Towards Shape-regularized Learning for Mitigating Texture Bias in CNNs

Harsh Sinha
harsh.sinha@pitt.edu
University of Pittsburgh
Pittsburgh, PA, USA

Adriana Kovashka
kovashka@cs.pitt.edu
University of Pittsburgh
Pittsburgh, PA, USA

ABSTRACT

CNNs have emerged as powerful techniques for object recognition. However, the test performance of CNNs is contingent on the similarity to training distribution. Existing methods focus on data augmentation to address out-of-domain generalization. In contrast, we enforce a shape bias by encouraging our model to learn features that correlate with those learned from the shape of the object. We show that explicit shape cues enable CNNs to learn features that are robust to unseen image manipulations *i.e*. novel textures with the same semantic content. Our models are validated on Toys4K dataset which consists of 4179 3D objects and image pairs. To quantify texture bias, we synthesize dataset variants called Style (style-transfer with GANs), CueConflict (conflicting texture & semantics), and Scrambled datasets (obfuscating semantics by scrambling pixel blocks). Our experiments show that the benefits of using shape is not subject to specific shape representations like point clouds, rather the same benefits can be obtained from a simpler representation such as the distance transform.

CCS CONCEPTS

• **Computing methodologies** → **Learning settings**; **Regularization**; *Supervised learning*.

KEYWORDS

shape bias, shape representation, out-of-domain robustness

ACM Reference Format:
Harsh Sinha and Adriana Kovashka. 2023. Towards Shape-regularized Learning for Mitigating Texture Bias in CNNs. In *International Conference on Multimedia Retrieval (ICMR '23), June 12–15, 2023, Thessaloniki, Greece.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3591106.3592231

1 INTRODUCTION

Recognition in humans is perceived as a composition of structure and abstraction [30]. From a distance, it is easier to recognize the *structure* as an animal, rather directly discerning differences as a horse or a zebra. Similarly, if a child sees a green elephant, they may still recognize it as an elephant by *abstracting* the object of its immediate atypical attributes. Thus, the role of structure and abstraction is fundamental to object recognition [1]. In fact, classical methods of object recognition utilized structure and abstraction by incorporating the *shape* of the object [6, 20, 37]. Studies of cognitive psychology [23, 31] suggest that shape cues play a dominant role over color and texture for inference in category membership.

CNNs [22, 41] have achieved impressive performance across several challenging tasks by composing non-linear functions capturing appearance information, without using the shape of the object explicitly. In earlier works [21, 24], the gains in performance have been attributed to the ability of deeper layers to learn complex representations of shape. However, recent studies [12] exhibit CNN's failure in presence of adversarial perturbations. This contradicts the earlier shape hypothesis: If deeper layers were able to learn structure of the object, CNNs would not have been susceptible to these minor perturbations. Geirhos *et al*. [10] suggest that CNN's impressive success builds on taking *shortcuts*, instead of arriving at the intended solution. These shortcuts allow CNNs to achieve high gains on the training data, often leading to surprising outcomes and unexpected results for out-of-domain generalization. Related studies show that CNNs have high texture bias and low shape bias [16, 19]. Given the wide acceptance of CNNs, the problem of intentional perturbations is crucial, especially when CNNs are applied to safety-critical real-world problems.

In this paper, we advocate that learning the shape of an object provides a more stable representation compared to purely CNN-based ones. We argue that shape is the inherent invariant for categorization across all domains. Fig. 1 presents this idea visually. Humans can recognize the object despite the difference in texture and appearance for *Original*, *Style*, & *CueConflict* images. But such robust shape-based recognition is not as easy for machine learning algorithms [12]. We refer to the consistency of shape across the variations as a *shape-invariant*, and propose explicit shape cues to find an *invariant* representation, which remains constant even after diverse transformations of the object.

Briefly, our method works as follows. By using contrastive loss between the input image and its corresponding shape, we enforce shape-bias combining the benefits of both—learning from the appearance of input images and their corresponding shapes. The framework utilizes shape explicitly during training to learn better image representations. However, shape data is not required for inference. In order to utilize shape efficiently, it is important that shape representation captures salient information such as gradients and surface contours. These attributes are captured by point clouds (PC), mesh and voxel grids, however the availability of 3D data is expensive and often not guaranteed. We introduce a simpler notion of shape, a distance transform (DT) which successfully captures local perceptual information (as shown in Fig. 2) bypassing any complex interface

