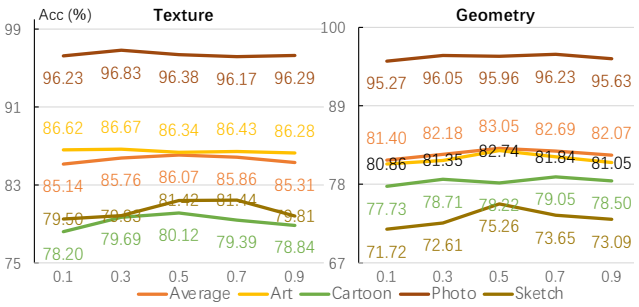

Figure 4. Influence of different probability thresholds for augmentation on classification. Results shown are for multi-source domain test on PACS with ResNet-18 backbone. Statistics on domains with different styles are colored differently.

Table 4. Single-source domain generalization performance on PACS. (A: Art Painting, C: Cartoon, S: Sketch, P: Photo). JiGen [8] results are reproduced with their official code. ADA [74] and SagNet [55] results are reported based on implementations from [55]. ResNet-18 and our results are produced with our implementation.

Methods	A→C	A→S	A→P	C→A	C→S	C→P	S→A	S→C	S→P	P→A	P→C	P→S	Avg.
ResNet-18	56.5	45.6	95.8	59.1	62.6	84.0	20.9	37.1	27.4	60.7	25.4	30.2	50.4
JiGen [8]	57.0	50.0	96.1	65.3	65.9	85.5	26.6	41.1	42.8	62.4	27.2	35.5	54.6
ADA [74]	64.3	58.5	94.5	66.7	65.6	83.6	37.0	58.6	41.6	65.3	32.7	35.9	58.7
SagNet [55]	67.1	**56.8**	95.7	72.1	**69.2**	85.7	41.1	62.9	46.2	**69.8**	35.1	**40.7**	61.9
Ours	**67.4**	51.7	**97.1**	**79.8**	66.4	**89.9**	**50.6**	**70.5**	**58.8**	69.4	**38.7**	39.1	**65.0**

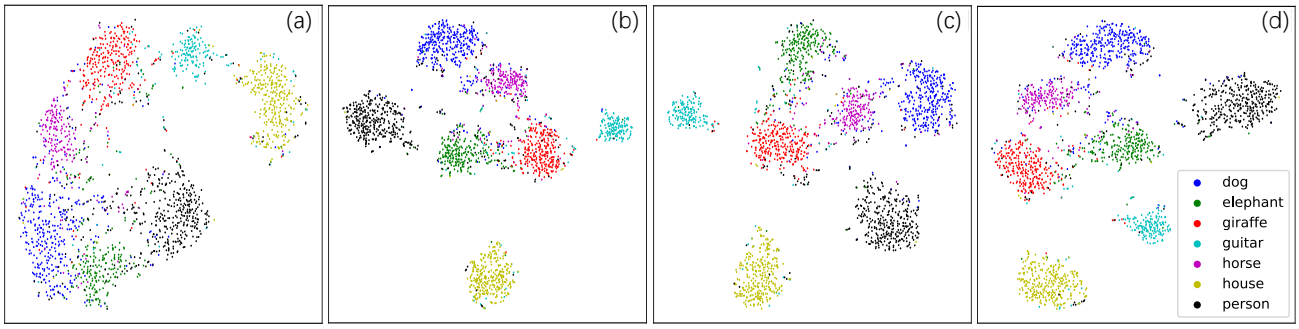

Figure 5. T-SNE visualization of the feature statistics using different augmentations on PACS. (a) Basic training without any augmentation. (b) Add only textural augmentation. (c) Add only geometric augmentation. (d) Using both textural and geometric augmentation. By adding textural and geometric augmentation, object clusters become denser and easier to separate.

works: Texture bias is bigger than geometric bias. According to our experimental results and the findings from some previous researches, there are some possible reasons for this phenomenon:

1. It is related to the sensitivity of CNNs to non-shape features. Studies [2, 27] have shown that CNNs are sensitive to a wide range of image manipulations that have little effect on human judgments.

