Image Translation using Texture Co-occurrence and Spatial Self-Similarity for Texture Debiasing

Myeongkyun Kang¹, Dongkyu Won¹, Miguel Luna¹, Philip Chikontwe¹, Kyung Soo Hong², June Hong Ahn², Sang Hyun Park¹*  

¹ Robotics Engineering, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, Korea  
{mkkang,shpark13135}@dgist.ac.kr  
² Division of Pulmonology and Allergy, Department of Internal Medicine, Regional Center for Respiratory Diseases, Yeungnam University Medical Center, College of Medicine, Yeungnam University, Daegu, Korea  
fireajh@yu.ac.kr

Abstract. Classification models trained on datasets with texture bias usually perform poorly on out-of-distribution samples since biased representations are embedded into the model. Recently, various debiasing methods have attempted to disentangle biased representations, but discarding texture biased features without altering other relevant information is still a challenging task. In this paper, we propose a novel texture debiasing approach to generate additional training images using the content of a source image and the texture of a target image with a different semantic label to explicitly mitigate texture biases when training a classifier. Our model ensures texture similarity between the target and generated images via a texture co-occurrence loss while preserving the content details from the source image with a spatial self-similarity loss. Both the generated and original training images are combined to train an improved classifier that is robust against inconsistent texture bias representations. We employ five datasets with known texture biases to demonstrate the ability of our method to mitigate texture bias. In all cases, our method outperformed existing state-of-the-art methods.

Keywords: Unpaired Image Translation · Debiasing · Self-Similarity · Texture Co-occurrence.

1 Introduction

Texture biases can be easily and unintentionally introduced during the data collection process. The source and properties of such biases are usually unknown, and can lead to a significant decrease in performance when a model trained on biased data is applied to out-of-distribution data [34]. In addition, Geirhos et al. [10] showed that common convolutional neural networks (CNNs) prioritize texture information over content features (e.g., shape), making them especially...
vulnerable to texture biased images. Therefore, transferring texture features between images of different classes allows a classifier to adjust its internal learned representation to be less dependent on biased information. As shown in Fig. 1(a), if a binary classifier is trained with images that have a distinct texture for each class (e.g., five is colored and six is grayscale), it is highly likely that the model will consider the texture representations as a differentiator between the two classes rather than the actual shape. In this case, the model will perform poorly on test data that does not include similar textures observed in the training data.

Bias mitigation or debiasing has been previously addressed by several methods that proposed to extract bias-independent features through adversarial learning and enabled models to solve the intended classification task without relying on biased features (e.g., color information) [21,45]. However, texture biased representations are often retained after training, since it is difficult to completely disentangle biased features through adversarial learning. In addition, prior knowledge regarding the types of biases in the training data is essential for adversarial learning, but it is often unknown and difficult to identify, especially when the data has been collected from multiple sources [3,4,12,28].

In this paper, we propose to debias a dataset by generating additional images using the existing training images. Specifically, we combine the structural information of a source image with the texture of another labeled image to extend the dataset and mitigate bias. Thus, the extended dataset helps the classifier learn class related features and avoid relying on biased features to perform correct data separation. As shown in Fig. 1(b), a debiased classifier can be trained without bias labels by combining the original data and generated images containing the textures of images with different labels.

Modern generative adversarial networks (GAN) [51,29,15,24,22,34] have shown remarkable success in effectively transferring textures (or styles) between pairs of images, but they also tend to adjust content information since no explicit supervision is provided during training. For instance, in Fig. 1, the ideal scenario would be the colored digit five is translated to black and white, but is instead translated into a six that matches the black and white style. In general, transferring only texture information without content discrepancy is very challenging, and poor results will adversely affect the classifier.
To address this issue, we propose a novel GAN that simultaneously optimizes spatial self-similarity between source and generated images, and texture similarity between the generated and target images, where the source and target images have different labels. The proposed GAN consists of an image generator with content and texture encoders, two discriminators that constrain texture similarity in both local and global views, and a pre-trained VGG [40] network to enforce spatial self-similarity. The generation of high quality images with the intended properties is achieved by using a spatial self-similarity loss to ensure content consistency, a texture co-occurrence loss and GAN loss to enforce similar local and global textures in the target image. Once generated images have been obtained using the training data, a debiased classifier can be learned using all the available data. Our main contributions are listed below:

- We introduce a method that learns a debiased classifier by extending the training dataset with additional images that incorporate textures from other classes while retaining structural information of its own class. Our method does not require any bias labeling and can effectively mitigate unknown texture biases during training.
- We propose a novel image generation method employing texture co-occurrence and spatial self-similarity losses. Our results demonstrate how using both losses can jointly generate high quality images that successfully combine structure and textures from separate images. We also show that optimizing both losses can produce images that are effective for debiasing.
- We show that our method outperforms existing debiasing methods and produces high quality images compared to prior generative models on five distinct texture biased datasets.

2 Related Works

Texture-Bias in CNN: Geirhos et al. [10] argued that texture bias mitigation is necessary to ensure the reliability of a classifier, since common CNNs prioritize texture information over content (shape). Especially in the task of domain generalization [49], texture bias has gained significant interest since changes in image texture are the main reason for domain shift [44,32,43,25,37]. Nuriel et al. [33] and Zhou et al. [50] applied feature-level adaptive instance normalization (AdaIN) [14] by shuffling (or swapping) the features of training samples across source domains to improve the generalizability of the trained model. Similarly, Nam et al. [32] introduced content- and style-biased networks that randomize the content and style (texture) features between two different samples via AdaIN. To obtain features robust to style-bias, they also leveraged adversarial learning to prevent the feature extractor from retaining style-biased representations. In general, prior methods only focused on the generalization performance of inaccessible-domain samples, and thus design models to learn common object features from multiple source domains. However, if the sample is biased along the label e.g., colored digit five and grayscale digit six in Fig. 1 existing methods
may fail to learn common object features, and cannot effectively ignore texture biases present in the training dataset. In this paper, we address this by training a classifier to only focus on the intended task without using bias information present in the training data.