ICMR '23, June 12–15, 2023, Thessaloniki, Greece

ACM ISBN 979-8-4007-0178-8/23/06…$15.00
https://doi.org/10.1145/3591106.3592231

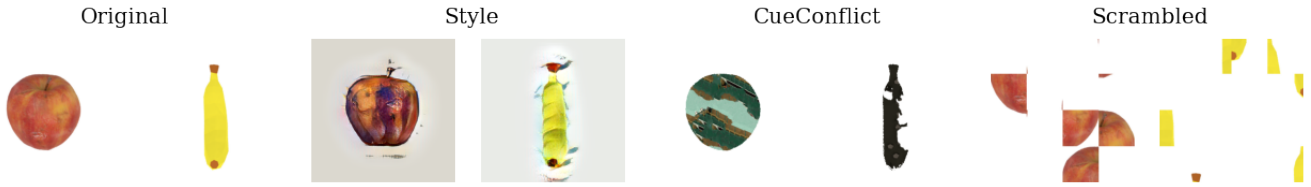

Figure 1: The dataset variations we use for evaluation. We expect a model robust to texture bias to perform well on Style and CueConflict, but poorly on Scrambled.

and costly sensors. We show both PC and DT obtain robust image representations for out-of-domain generalization.

We demonstrate the advantage of proposed explicit shape bias over methods that capture shape implicitly through data augmentations. We test across different machine learning tasks (*Unseen images*, *Unseen instances* and *Unseen categories*) taking into account supervised and unsupervised scenarios in the Toys4K dataset [40] (Fig. 4) without necessitating any retraining on new dataset distributions such as Style or CueConflict. In summary, this paper makes the following contributions:

- By using contrastive loss between the input image and its corresponding shape, we enforce a shape bias to learn better image representations, robust to unseen image manipulations.
- We break the correlation between objects' shape and their natural texture by synthesizing variants of the original Toys4K dataset, and evaluate the susceptibility of a model to local texture by evaluating on Scrambled images.
- We show the efficacy of the explicit shape bias across different un-/supervised tasks without need of retraining on new dataset distributions.
- We demonstrate the advantages of a 2D distance transform to capture shape and show that both this simpler shape representation, and 3D point clouds, are useful to inject shape bias.

2 RELATED WORK

Generalization CNNs have been shown to be over-reliant on textural information in images [28]. Therefore, CNNs can classify textures even when object structure is absent [9]. Thus, the impressive results on ImageNet is attributed to texture cues rather than learning shape of the object [16]. Naturally, even the most trivial (often intentional) perturbations in texture degrades the performance of CNNs drastically [12]. Over the past years, there have been several data augmentation techniques such Stylized-ImageNet [11], Mixup [51], CutMix [50], AugMix [15] which address this discrepancy in performance. Mummadi *et al*. [32] suggests that the improvement on corruption-robustness by data augmentation, may not always lead to shape-bias. Our work has similar motivation as in domain generalization [5, 18, 25, 44] which guide CNNs to be robust against novel domains. In contrast to data augmentation & domain generalization methods, we use an explicit shape bias as regularization for learning robust representations. We aim for a *shape-invariant* model which uses shape-cues explicitly such that the learned representation is consistent across domain boundaries.

Shape Extraction Early research in shape modeling extract *shapes* as closed boundary edges which have been studied extensively using Fourier descriptors [2, 27], medial axis transform [3], skeletons [8], bone graphs [29], shock graphs [4, 39], AMAT [42], and flux graphs [36, 47]. Skeletons are imperfect as they often contain spurious branches that do not correspond to actual parts of the object. Narayanan *et al*. [33] used shock graphs with GNNs for graph-based shape transformation of input image to replace standard CNNs. On the contrary, we investigate if shape cues can be used to enhance classical CNNs. Appearance representations are ubiquitous in computer vision, and therefore we deliberately address alignment of image-to-shape feature representations via contrastive learning. However, analyzing the efficacy of different shape extraction techniques is outside the scope of this work.

Shape-Texture Bias Recent works have addressed texture-bias in CNNs by training on *uninformative-styled* datasets [11, 32] (*i.e*. breaking correlation between identity and natural texture of the object), mixing feature statistics between image instances [52], using gradient information [49] and by aliasing [43]. Li *et al*. [26] show that augmenting the dataset with conflicting shape and texture cues can improve CNN performance. In this paper, we focus on utilizing shape to find an invariant representation rather than discarding texture-cues. Recently, Stojanov *et al*. [40] used shape to improve generalization to novel object categories *in the same source domain*. Inspired from this work, we refine and expand the advantages of using shape for generalization to *unseen image manipulations (novel domains*) over a variety of tasks in machine learning (Fig. 4). We use contrastive learning to achieve shape-regularized feature representations in order to alleviate the texture bias in CNNs. We show that shape invariance can be derived by using a 2D distance transform rather than the more costly 3D point cloud data used by Stojanov *et al*. [40].