2. CNNs with texture preferences could indicate an inductive texture bias [18], making it difficult for models to learn geometric-relevant features in small-data regimes, and to generalize to different distributions than the distributions that they were trained on.

The role of geometry and texture in different datasets. Our experimental results also indicate that the effectiveness of geometry and texture augmentations vary from dataset to dataset. One of the biggest reasons is the object and style variances between datasets. Some object classes have geometric shape variances themselves, such as animals in PACS and handwritten numbers in Digits-DG. By comparison, the static objects in Office-Home are of fewer intra-class shape variations. This means their dependencies on geometric style are different.
Limitations. As our geometric and texture distributions are constructed based on the corresponding feature representations of source images, they are strongly dependent on the image quality. If the feature representations are far from being good, the augmentation spaces will be sub-optimal.

In addition, for different tasks such as classification on scene-level, multi-object images, our textural augmentation is applicable but geometric augmentation can not be used directly, as it is likely to introduce distortions without considering the scene content. A potential improved way is to augment individual objects in the scene, which in turn requires object detection, a distinct research area from classification. This is beyond the scope of the present paper, but is a good future direction to explore.

7. Conclusion

In this work, we have presented a framework for domain generalization. Our approach starts from the observation that the differences among images span two aspects: textural appearance differences between domains and geometric shape differences between object class categories. We treat geometry and texture as two complementary roles and have shown that using both to augment the visual object class proves effective in the robustification of classification. Experimental results show that our method outperforms state-of-the-art methods on both multi- and single-source domain generalization tasks. The specific degree of improvement depends on the character of each dataset, but in general, we conclude that widening visual object classes to include geometric style always leads to an improvement.
Acknowledgements. We thank the anonymous reviewers for their insightful comments and suggestions. This work was partially funded by the China Scholarship Council under Grant No. 201906200059.

References

[1] Asha Anoosheh, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. Combogan: Unrestrained scalability for image domain translation. In *CVPR Workshops*, 2018. 2

[2] Nicholas Baker, Hongjing Lu, Gennady Erlikhman, and Philip J Kellman. Deep convolutional networks do not classify based on global object shape. *PLoS computational biology*, 14(12), 2018. 8

[3] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *NIPS*, 2018. 2, 6, 7

[4] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NIPS*, 2019. 2

[5] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *NeurIPS*, 2011. 2

[6] Francesco Cappio Borlino, Antonio D'Innocente, and Tatiana Tommasi. Rethinking domain generalization baselines. In *ICPR*, 2020. 2

[7] Charles F Cadieu, Ha Hong, Daniel LK Yamins, Nicolas Pinto, Diego Ardila, Ethan A Solomon, Najib J Majaj, and James J DiCarlo. Deep neural networks rival the representation of primate it cortex for core visual object recognition. *PLoS computational biology*, 10(12):e1003963, 2014. 1

[8] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, 2019. 2, 6, 7, 8

[9] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. In *NIPS*, 2021. 2

[10] Tian Qi Chen and Mark Schmidt. Fast patch-based style transfer of arbitrary style. *Workshop in Constructive Machine Learning, NIPS*, 2016. 2

[11] Sanghyuk Chun and Song Park. Styleaugment: Learning texture de-biased representations by style augmentation without pre-defined textures. `arXiv:2108.10549[cs.CV]`, 2021. 2

[12] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *NIPS*, 2019. 2, 6

[13] Yingjun Du, Jun Xu, Huan Xiong, Qiang Qiu, Xiantong Zhen, Cees GM Snoek, and Ling Shao. Learning to learn with variational information bottleneck for domain generalization. In *ECCV*, 2020. 2

[14] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In *ICLR*, 2016. 5

[15] Chuang Gan, Tianbao Yang, and Boqing Gong. Learning attributes equals multi-source domain generalization. In *CVPR*, 2016. 2

[16] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015. 6

[17] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. 2, 4

[18] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019. 1, 8

[19] Robert Geirhos, Carlos R Medina Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A Wichmann. Generalisation in humans and deep neural networks. *NIPS*, 2018. 1