**Image Generation and Style Transfer Models:** Zhu et al. [51] and Liu et al. [29] proposed deep generative models using a pair of generators and discriminators that translate one domain into another using cycle-consistency and a shared-latent space assumption, respectively. Huang et al. [15] proposed a multi-domain translation model by varying the style code with fixed content for diverse style image generation. For feature disentanglement, Lee et al. [24] proposed a content and attribute encoder with a cross-cycle consistency loss to enforce consistency between domains. Furthermore, Kolkin et al. [22] proposed an optimization-based style transfer method using the concept of self-similarity. Park et al. [34] proposed a conditional image translation method with a contrastive loss to maximize the mutual information between positive and negative patches, whereas Zheng et al. [48] proposed a spatially-correlative loss for consistent image translation to preserve scene structures. However, as most GAN-based methods focused on transferring image style from one domain to another instead of maintaining image content; spatial discrepancy or texture corruption is observable in the generated images. Unlike prior generation models, our model only updates textures while preserving content by minimizing a spatial self-similarity loss [22].

**Debiasing and Fairness Aware Models:** To mitigate bias, Alvi et al. [2] employed a bias prediction layer to make latent features indistinguishable using a confusion loss [42]. Kim et al. [21] used a gradient reversal layer [7] that minimizes the mutual information of bias predictions to constrain the use of bias-related features for classification. In addition, few related works [39,5] used the fairness term in GAN. However, these methods were proposed for the generation of a fraction (e.g., 10%) of protected attribute samples for a model trained on a biased dataset. Moreover, since these methods could not generate images with a specific texture, we employ an image translation technique that can explicitly generate an image with the texture of a target image. On the subject of fairness aware training, Li et al. [27] formulated bias minimization in terms of data re-sampling to balance the preference of specific representations for classification. On the other hand, Wang et al. [45] employed an adversarial learning approach [7] to remove protected attributes correspondence (e.g., gender) in the intermediate features of the model. Moreover, Louppe et al. [30] proposed an adversarial network to enforce the pivotal property (e.g., fairness) on a predictive model, and Zhang et al. [46] proposed three terms, i.e., primary, adversary, and projection, to improve the stability of debias training. However, these methods require bias labels for training and stability in adversarial learning is often hard to achieve. In contrast, our method solves the bias problem by explicitly utilizing texture generated images and only requires images of different domains. Thus, our approach does not require laborious bias labeling and is free from intractable adversarial classifier training.
Fig. 2. Our proposed image generation model. Our model consists of a content encoder $E_c$, texture encoder $E_t$, StyleGAN2 generator $G$, ImageNet pre-trained VGG model, and two discriminators $D$ and $D_{patch}$. Each encoder extracts the content feature $c_i$ from an image $X_i$ and the texture feature $t_j$ in an image $X_j$ with a different label, respectively. $G$ generates an image $X'_i$ using $c_i$ and $t_j$ with a spatial self-similarity loss $L_{spatial}$, a texture co-occurrence loss $L_{texture}$, and a GAN loss $L_{GAN}$.

3 Method

Given an image dataset $D = \{X_1, ..., X_n\}$ with binary classification labels $y_i \in \{0, 1\}$, and a bias property $b_i \in \{0, 1\}$ aligned with the labels i.e., $y_i = b_i$, our main goal is to build an augmented dataset $D' = \{X'_1, ..., X'_n\}$ that reduces the importance of $b_i$ during the classification process. In this work, we consider texture as the dominant property that creates bias during the classification process and prevents the model from using shape information of the object of interest. Therefore, we propose a generative data augmentation framework that updates an input image $X_i$ using the texture of a randomly selected image with a different label i.e., $X_j, y_i \neq y_j$, while retaining the content information of $X_i$.

Our framework consists of three main sections as shown in Fig. 2. First, $X_i$ and $X_j$ are encoded by a content encoder $E_c$ and a texture encoder $E_t$, respectively. Then, the encoded features are used to generate a target augmented image $X'_i$ via $G$, and a combination of two discriminators with an ImageNet pre-trained VGG model [40] ensure the generated image $X'_i$ has the texture of $X_j$ while retaining the content information of $X_i$. Modifying texture information while maintaining the content requires the use of additional terms in the standard adversarial objective. Thus, we add a texture co-occurrence loss term to enable correct texture transfer and a spatial self-similarity loss term to ensure the original content is unchanged. The generator $G$ and discriminator $D$ follow the architecture proposed in StyleGAN2 [20], and are used to compute the adversarial loss between $X'_i$ and $X_j$. Texture co-occurrence loss between $X_i$ and $X_{j,y_i \neq y_j}$ is computed by the patch-discriminator $D_{patch}$, and the spatial self-similarity loss between $X_i$ and $X'_i$ is computed by the VGG model, respectively.
Finally, the original dataset $D$ is combined with its augmented version $D'$ to train a classifier that is robust to inconsistent texture bias representations for classification, and can avoid texture bias present in the training data.

### 3.1 Image Generation using Content and Texture

Generating a realistic image using the content information in $X_i$ and the texture of $X_j$ requires the extraction of specific types of features. To this end, we use two different encoders $E_c$ and $E_t$ to extract a content-encoded tensor $c_i$ and a texture-encoded vector $t_j$, respectively. Then, $G$ generates a texture transferred image $X'_i$ by taking $c_i$ and $t_j$ as a constant and style vector following Karras et al. [20], and discriminator $D$ enforces a non-saturating adversarial loss [11] for generative training. The adversarial loss is defined as:

$$L_{GAN}(E_c, E_t, G, D) = \mathbb{E}_{X_i, X_j \sim \mathcal{D}, y_i \neq y_j}[-\log(D(G(c_i, t_j)))]$$

However, this setting often fails to retain content information since the discriminator is heavily enforced to generate an image that preserves entangled content features in the target domain. Thus, we add additional constraints to ensure generator $G$ retains content information in $c_i$ and uses texture information in $t_j$. These constraints take the form of two additional modules and loss terms, i.e., a texture co-occurrence loss and spatial self-similarity loss.