3 USING SHAPE FOR CLASSIFICATION

Our work focuses on utilizing shape invariance to alleviate texture bias in CNNs. Shape is interesting especially due to it robustness

Figure 2: Visualization of distance transform (DT) computed for original objects in the Toys4K dataset (images shown first, then DTs). The distance transform captures the skeleton and boundary of an image in a compact representation.

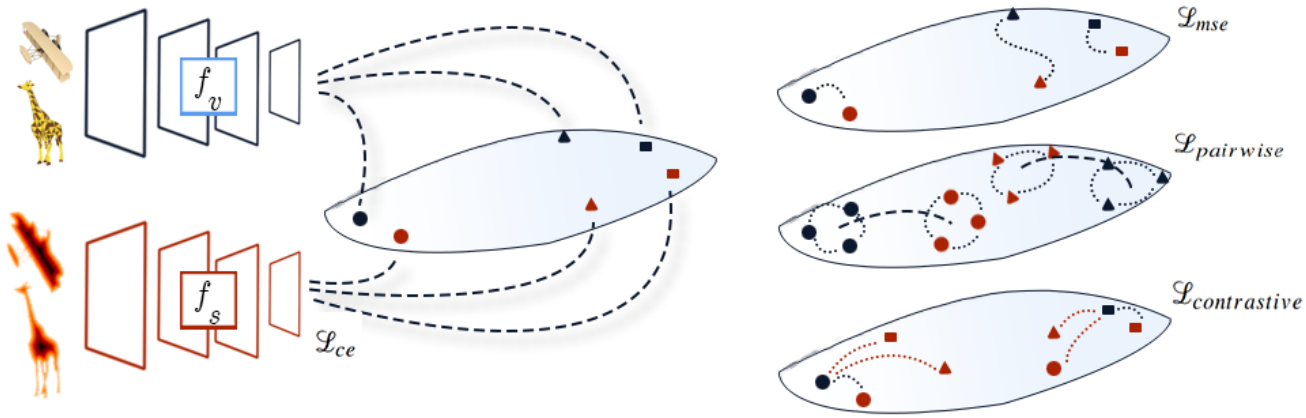

Figure 3: (Left) ResNet18 (f_v) is trained to learn image features using the shape-bias loss. A pretrained ResNet18 (f_s) is used as feature extractor for 2D distance transforms. Different categories are depicted by distinct shapes (▲, ●, ■) in the latent space. (Red denotes shape features. Blue denotes image features. (Right) The shape-bias loss is composed of 3 losses, (a) mean squared error (b) pairwise loss and (c) contrastive loss. In addition, we use cross-entropy loss (for supervised learning). Blue dotted lines denote distance minimization. Red dotted lines denote distance maximization.

towards noise caused by changes in illumination, scale, and style. Even though the idea of using shape invariance is intuitive, the importance of shape is shadowed by the tremendous success of appearance-based methods, and the challenges of capturing shape information in the complex representations learned by CNNs [16, 19].

Consider the objective of obtaining a mapping M: $I \rightarrow \mathbb{R}^{C\times 1}$ where I denotes image domain, and C denotes the number of classes. We can approximate the optimal function M using a computationally convenient embedding form, f, which extracts representations ϵ from images. ϵ is optimal feature for classification, which may (or may not) capture shape. We hypothesize there is an implicit feature s, which can capture shape holistically. In practice, inferring a function f^* that uses shape invariance s explicitly, is not enforced in machine learning. Rather, one looks for f which is generally associated with optimal classification. The problem can be viewed as having two extreme ends leading to appearance-only and shape-only models, both leading to inferior generalization.

We deal with the two problems implied above as follows. In lieu of deriving the underlying shape domain from 3D point cloud data, we approximate it using the more accessible 2D distance transform. We encourage models to learn both appearance and shape simultaneously in a contrastive manner, which has a regularizing effect and achieves superior generalization performance.

3.1 Shape Representation

An intuitive shape representation are 3D point clouds (PC) [40]. We use this representation as a variant of our method. PC is fed directly into DGCNN [46] for processing. However, PC data is not available for most computer vision tasks. Thus, we also consider shape data to be in the form of a distance transform (DT) computed from the input image. First, a mask is used to identify non-background pixels, then DT is computed over the masked images. DT (often referred as Euclidean DT) has positive values inside the object computed using different radii of dilation. Fig. 2 shows a visualization for DT computed for different objects.