[20] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. In *BMVC*, 2017. 4, 5

[21] Muhammad Ghifary, David Balduzzi, W Bastiaan Kleijn, and Mengjie Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *PAMI*, 39(7):1414–1430, 2016. 2

[22] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. Dlow: Domain flow for adaptation and generalization. In *CVPR*, 2019. 2

[23] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *ICLR*, 2021. 1

[24] Peter Hall, Hongping Cai, Qi Wu, and Tadeo Corradi. Cross-depiction problem: Recognition and synthesis of photographs and artwork. *Computational Visual Media*, 1(2):91–103, 2015. 1, 2

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[26] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *ICLR*, 2020. 2

[27] Katherine L Hermann, Ting Chen, and Simon Kornblith. The origins and prevalence of texture bias in convolutional neural networks. *NIPS*, 2020. 1, 8

[28] Minui Hong, Jinwoo Choi, and Gunhee Kim. Stylemix: Separating content and style for enhanced data augmentation. In *CVPR*, 2021. 2

[29] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 2, 4

[30] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *ECCV*, 2020. 2

[31] Philip TG Jackson, Stephen Bonner, Toby P Breckon, and Boguslaw Obara. Style augmentation: data augmentation via style randomization. In *CVPR Workshop*, 2019. 2, 4

[32] Yunpei Jia, Jie Zhang, Shiguang Shan, and Xilin Chen. Single-side domain generalization for face anti-spoofing. In *CVPR*, 2020. 2

[33] Xin Jin, Cuiling Lan, Wenjun Zeng, Zhibo Chen, and Li Zhang. Style normalization and restitution for generalizable person re-identification. In *CVPR*, 2020. 2

[34] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012. 1

[35] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning*, 2020. 2

[36] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In *CVPR*, 2020. 2

[37] Sunnie SY Kim, Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. Deformable style transfer. In *ECCV*, 2020. 1, 2

[38] Jogendra Nath Kundu, Akshay Kulkarni, Amit Singh, Varun Jampani, and R. Venkatesh Babu. Generalize then adapt: Source-free domain adaptive semantic segmentation. In *ICCV*, 2021. 2

[39] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017. 1

[40] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[41] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017. 1, 5

[42] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018. 2

[43] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *ICCV*, 2019. 2, 6, 7

[44] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *CVPR*, 2018. 2, 6, 7

[45] Yiying Li, Yongxin Yang, Wei Zhou, and Timothy Hospedales. Feature-critic networks for heterogeneous domain generalization. In *ICML*, 2019. 2

[46] Yingwei Li, Qihang Yu, Mingxing Tan, Jieru Mei, Peng Tang, Wei Shen, Alan Yuille, and Cihang Xie. Shape-texture debiased neural network training. *ICLR*, 2021. 1

[47] Hubert Lin, Mitchell van Zuijlen, Sylvia C Pont, Maarten WA Wijntjes, and Kavita Bala. What can style transfer and paintings do for model robustness? In *CVPR*, 2021. 2, 4

[48] Alexander H Liu, Yen-Cheng Liu, Yu-Ying Yeh, and Yu-Chiang Frank Wang. A unified feature disentangler for multi-domain image translation and manipulation. In *NIPS*, 2018. 2

[49] Xiao-Chang Liu, Yong-Liang Yang, and Peter Hall. Learning to warp for style transfer. In *CVPR*, 2021. 1, 2, 3

[50] Toshihiko Matsuura and Tatsuya Harada. Domain generalization using a mixture of multiple latent domains. In *AAAI*, 2020. 6, 7

[51] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *ICCV*, 2017. 2, 6, 7

[52] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, 2013. 1, 2

[53] Chaithanya Kumar Mummadi, Ranjitha Subramaniam, Robin Hutmacher, Julien Vitay, Volker Fischer, and Jan Hendrik Metzen. Does enhanced shape bias improve neural network robustness to common corruptions? *ICLR*, 2021. 2