### 3.2 Texture Co-occurrence Loss

To encourage the transfer of texture information $t_j$ from $X_j$ to $X'_i$, i.e., $G(c_i, t_j)$, we employ a texture co-occurrence loss with a patch-discriminator $D_{patch}$ that measures the texture difference between $X'_i$ and $X_j$. The texture co-occurrence loss and patch discriminator $D_{patch}$ were initially proposed for image editing [35] to disentangle texture information from structure. $D_{patch}$ encourages joint feature statistics to appear perceptually similar [16, 17, 8, 35], and can be achieved by cropping multiple random patch sizes between $1/8$ to $1/4$ of the full image size, and feeding them to $D_{patch}$. In particular, we average the features in $X_j$ patches and concatenate with the features of $X'_i$, and feed them to the last layers of $D_{patch}$ to calculate the discriminator loss. Consequently, $G$ is enforced to satisfy the joint statistics of low-level features for consistent texture transfer. Formally,

$$L_{texture}(E_c, E_t, G, D_{patch}) = \mathbb{E}_{X_i, X_j \sim \mathcal{D}, y_i \neq y_j}[-\log(D_{patch}(\text{crop}(X'_i), \text{crop}(X_j)))]$$

### 3.3 Spatial Self-Similarity Loss

To retain the content information of the source image, we employ spatial self-similarity as a domain invariant content constraint. The self-similarity loss has
been used to maintain the structure of content images in artistic image style transfer [22]. Formally, we consider a spatial self-similarity map as follows:

\[ S_{X_i} = f^{T}_{X_i} \cdot f_{X_i}, \]

(3)

where \( f_{X_i} \in \mathbb{R}^{C \times H \times W} \) denotes the spatially flattened features extracted from \( VGG \) with channel \( C \), height \( H \), and width \( W \), respectively. By applying the dot product, \( S_{X_i} \in \mathbb{R}^{H \times W \times H \times W} \) captures spatial correlation from one location \((\mathbb{R}^C)\) to the rest in all feature maps. Thus, our domain invariant content constraint (spatial self-similarity loss) can be calculated between \( X'_i \) and \( X_i \) as follows:

\[ L_{spatial}(E_c, E_t, G) = \mathbb{E}_{X_i, X_j \sim \mathcal{D}, y_i \neq y_j} [ \|1 - \cos(S_{X_i}, S_{X'_i})\|_1 ], \]

(4)

where \( \cos \) denotes the cosine similarity. In conventional generative models, the reconstruction loss (e.g., L1, MSE) or perceptual losses are used to provide constraints using the entangled feature (contents and texture) for minimizing content discrepancy (Fig. 3). Since content information can be decoupled from texture information using spatial self similarity, we can explicitly control the content features in \( G \). Consequently, we can successfully preserve the content of \( X_i \) while updating texture.

### 3.4 Full Objective and Implementation Details

The objective function of our framework is defined as:

\[ L_{total} = \lambda_g L_{GAN} + \lambda_t L_{texture} + \lambda_s L_{spatial}, \]

(5)

where hyper-parameters \( \lambda_g \), \( \lambda_t \), and \( \lambda_s \) balance the importance of spatial and texture loss terms, respectively. We set \( \lambda_g = 0.1 \), \( \lambda_t = 1.0 \), and \( \lambda_s = 100 \), respectively. The objective and network design of discriminators \( D \) and \( D_{patch} \) closely follow StyleGAN2 [20], and 8 patches are used in the texture co-occurrence loss. \( E_c \) and \( E_t \) downsample their inputs 2x and 6x each to extract \( c_i \) and \( t_j \) (1x and 4x for small resolution inputs e.g., digit), respectively. For \( t_j \), we explicitly discard spatial information by applying global average pooling. To reduce computational costs, we select 256 random features from \( f^{T}_{X_i} \) to obtain \( \hat{f}^{T}_{X_i} \), thus reducing the size of the self-similarity map, i.e., \( \hat{S}_{X_i} = \hat{f}^{T}_{X_i} \cdot f_{X_i} \). Herein, \( \hat{S}_{X_i} \in \mathbb{R}^{256 \times H \times W} \) is used to calculate \( L_{spatial} \) instead of \( S_{X_i} \in \mathbb{R}^{H \times W \times H \times W} \) (generally \( 256 < H \times W \)). To construct \( D' \), we randomly select texture sources from images with a different label.
3.5 Extension to Multi-domain Debiasing

To debias a multi-domain biased dataset, the number of models required for texture updates is a factorial of the number of domains in the set. In other words, it is challenging to use the proposed method on datasets with a large number of labels (domains). Herein, we introduce a conditional version of the proposed method constructed by using 2D embedding layers that simply change the statistics of an intermediate feature $\mathcal{F}$ of CNNs. We placed the 2D embedding layer at the second and pre-last layer of $E_c$, $E_t$, and $D$. The domain label is fed into a 2D embedding layer and returns 1D vectors (weight $e_w$ and bias $e_b$ in the same dimensions as $\mathcal{F}$) to modify $\mathcal{F}$. The updated features $\mathcal{F}'$ are calculated as $\mathcal{F} \times e_w + e_b$, and are fed to the next layer instead of $\mathcal{F}$. Consequently, a condition is provided to the model.

4 Experiments

4.1 Datasets

Existing datasets often have similar distributions between training and testing data [3,4], making them unsuitable for direct bias mitigation analysis. Thus, we constructed five texture biased datasets following previous works [2,21], where the training and testing sets have opposite distributions allowing us to focus our evaluation solely on texture bias mitigation (Fig. 4). Note that, in the balanced setting, the model will perform well on the same distribution samples, but may present some errors on the remainder. Thus, our evaluation used the inverse distribution to quantify the true performance in the most extreme cases. On the other hand, to avoid indirectly learning about the existing bias during training, the validation set also has the same distribution as the training set, thus high accuracy on the validation set does not reveal any robustness to bias.

**Five vs. Six:** This set is based on MNIST [23] and MNIST-M [7] datasets where only the numbers five and six are used to construct a training split. Images with a five are taken solely from MNIST-M, whereas those with six are from MNIST, the opposite case is considered for the testing split.

