Improving Domain Generalization using Style Regularization

Gustavo Pérez
University of Massachusetts Amherst
gperezsarabi@umass.edu

Abstract

We study the problem of improving domain generalization on deep networks by reducing the bias towards texture learned by these models when pre-trained in large color image datasets like ImageNet. To do so, we present a style regularization to enforce more shape-biased learning. Also, we propose an experimental setup using synthetically created test sets using state-of-the-art style transfer methods. We report our experiments on stylized versions of CIFAR-10 and STL-10 datasets. In our preliminary results presented here, we show that our style regularization improves performance on new domains but not as significantly as with style augmentation.

1. Introduction

Despite shape appearing to be more important than size and texture to categorize objects for children and adults [13], ImageNet pre-trained deep networks are biased toward texture [7] which might explain why a model trained in a source domain performs poorly in a target domain, even if the domain shift is not large. Texture information is usually domain-specific and shape can be more important than size and texture to categorize objects.

Improving network generalization by reducing bias toward texture is already being investigated. In [7], the authors show that ImageNet pre-trained networks learn texture-biased feature extractors and that it is possible to learn shape-biased representations using stylized images providing improvement in object detection and robustness towards image distortions. Following this work, many [11, 3, 17] have explored different ways of using style augmentation to successfully improve domain generalization.

Although style augmentation has proven useful, it needs a longer training time to converge than most data augmentation techniques, but also requires additional pre-trained deep models to generate the stylized images and in most cases additional datasets to extract the styles from. We explore techniques to learn transferable representations to more distant domains without the need for style augmentation. In specific, we study a way to learn these unbiased representations by means of a training regularization. We hypothesize that style augmentation reduces the spectral norm of the Gram matrix built with the feature activations in a deep network layer. In this work, we propose an experimental framework using random styles generated as in [11]. We also experiment in CIFAR-10 [12] and STL-10 [4] datasets using LeNet [14] for CIFAR-10 and Inception_v3 [16] for STL-10. We also present experiments varying the number of convolutional layers included in the regularization. In these preliminary results, we find that our regularization improves domain generalization in our experimental setting but it is still far from achieving results using style augmentation. In the discussion section, we consider flaws in our experimental setup that might cause this gap between our results and style augmentation and discuss future work.

2. Related work

Domain generalization In [9], the authors propose a benchmark to study domain generalization techniques including seven multi-domain datasets, nine baselines algorithms, and three model selection criteria. In [7], the authors show that ImageNet pre-trained networks learn texture-biased feature extractors and that it is possible to learn shape-biased representations using stylized images (style transfer) providing improvement in object detection and robustness towards image distortions. Concurrently with [9], [3] study the advantages of style augmentation in domain generalization. They use AdaIN [10] to perform style transfer between images of a set of source datasets during training. In more related work, the authors in [11] propose to use data augmentation with style randomization from a multivariate normal distribution to improve the generalization of CNNs. Also, they argue that domain bias is a form of over-fitting and it can be reduced using style augmentation (not as a domain adaptation technique but it can reduce the need for it). Similarly to [3] and [11], [17] also investigates the use of style transfer as data augmentation to improve domain generalization. Finally, [2] learns a regularization function using the variability in the source domains following a meta-learning framework.

Figure 1. **Selected styles. Left.** Training styles selected manually from the pool of 200 styles chosen by maximizing the fractional distance (p = 0.3) between them from the N generated styles with [11]. **Right.** Testing styles. Mug picture taken from *https://github.com/philipjackson/style-augmentation*.

Style transfer In [6], the authors show that learned feature representations from a high-performing CNN can be used to independently process content (activations of the top layers) and style (correlations between filter responses) of an image by matching the Gram matrices of their feature maps. In [15], the authors theoretically show that matching these Gram matrices is equivalent to minimize the maximum Mean Discrepancy (MMD) with the second-order polynomial kernel and experiment with several other distribution alignment methods. [5] demonstrates that the manipulation of the normalization parameters in each unit's activation was sufficient to train a single style transfer network across 32 varied painting styles. In [8], the authors present a method trained with a large number of paintings that allows real-time stylization and generalizes to previously unobserved styles.