[54] Hyeonseob Nam and Hyo-Eun Kim. Batch-instance normalization for adaptively style-invariant neural networks. In *NIPS*, 2018. 2

[55] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *CVPR*, 2021. 1, 2, 4, 6, 7, 8

[56] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. 6

[57] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *ECCV*, 2018. 2

[58] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. 2

[59] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. In *ICML*, 2020. 6, 7

[60] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *CVPR*, 2020. 2

[61] Mohammad Mahfujur Rahman, Clinton Fookes, Mahsa Baktashmotlagh, and Sridha Sridharan. Multi-component image translation for deep domain generalization. In *WACV*, 2019. 2

[62] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019. 1

[63] Jongbin Ryu, Gitaek Kwon, Ming-Hsuan Yang, and Jongwoo Lim. Generalized convolutional forest networks for domain generalization and visual recognition. In *ICLR*, 2019. 2

[64] Seonguk Seo, Yumin Suh, Dongwan Kim, Geeho Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization. In *ECCV*, 2020. 2

[65] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *ICLR*, 2018. 2, 6, 7

[66] Rui Shao, Xiangyuan Lan, Jiawei Li, and Pong C Yuen. Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *CVPR*, 2019. 2

[67] Baifeng Shi, Dinghuai Zhang, Qi Dai, Zhanxing Zhu, Yadong Mu, and Jingdong Wang. Informative dropout for robust representation learning: A shape-bias perspective. In *ICML*, 2020. 6

[68] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019. 2

[69] Nathan Somavarapu, Chih-Yao Ma, and Zsolt Kira. Frustratingly simple domain generalization via image stylization. `arXiv:2006.11207[cs.CV]`, 2020. 2

[70] Jifei Song, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Generalizable person re-identification by domain-invariant mapping network. In *CVPR*, 2019. 2

[71] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 1

[72] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017. 5

[73] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, 2019. 2

[74] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *NIPS*, 2018. 6, 8

[75] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, and Tao Qin. Generalizing to unseen domains: A survey on domain generalization. *IJCAI*, 2021. 2

[76] Shujun Wang, Lequan Yu, Caizi Li, Chi-Wing Fu, and Pheng-Ann Heng. Learning from extrinsic and intrinsic supervisions for domain generalization. In *ECCV*, 2020. 6

[77] Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. Learning to diversify for single domain generalization. In *ICCV*, 2021. 2

[78] John Willats. *Art and representation: New principles in the analysis of pictures*. Princeton University Press, 1997. 2

[79] Sebastien C Wong, Adam Gatt, Victor Stamatescu, and Mark D McDonnell. Understanding data augmentation for classification: when to warp? In *International conference on digital image computing: techniques and applications (DICTA)*. IEEE, 2016. 2

[80] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *CVPR*, 2021. 6, 7

[81] Zhenlin Xu, Deyi Liu, Junlin Yang, Colin Raffel, and Marc Niethammer. Robust and generalizable visual representation learning via random convolutions. *ICLR*, 2021. 2

[82] Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the national academy of sciences*, 111(23):8619–8624, 2014. 1

[83] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *ICCV*, 2019. 2, 4

[84] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 2

[85] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *ICLR*, 2018. 2

[86] Xu Zheng, Tejo Chalasani, Koustav Ghosal, Sebastian Lutz, and Aljosa Smolic. Stada: Style transfer as data augmentation. In *International Conference on Computer Vision Theory and Applications*, 2019. 4

[87] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization in vision: A survey. `arXiv:2103.02503[cs.LG]`, 2021. 2

[88] Kaiyang Zhou, Chen Change Loy, and Ziwei Liu. Semi-supervised domain generalization with stochastic stylematch. `arXiv:2106.00592[cs.CV]`, 2021. 2, 4

[89] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *AAAI*, 2020. 5, 6, 7

[90] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *ECCV*, 2020. 6, 7

[91] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain adaptive ensemble learning. *IEEE Transactions on Image Processing (TIP)*, 2021. 6

[92] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021. 2, 4, 6, 7

[93] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 4