**Dogs vs. Cats:** This dataset was originally proposed by Kim et al. [21] by aligning the hair color differences of animals. Images with bright dogs and dark
cats images are used in the training split, whereas dark dogs and bright cats are employed for testing. Consequently, the dataset exhibits hair color bias. **COVID-19 vs. Bacterial pneumonia:** Pneumonia caused by different pathogens requires specific treatments. However, images obtained by computed tomography (CT) scans might share similar properties making accurate diagnosis challenging. One source of ambiguity for automatic diagnostic systems is the choice of the CT protocol during the image acquisition process, i.e., using a contrast agent vs. another protocol. To evaluate our method for real-world texture bias, CT scans were collected at anonymous medical center. In this dataset, we selected Non-contrast COVID-19 for training and contrasted scans for testing. The opposite applies for Bacterial pneumonia scans. 

**Multi-class Biased Datasets:** To evaluate our method on a multi-class texture biased dataset, we constructed Digit and Biased PACS datasets. In the Digit dataset, samples with labels from zero to four are taken from MNIST [23], and those with five to nine from MNIST-M [7]. Biased PACS dataset was constructed using PACS dataset [25], which consists of four domains (Photo, Art, Cartoon, Sketch) and has seven classes. We selected the top four labels (Dog, Elephant, Giraffe, Horse) after sorting the number of images. Each class is taken from a different domain (e.g., Dog - Cartoon, Elephant - Sketch) for the training dataset, and the remaining domain images are used for the test set. Additionally, we constructed an Inverse Biased PACS dataset that inversely uses the biased PACS dataset, i.e., replace the test split samples with the training samples, and replace the train/val split samples with the testing samples, respectively. Compared to the binary-class biased datasets, this dataset has access to samples in three domains during training and be evaluated on samples from a single domain that does not match the training data.

### 4.2 Experimental Settings

For our baseline, we train an ImageNet pre-trained ResNet50 classifier only using the biased training data without any debiasing methods. For COVID-19 vs. Bacterial pneumonia, we aggregated slice predictions via majority voting to obtain patient-level diagnosis. We evaluated accuracy using the macro F1-score, which treats the label distribution equally. For consistency, training was repeated three times, and we report the average performance with the standard deviation of each method as the final performance.

We compare our method against five conventional debiasing methods [30, 24, 45, 21], two domain generalization methods [33, 50], and five generative models [51, 29, 15, 24, 34]. For non-generative debiasing methods, bias information (e.g., data-source, color, and CT protocol) was used to train the classifier. For generative models, we constructed augmented datasets using each generative model and employed them to train classifiers. For fair evaluation, the classifiers in the compared methods were trained using the same backbone (ResNet50) under the same training settings. In the multi-domain data scenario, we did not perform experiments for methods addressing binary classification, as well as generation methods [51, 29, 15, 24, 34] that had to learn models for all domain labels.
Table 1. Classification performance of non-generative debiasing methods (The first sub-row), domain generalization methods (The second sub-row), image generative models (The third sub-row), and the proposed model (The fourth sub-row) on five datasets. F1-score with ± std was used as the metric. The second and third sub-columns indicate binary- and multi-class biased dataset classification performances, respectively.

| Method       | (a) Five vs. | (b) Dogs vs. | (c) COVID vs. | (d) Digit | (e) B. Pneumonia | (f) In. B. Pneumonia |
|--------------|--------------|--------------|---------------|-----------|------------------|-----------------------|
| Baseline [13] | 1.34 ± 0.04  | 25.47 ± 2.28 | 18.61 ± 7.81  | 3.68 ± 1.04 | 12.07 ± 2.83     | 27.01 ± 2.67          |
| Learn to [29] | 1.14 ± 0.54  | 27.95 ± 2.24 | 16.63 ± 7.59  | -          | -                | -                     |
| Art. Domain [47] | 1.37 ± 0.21  | 27.42 ± 3.06 | 19.71 ± 8.55  | -          | -                | -                     |
| BlindEye [27] | 1.60 ± 0.70  | 37.13 ± 4.74 | 25.10 ± 4.67  | 3.93 ± 1.26 | 10.12 ± 1.34     | 28.33 ± 3.24          |
| Not Enough [13] | 2.32 ± 0.49  | 22.00 ± 1.24 | 21.97 ± 4.24  | 6.47 ± 3.24 | 10.25 ± 1.51     | 17.58 ± 12.46         |
| CUT [44]      | 1.85 ± 1.91  | 20.57 ± 1.24 | 19.15 ± 1.24  | 11.02 ± 1.34 | 12.10 ± 2.21     | 24.01 ± 5.49          |
| C-AFIN [33]   | 1.67 ± 0.70  | 37.41 ± 4.60 | 59.71 ± 6.58  | 1.85 ± 0.61 | 11.63 ± 1.02     | 37.46 ± 5.69          |
| MixStyle [29] | 16.47 ± 2.36 | 72.51 ± 3.08 | 55.18 ± 7.96  | 5.74 ± 2.38 | 10.39 ± 0.96     | 34.80 ± 4.44          |
| CYGAN [44]    | 28.92 ± 4.17 | 89.96 ± 1.20 | 69.18 ± 2.84  | 5.90 ± 0.24 | -                | -                     |
| UNIT [13]     | 1.09 ± 0.96  | 38.90 ± 2.19 | 35.96 ± 4.34  | 6.90 ± 0.84 | -                | -                     |
| MUNIT [29]    | 1.48 ± 0.31  | 46.82 ± 1.81 | 50.54 ± 7.91  | 8.63 ± 1.08 | -                | -                     |
| DRIT++ [44]   | 23.92 ± 2.47 | 82.87 ± 1.31 | 74.91 ± 2.73  | 10.30 ± 0.93 | -                | -                     |
| Cut [44]      | 9.50 ± 0.76  | 80.19 ± 1.24 | 72.76 ± 3.45  | 2.95 ± 0.75 | -                | -                     |
| Proposed      | 15.14 ± 3.52 | 91.80 ± 0.44 | 76.50 ± 3.38  | 58.82 ± 1.27 | -                | -                     |
| Proposed (Ext.) | 12.63 ± 2.92 | 90.90 ± 0.53 | 70.30 ± 2.91  | 63.52 ± 0.41 | 36.46 ± 1.24     | 48.48 ± 4.09          |

5 Results

Five vs. Six: In this task, none of the compared models yielded satisfactory results (see Table 1(a)). Non-generative models report extremely low scores, leading us to conclude that texture features were still used for classification instead of the digit shape. Furthermore, domain generalization methods also show poor F1-scores. We observed that generative models mainly transferred texture and shape jointly (See Fig. 5). Consequently, the resulting augmented dataset had several instances with the shape of a different category. As the classifier does not have a clear cue to distinguish between numbers, this led to poor performance. Results on this dataset clearly show the benefit of using our method to transfer texture features while retaining the underlying information for successful bias mitigation.