3. Style regularization

We hypothesize that same as an L2 regularization keeps low parameter values, the style augmentation would reduce the spectral norm of the covariance matrix of the feature activations in a deep network layer. Intuitively, the regularization should reduce the norm of the covariance matrix of the feature activations, hence it would tend to the identity matrix (i.e. minimizing the correlation between feature activations). Formally, given a feature map tensor $A \in \mathbb{R}^{C\times H\times W}$, its Gram matrix $G \in \mathbb{R}^{C\times C}$ is produced by $G = \hat{A}\hat{A}^T$, where $\hat{A} \in \mathbb{R}^{C\times HW}$ is a reshaped version of the feature activations. Our regularization for a batch of N images is given by:

$$R = \alpha \sum_{i}^{N} ||G_i||_p \tag{1}$$

where p is the order of the norm and α is a scaling factor. In our experiments, we get best results using the spectral norm (p = 2) and a scaling factor of α = 1E-2.

4. Experiments

We first describe our experimental setup with the datasets and methods used in our experiments. Then, we present our results. We evaluate our results using the per-image accuracy which are reported in Tables 1 and 2.

4.1. Experimental setup

For our experiments, we use CIFAR-10 [12] and STL-10 [4] datasets with the train and test splits and classes as proposed in each of the datasets. In addition, we propose stylized train and test sets using the style transfer method proposed in [11]. To do so, we sample randomly $N >> 200$ styles. Because of the high dimensionality of the vectors required by [11] to produce each new style, we use a fractional distance as proposed in [1] to select 200 "far-away" styles for each train/test subset. Finally, we hand-pick 20 styles from the train set that differ the most from the 20 other styles of the test set (trying to guarantee that the styles seen during training are sufficiently different from those in testing). In Figure 1, we show the selected styles for training and testing.

During training, we use four different configurations for the two datasets: no augmentation or regularization, traditional augmentation, style augmentation, and style regularization. For traditional augmentation, we use random crops, random grayscale, color jitter, random horizontal flips, random rotations of up to 20 degrees, random erasing, and random shear. For style augmentation, we use the method pro-

posed in [11]. For style regularization we use the method proposed in Section 3, varying the number of convolutional layers used in the regularization and the scaling factor of the regularization during training. In the case of the STL-10 dataset, we get the best results using a scaling factor of 1E-2 and applying the regularization to the first 5 convolutional layers and the first 3 inception modules (i.e. to all convolutional layers of each module) of the inception_v3. For evaluation, we report the overall accuracy on three different sets: no style modification (no-S), test stylization (test-S), and train stylization (train-S). no-S is used as an oracle performance and to see how much the introduced methods decrease the accuracy when testing on non-stylized images. train-S and test-S are used to differentiate when style augmentation is used during training.

4.2. Results and discussion

Color jitter behaves similar to style augmentation when images are too small. We perform experiments on CIFAR-10 [12] following the experimental framework described in Section 4.1. When image size is too small, using color jitter augmentation with high jittering values of hue, saturation, and contrast increase performance on stylized test sets comparatively with style augmentation performance. Thus, this suggests that style changes on small images discard texture changes, resulting in only color modifications, which might explain why our style regularization does not improve performance in CIFAR-10. In Table 1, we present results on CIFAR-10 dataset using traditional augmentation (Trad.), style augmentation (Style), style regularization (Reg.), and only color jitter augmentation (bottom row) with increased jittering values. As shown in the last row of Table 1, using color jitter augmentation only with high hue, saturation, and contrast values improves performance on the stylized test images comparatively with using only style augmentation performance (compare second row with bottom row in Table 1). In addition, we can see that other traditional data augmentation techniques are not doing much compared to increased color jitter only (compare first row with bottom row in Table 1).