Dogs vs. Cats: Performing classification on real-world animals requires more complex features than color alone. In Table 1(b), we observed an increase in performance when a debiasing- or domain generalization method was applied. Overall, generative models show better performance over the non-generative counterparts, i.e., F1-score +10%. While the performance of generative models was reasonable, several limitations were noted. In the case of CycleGAN, CUT, and DRIT++, unsatisfactory images were obtained, i.e., texture was not entirely translated and distortions were present. Moreover, UNIT and MUNIT generate images with small texture updates, thus bias remains in the generated images (see Fig. 5). On the other hand, our model simultaneously shows improved performance with high-quality image generation. We believe this is mainly due to the proposed texture translation strategy. Leveraging both texture co-occurrence and spatial self-similarity losses enabled our model to generate consistent and natural images leading to improved classification performance.

COVID-19 vs. Bacterial Pneumonia: In contrast to the results observed on the Dogs vs. Cats task, non-generative models report improved performance
Our model uses the content from the source image, and the texture from an image with a different label. For each dataset, same row image is used as source, whereas the texture is from the opposite pair. Our model successfully transfers texture while retaining the content.

| Input | Cycle | UNIT | MUNIT | DRIT | CUT | Ours |
|-------|-------|------|-------|------|-----|------|
| Five  | 5     | 6    | 6     | 6    | 5   | 5    |
| Six   | 6     | 5    | 6     | 5    | 6   | 6    |
| Five  | 5     | 6    | 6     | 6    | 5   | 5    |
| Six   | 6     | 5    | 6     | 5    | 6   | 6    |
| Cats  | Cats  | Cats | Cats  | Cats | Cats| Cats |
| Dogs  | Dogs  | Dogs | Dogs  | Dogs | Dogs| Dogs |
| Bact. | Bact. | Bact.| Bact. | Bact.| Bact.| Bact.|
| COVID | COVID | COVID| COVID | COVID| COVID| COVID|

Fig. 5. Qualitative results of five generative models and the proposed. Our model uses the content from the source image, and the texture from an image with a different label. For each dataset, same row image is used as source, whereas the texture is from the opposite pair. Our model successfully transfers texture while retaining the content.

over the rest (Table 1(c)). Domain generalization methods obtain lower F1-scores than non-generative approaches, whereas non-generative models mitigate distinctly recognizable bias in the image such as color over generative models, since they do not impose any structural changes in the image. On the other hand, generative-based models tend to modify patterns on regions of diagnostic interest, thus losing the properties that identify the disease, leading to lower performance (See Fig. 5). For example, CycleGAN tends to erase (row 10, 12) or create lesions (row 9) in the CT scans. UNIT and MUNIT show minor texture updates, hence are insufficient to mitigate bias in the classifier during training (all rows). DRIT++ and CUT create artifacts such as non-existent lesions, checkerboards and change the images’ properties (row 9, 10, 12), resulting in the lowest classification accuracy among the generative models. Meanwhile, our
method jointly optimizes the texture co-occurrence and spatial self-similarity losses, each imposing structural and texture constraints for image generation. Herein, our method can successfully update texture without introducing artifacts in the original CT scan. Note that while contrast CT is a standard protocol for common lung disease diagnosis, curating contrast CT is challenging since extra processes such as contrast agent injection and disinfection are required [38,19], especially during the pandemic. Finally, we believe texture biases will be unexpectedly or sometimes unavoidably be introduced during the data collection process; thus, a debiasing method that can maintain key structures in their images is vital in the medical domain.

**Multi-class Biased Datasets:** For the multi-class Digit dataset, none of non-generative models and domain generalization methods report satisfactory performance (see Table 1(d)). On the other hand, our method achieved the best score over all compared methods (i.e., 68.82%). On Biased PACS dataset, all comparison methods report F1-scores around 10%, while our method significantly improved to 36.68% (Table 1(e)). The translated images from Photo to other domains are satisfactory as shown in Fig. 6(a), whereas Sketch to Photo (and Art) have differences between the realistic Photo (and Art) images as shown in Fig. 6(b). We found that this is due to the scale of Biased PACS dataset (small with only a few hundred images). Despite this, our method achieves reasonable generation quality, retains content and accurately transfers textures for debiasing. On Inverse Biased PACS dataset, even though non-generative models and domain generalization methods utilize multiple domain samples for training, they perform poorly on out-of-distribution samples as shown Table 1(e’). This highlights the difficulty of learning domain agnostic features. In summary, we empirically show that images generated with our method can help the classifier mitigate texture biases towards improved performance, and can be applied to not only binary-domain & binary-class classification, but also multi-domain & multi-class classification tasks.
Table 2. Classification performance of the ablation study of the proposed model using F1-score with ± std on three datasets. w/o $L_s$ and w/o $L_t$ indicate training without $L_{spatial}$ and $L_{texture}$, respectively. Replace $L_s$, Replace $L_t$, and Replace $L_s$ & $L_t$ indicate replacing with perceptual loss [9], style loss [26], and perceptual & style loss for each loss function.

| Method       | (a) Five vs. Six | (b) Dogs vs. Cats | (c) COV. vs. Bact. |
|--------------|-----------------|-------------------|-------------------|
| w/o $L_t$   | 63.88 ± 4.97    | 69.92 ± 6.50      | 52.68 ± 4.80      |
| w/o $L_s$   | 3.57 ± 1.93     | 88.28 ± 0.98      | 48.19 ± 5.92      |
| Re. $L_s$   | 56.87 ± 1.20    | 89.45 ± 0.92      | 52.08 ± 10.04     |
| Re. $L_t$   | 13.16 ± 1.64    | 87.07 ± 1.61      | 42.37 ± 5.21      |
| Re. $L_s$ & $L_t$ | 19.98 ± 10.37  | 86.95 ± 0.93      | 48.59 ± 5.58      |
| Proposed    | 75.18 ± 3.52    | 91.80 ± 0.44      | 76.50 ± 3.58      |

Ablation Study: We performed ablation studies to evaluate the impact of the content $L_{spatial}$ and texture $L_{texture}$ losses by either removing one or replacing them with a different loss function. It is essential to validate whether texture co-occurrence and self-similarity are the key components for improving image generation quality in biased settings. Thus, we used style [26] and perceptual [9] losses to replace $L_{spatial}$ and $L_{texture}$, respectively, as they are the most common techniques used in style transfer methods to retain content features for image generation. On the other hand, the adversarial $L_{GAN}$ loss is an essential part of our image generation pipeline and unrealistic results were produced when omitted, thus we include this loss in our ablations.

Results on the Five vs. Six dataset in Table 2(a) show that the use of texture co-occurrence is a key in enabling the classifier to correctly differentiate digits. However, qualitative results in Fig. 7 indicate that the absence of $L_{texture}$ results in a failure to change the texture and correctly mitigate bias information. Likewise, replacing $L_{texture}$ with a style loss overcomes the content constraints by modifying both texture and shape. In addition, while the absence of $L_{spatial}$ lets the classifier achieve higher performance than most methods, qualitative results show that even though the texture is correct, the shape does not resemble neither a number five nor six, which can be exploited by the binary classifier and can lead to incorrect predictions. Moreover, replacing $L_{spatial}$ or $L_{spatial}$ & $L_{texture}$ results in poor image quality with ambiguous shapes. Consequently, we show that the combination of texture co-occurrence and self-similarity losses leads to higher image quality towards mitigating the underlying texture bias.

For Dogs vs. Cats, the model seems to be less susceptible to a drop in performance due to a change in the loss term, and usually generates good quality images as is reflected by the good classification results. The only exception is the absence of $L_{spatial}$, which reported a lower score (Table 2(b)), and it is clear from Fig. 7 that the shape is lost in a similar fashion to the results observed in Five vs. Six data. In contrast, COVID-19 vs. Bacterial pneumonia results show high dependency on the loss functions leading to decrease in performance across all ablation studies. We believe this due to the explicit constraints of content and texture, as they are important to reduce the risk of unintended distortion, and insufficient texture translation, respectively.
In summary, the ablation studies show that our loss function design choices played an essential role in alleviating the inherited texture bias problem observed in training data, and proved that our method could mitigate texture bias across multiple datasets with high performance by adopting an image generation strategy. Our ablation studies also demonstrate superior performances shown by the experiments with replacement as they have the same settings with previous style transfer methods [9,26]. Our method not only achieved higher quantitative performance against style transfer competitors, but also showed better visual quality (Fig. 7).

**Discussion:** We analyzed the ability of image translation models to mitigate biases on multiple datasets. Our choice to generate additional training images by transferring texture between sets of images that have different classes proved an effective solution to train robust classifiers. Moreover, the ablation studies support the choice of spatial and texture based losses to correctly translate the texture of images without modifying their content, leading to high debiasing performance. This is a vital requirement for most medical and industrial domains that have strict structural constraints on their images, and cannot solely rely on web or social media to collect a large number of data samples. Usually, data collection is often limited and biased to some local settings (e.g., acquisition equipment or protocols), where the labels might be aligned with features not related to the classes intrinsic properties. Notably, we empirically demonstrate the effectiveness of our method to solve a real world scenario by mitigating biases induced by acquisition protocols in the COVID-19 vs. Bacterial pneumonia experiments.

### 6 Conclusion

In this work, we have proposed a novel strategy to augment an initially texture biased dataset with new instances that mitigate biased properties using a generative model. We have demonstrated that a dataset extended by our proposed method can be used to train a debiased classifier, and the strategy of transferring texture while retaining content information is a valid choice for addressing data bias in multiple types of datasets. In particular, our model was able to handle simple biases such as color, as well as more complex and realistic texture features i.e., induced by different CT scanning protocols. Our results show that using both texture co-occurrence and spatial self-similarity losses to impose constraints in our generative model is key, as the losses are complementary. Finally, we report high accuracy with significant margins across all evaluated datasets, and generated higher quality images compared to prior state-of-the-art methods.

### References

1. Aerts, H.J., Velazquez, E.R., Leijenaar, R.T., Parmar, C., Grossmann, P., Carvalho, S., Bussink, J., Monshouwer, R., Haibe-Kains, B., Rietveld, D., et al.: Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach. Nature communications 5(1), 1–9 (2014)
2. Alvi, M., Zisserman, A., Nelläker, C.: Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. In: Proceedings of the European Conference on Computer Vision Workshops (2018)

3. Barbu, A., Mayo, D., Alverio, J., Luo, W., Wang, C., Gutfreund, D., Tenenbaum, J., Katz, B.: Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In: Advances in Neural Information Processing Systems. vol. 32 (2019)

4. Bissoto, A., Valle, E., Avila, S.: Debiasing skin lesion datasets and models? not so fast. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 740–741 (2020)

5. Choi, K., Grover, A., Singh, T., Shu, R., Ermon, S.: Fair generative modeling via weak supervision. In: International Conference on Machine Learning. pp. 1887–1898. PMLR (2020)

6. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)

7. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: Proceedings of the 32nd International Conference on Machine Learning. vol. 37, pp. 1180–1189 (2015)

8. Gatys, L., Ecker, A.S., Bethge, M.: Texture synthesis using convolutional neural networks. Advances in neural information processing systems 28, 262–270 (2015)

9. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2414–2423 (2016)

10. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In: International Conference on Learning Representations (2019)

11. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems 27 (2014)

12. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)

13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

14. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1501–1510 (2017)

15. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)

16. Julesz, B.: Visual pattern discrimination. IRE transactions on Information Theory 8(2), 84–92 (1962)

17. Julesz, B.: Textons, the elements of texture perception, and their interactions. Nature 290(5802), 91–97 (1981)

18. Jun, M., Cheng, G., Yixin, W., Xingle, A., Jiantao, G., Ziqi, Y., Minqing, Z., Xin, L., Xueyuan, D., Shucheng, C., et al.: Covid-19 ct lung and infection segmentation dataset. Zenodo, Apr 20 (2020)
19. Kalra, M.K., Homayounieh, F., Arru, C., Holmberg, O., Vassileva, J.: Chest CT practice and protocols for COVID-19 from radiation dose management perspective. European Radiology pp. 1–7 (2020)
20. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8110–8119 (2020)
21. Kim, B., Kim, H., Kim, K., Kim, S., Kim, J.: Learning not to learn: Training deep neural networks with biased data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9012–9020 (2019)
22. Kolkin, N., Salavon, J., Shakhnarovich, G.: Style transfer by relaxed optimal transport and self-similarity. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
23. LeCun, Y., Cortes, C., Burges, C.: Mnist handwritten digit database. ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist (2010)
24. Lee, H.Y., Tseng, H.Y., Mao, Q., Huang, J.B., Lu, Y.D., Singh, M.K., Yang, M.H.: Drit+: Diverse image-to-image translation via disentangled representations. International Journal of Computer Vision pp. 1–16 (2020)
25. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Deeper, broader and artier domain generalization. In: Proceedings of the IEEE international conference on computer vision. pp. 5542–5550 (2017)
26. Li, Y., Wang, N., Liu, J., Hou, X.: Demystifying neural style transfer. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. p. 2230–2236. IJCAI’17 (2017)
27. Li, Y., Vasconcelos, N.: Repair: Removing representation bias by dataset resampling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
28. Li, Y., Li, Y., Vasconcelos, N.: Resound: Towards action recognition without representation bias. In: Proceedings of the European Conference on Computer Vision (ECCV) (September 2018)
29. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30 (2017)
30. Louppe, G., Kagan, M., Cranmer, K.: Learning to pivot with adversarial networks. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30 (2017)
31. Morozov, S., Andreychenko, A., Pavlov, N., Vladymyrskyy, A., Ledikhova, N., Gombolevskiy, V., Blokhin, I.A., Gelezhe, P., Gonchar, A., Chernina, V.Y.: Mosmeddata: Chest CT scans with COVID-19 related findings dataset. arXiv preprint arXiv:2005.06465 (2020)
32. Nam, H., Lee, H., Park, J., Yoon, W., Yoo, D.: Reducing domain gap by reducing style bias. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8690–8699 (June 2021)
33. Nuriel, O., Benaim, S., Wolf, L.: Permutated adain: Reducing the bias towards global statistics in image classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9482–9491 (2021)
34. Park, T., Efros, A.A., Zhang, R., Zhu, J.Y.: Contrastive learning for unpaired image-to-image translation. In: European Conference on Computer Vision (2020)
35. Park, T., Zhu, J.Y., Wang, O., Lu, J., Shechtman, E., Efros, A., Zhang, R.: Swapping autoencoder for deep image manipulation. In: Advances in Neural Information Processing Systems. vol. 33, pp. 7198–7211 (2020)
36. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems 32, 8026–8037 (2019)
37. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1406–1415 (2019)
38. Pontone, G., Scafuri, S., Mancini, M.E., Agalbato, C., Guglielmo, M., Baggiano, A., Muscogiuri, G., Fusini, L., Andreini, D., Mushtaq, S., et al.: Role of computed tomography in covid-19. Journal of cardiovascular computed tomography (2020)
39. Sattigeri, P., Hoffman, S.C., Chentharmarshkan, V., Varshney, K.R.: Fairness gan: Generating datasets with fairness properties using a generative adversarial network. IBM Journal of Research and Development 63(4/5), 3–1 (2019)
40. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2014)
41. Simpson, A.L., Antonelli, M., Bakas, S., Bilello, M., Farahani, K., Van Ginneken, B., Kopp-Schneider, A., Landman, B.A., Litjens, G., Menze, B., et al.: A large annotated medical image dataset for the development and evaluation of segmentation algorithms. arXiv preprint arXiv:1902.09063 (2019)
42. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (December 2015)
43. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5018–5027 (2017)
44. Wang, H., He, Z., Lipton, Z.L., Xing, E.P.: Learning robust representations by projecting superficial statistics out. In: International Conference on Learning Representations (2019)
45. Wang, T., Zhao, J., Yatskar, M., Chang, K.W., Ordonez, V.: Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)
46. Zhang, B.H., Lemoine, B., Mitchell, M.: Mitigating unwanted biases with adversarial learning. In: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society. pp. 335–340 (2018)
47. Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Zhang, Z., Lin, H., Sun, Y., He, T., Muller, J., Mannath, R., Li, M., Smola, A.: Resnest: Split-attention networks. arXiv preprint arXiv:2004.08955 (2020)
48. Zheng, C., Cham, T.J., Cai, J.: The spatially-correlative loss for various image translation tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16407–16417 (June 2021)
49. Zhou, K., Liu, Z., Qiao, Y., Xiang, T., Loy, C.C.: Domain generalization: A survey. arXiv preprint arXiv:2103.02503 (2021)
50. Zhou, K., Yang, Y., Qiao, Y., Xiang, T.: Domain generalization with mixstyle. In: International Conference on Learning Representations (2021)
51. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017)
S.1 Additional Qualitative Results

We show additional results on three datasets. We compare our method with the baseline generative methods: CycleGAN [51], UNIT [29], MUNIT [15], DRIT++ [24], and CUT [34]. Figures S.1, S.2, and S.3 show images generated by pairs of images in each of two rows. Since our proposed method creates an image using the contents and texture information of the pair image, only the texture is transformed like the target image while maintaining the contents of the source image. On the other hand, since other methods transform an image in one domain into textures of other domain images, the image is transformed by considering various texture characteristics of the different domain, and thus texture conversion between image pairs cannot be performed.