Experiments on STL-10. Style augmentation improves performance on stylized test sets (compare second and bottom rows in Table 2) but not significantly enough compared to style augmentation (compare third and bottom rows in Table 2). We suspect that one reason for the large difference in performance between style regularization and style augmentation might be that we are not capturing different enough styles between test and training sets, which causes the model to overfit to the testing styles when using style augmentation during training. A reason for us to believe this is that the difference between test-S and train-S results is not large enough. Finally, when improving performance in stylized test sets, style augmentation usually reduces performance on the original STL-10 test set (no-S). However, even though the increase in stylized test sets is not as big as with style augmentation, using style regularization does not harm performance in the original STL-10 test set.

Data augmentation			Accuracy (%)		
Trad.	Style	Reg.	no-S	test-S	train-S
✓			66.2	21.9	21.5
✓	✓		**67.6**	**50.1**	**49.2**
	✓		63.3	39.7	39.8
✓		✓	65.9	20.8	21.1
○			66.5	35.9	35.8

○**:** Color jitter only using high hue, contrast, and saturation values.

no-S: No style transfer on test set.

train-S: Applying train styles transfer on test set.

test-S: Applying test styles transfer on test set.

Table 1. **Results on CIFAR-10.** Accuracy (%) on CIFAR-10 test set using style transfer (see figure footnote). Training with and without traditional (trad.), style augmentation, and style regularization. Traditional augmentation refers to random crop, grayscale, color jitter, horizontal flip, rotation, and shear.

Data augmentation			Accuracy (%)		
Trad.	Style	Reg.	no-S	test-S	train-S
			62.8	15.3	14.0
✓			65.4	20.6	20.2
	✓		63.2	41.0	45.9
✓	✓		63.3	**58.2**	**58.6**
✓		spe.	**65.9**	23.6	25.2

no-S: No style transfer on test set.

train-S: Applying train styles transfer on test set.

test-S: Applying test styles transfer on test set.

* increased scaling value.

Table 2. **Results on STL-10.** Accuracy (%) on STL-10 using style transfer (see figure footnote). Training with and without traditional (trad.) and style augmentation, and regularization (reg.).

5. Conclusion and future work

In this work, we proposed a style regularization to reduce texture-biased representation learning and improve domain generalization on deep networks. In addition, we proposed a benchmark of stylized test sets to emulate domain shifts between subsets. For future work, we will explore different experimental setups like the ones used in [3, 17], and study different configurations of our style regularization (e.g. reducing the nuclear norm instead of the spectral).

References

[1] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *Proceedings of the 8th International Conference on Database Theory*, ICDT '01, page 420–434, Berlin, Heidelberg, 2001. Springer-Verlag.

[2] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 998–1008. Curran Associates, Inc., 2018.

[3] Francesco Cappio Borlino, Antonio D'Innocente, and Tatiana Tommasi. Rethinking domain generalization baselines, 2021.

[4] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011.

[5] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *CoRR*, abs/1610.07629, 2016.

[6] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[7] 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*. OpenReview.net, 2019.

[8] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *CoRR*, abs/1705.06830, 2017.

[9] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization, 2020.

[10] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization, 2017.

[11] Philip T. G. Jackson, Amir Atapour Abarghouei, Stephen Bonner, Toby P. Breckon, and Boguslaw Obara. Style augmentation: Data augmentation via style randomization. *CoRR*, abs/1809.05375, 2018.

[12] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[13] B. Landau, L. B. Smith, and S. S. Jones. The importance of shape in early lexical learning. *Cognitive Development*, 3:299–321, 1988.

[14] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551, 1989.

[15] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. *CoRR*, abs/1701.01036, 2017.

[16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015.

[17] Y. Zhang, Y. Zhang, Q. Xu, and R. Zhang. Learning robust shape-based features for domain generalization. *IEEE Access*, 8:63748–63756, 2020.