Fig. S.1. Qualitative results of Five vs. Six dataset on five generative models and ours.
Fig. S.2. Qualitative results of Dogs vs. Cats dataset on five generative models and ours.
Fig. S.3. Qualitative results of COVID-19 vs. Bacterial pneumonia dataset on five generative models and ours.
S.2 Additional Qualitative Results on Multi-class Biased Datasets

We show additional results on multi-class biased datasets. Figures S.4 and S.5 show generated images of our proposed method.

Fig. S.4. Qualitative results of Digit dataset on our method.

Fig. S.5. Qualitative results of Biased PACS dataset on our method.
S.3 Additional Dataset Details

The bias information and dataset statistics of Five vs. Six, Dogs vs. Cats, and COVID-19 vs. Bacterial pneumonia datasets are shown in Table S.1. The bias information and dataset statistics of Digit and Biased PACS datasets are shown in Table S.2. For COVID-19 vs. Bacterial pneumonia dataset, the CT scans were collected at anonymous medical center from multiple scanners (e.g. Siemens SOMATOM Definition AS+). Window width normalization was applied between 1500 and -600 HU, and the lesions were extracted by applying a Mask-cascade-RCNN-ResNet-200 [47] with deformable convolution neural network (DCN) [6] trained on [1183141].

Table S.1. The bias information and statistics of the train/validation/test of Five vs. Six, Dogs vs. Cats, and COVID-19 vs. Bacterial pneumonia datasets. For COVID-19 vs. Bacterial pneumonia, the number of slices and patients are listed.

| Split | Five vs. Six | Dogs vs. Cats | COVID-19 vs. Bact. |
|-------|-------------|---------------|-------------------|
|       | Five | Six | Dogs | Cats | COVID-19 | Bacterial |
| Train | MNIST-M (4,798) | MNIST (5,327) | Bright (4,770) | Dark (4,274) | Non-Cont. (2,119 (46)) | Cont. (2,766 (61)) |
| Val   | MNIST-M (533) | MNIST (591) | Bright (529) | Dark (474) | Non-Cont. (206 (5)) | Cont. (268 (6)) |
| Test  | MNIST (5,421) | MNIST-M (5,812) | Dark (3,652) | Bright (3,086) | Cont. (482 (8)) | Non-Cont. (326 (8)) |

Table S.2. The bias information and statistics of the train/validation/test of Digit and Biased PACS datasets.

| Split | Five vs. Six | Biased PACS |
|-------|-------------|-------------|
|       | One ∼ Four | Five ∼ Nine | Dog | Elephant | Giraffe | Photo | Art |
| Train | MNIST-M (27,085) | MNIST (26,466) | Cartoon (351) | Sketch (666) | Photo (164) | Art (181) |
| Val   | MNIST-M (3,007) | MNIST (2,938) | Cartoon (38) | Sketch (74) | Photo (18) | Art (20) |
| Test  | MNIST (30,596) | MNIST-M (28,909) | Sketch (772) | Photo (202) | Art (285) | Cartoon (324) |
|       |       |            | Cartoon (379) | Art (457) | Sketch (753) | Cartoon (816) |
|       |       |            | Photo (199) |  | Photo (199) |  |
S.4 Additional Experiment Details

The classifier was trained for 100 epochs with batch size 64, and the learning rate was 0.01. Image size was $224 \times 224$ for Dogs vs. Cats, COVID-19 vs. Bacterial pneumonia, Biased PACS, and Inverse Biased PACS datasets, and $32 \times 32$ for Five vs. Six and Digit datasets. We performed data augmentation by applying random-affine translation (rotation, translation, and scale), horizontal flips, and color augmentations (i.e., brightness, contrast, saturation, and hue). For domain generalization models, the best hyper-parameters reported in their paper were used for training (e.g., alpha in [53] and probability in [33]), other settings were left the same as the baseline classifier. For generative models, all experiments were trained for 200 epochs approximately and batch size was set to 8, however, UNIT and MUNIT were set to 4 because of hardware (CUDA 9) compatibility. In the classification step, we applied random affine translation (rotation, translation, and scale) and horizontal flips to Dogs vs. Cats, COVID-19 vs. Bacterial pneumonia, Biased PACS, and Inverse Biased PACS datasets, but no augmentation was applied to Five vs. Six and Digit datasets. The rest of settings followed the same as the baseline classifier.

S.5 Additional Implementation Details

For baseline [13], we employed the official Pytorch [36] ImageNet code\(^3\) for training. To run the experiments in same settings, we integrated all non-generative debiasing implementations to the baseline code. To integrate non-generative debiasing models, we prioritized using the author’s implementation. In case when the author’s implementation is not available, we employed a publicly available code as an alternative. For Learning-to-pivot [30], we used a publicly available code\(^4\). For Adv debias [46], we used a publicly available code\(^5\). For BlindEye [2], we used the author’s implementation of confusion loss\(^6\). For Learning-not-to-learn [21], we used the author’s implementation\(^7\). For Not Enough [45], we used the author’s implementation\(^8\). For domain generalization methods, all author’s implementations are available. For Permuted AdaIN [33], we used the author’s implementation\(^9\). For MixStyle [50], we used the author’s implementation\(^10\). To train generative models, all author’s implementations are available. For CycleGAN [51], we used the author’s implementation\(^11\). For UNIT [29] and MUNIT

\(^3\) https://github.com/pytorch/examples/blob/master/imagenet/main.py
\(^4\) https://github.com/equalgo/fairness-in-ml
\(^5\) https://github.com/Trusted-AI/AIF360
\(^6\) https://github.com/mahfujau/domain_adaptation_iccv15
\(^7\) https://github.com/feidfoe/learning-not-to-learn
\(^8\) https://github.com/uvavision/Balanced-Datasets-Are-Not-Enough
\(^9\) https://github.com/onuriel/PermutedAdaIN
\(^10\) https://github.com/KaiyangZhou/mixstyle-release
\(^11\) https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix
we used the author’s implementation. For DRIT++, we used the author’s implementation. For CUT, we used the author’s implementation.