DT is a widely used technique in shape analysis [7], but our proposed use of DT for recognition and representation learning via a contrastive loss is novel. DT simulates a convenient and succinct representation of shape without any significant computational overhead. To compute a shape representation, we train a ResNet18 on DT and DGCNN on PC for supervised classification. The trained model is used as a feature extractor to compute the shape representation for each instance. We extract multiple features by sampling images (different views) for the instance. Finally, we compute a mean shape representation for each instance. The image-based model is encouraged to learn a representation similar to the pre-trained shape features as shown in Fig. 3. Note that these shape features are not updated while training image-based models.

3.2 Shape Bias Loss

Appearance feature representations are ubiquitous, so we deliberately address alignment of image-to-shape feature representations. We rely on two losses used in prior work [40], and a contrastive loss which has *not previously been used* to bridge appearance and shape representations. We first minimize the pointwise squared Euclidean distance between the shape features and corresponding image features. For a minibatch $\mathcal{B}$,

$$\mathcal{L}_{mse} = \sum_{k\in C} \sqrt{\frac{1}{n_k}\sum_{j=1}^{n_k}\left(\phi_j^v - \phi_j^s\right)^2} \tag{1}$$

where C denotes the number of classes[1], n_k denotes the number of images in each class, ϕ^v denotes image feature and ϕ^s denotes shape feature.

[1]For notation, we use the term 'classes' to denote the target labels for a classification task. We use 'categories' to denote objects in Toys4K dataset. The number of classes need not necessarily be equal to number of categories, especially in a low-shot setting.

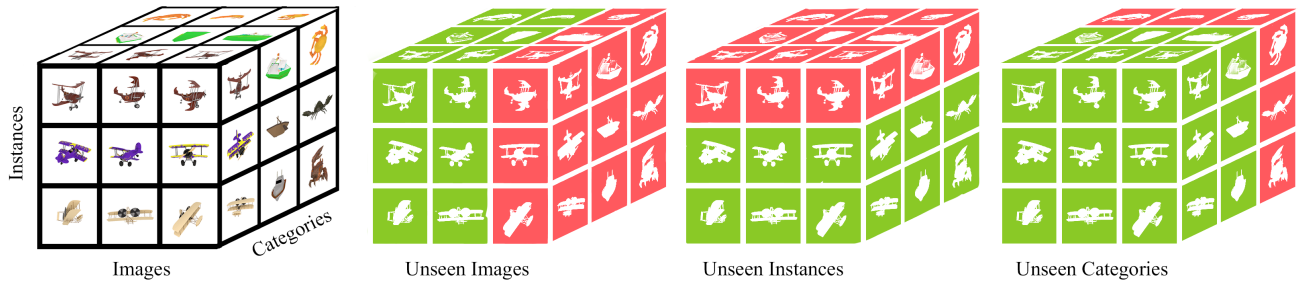

Figure 4: We evaluate shape bias loss across all three axes - *images*, *instances* and *categories* of Toys4K dataset. Each block along the *category* axis is an object, say airplane, boat and crab. Each category has multiple instances, *e.g.* yellow, purple and brown airplanes & each instance has multiple views along the *image* axis.

A pointwise loss is not sufficient as it fails to capture intra-class compactness against inter-class differences. An additional loss constrains the pairwise distances between image representations of object instances to be same as the pairwise distances of the learned shape representations.

$$\mathcal{L}_{pairwise} = \sum_{k \in C} \sqrt{\frac{1}{n_k} \sum_{\substack{j=1 \\ j \neq j'}}^{n_k} \left(\phi_j^v - \phi_{j'}^v\right)^2 - \frac{1}{n_k} \sum_{\substack{j=1 \\ j \neq j'}}^{n_k} \left(\phi_j^s - \phi_{j'}^s\right)^2} \tag{2}$$

where C denotes the number of classes, n_k denotes the number of images in each class, ϕ^v denotes image feature and ϕ^s denotes shape feature. These losses originally appeared in Stojanov *et al.* [40], but unlike their formulation, we implement both the losses $\mathcal{L}_{mse}$ and $\mathcal{L}_{pairwise}$ in a stratified manner *i.e.* the image-to-shape loss is computed within a given class to ensure that the model is forced to minimize distances for shapes within the same class.

We further align image-to-shape features by incorporating a contrastive loss, InfoNCE [13] with an easy-semihard miner [48]. We choose the image feature ϕ^v as an anchor and a shape feature ϕ^s is chosen as a positive sample or negative sample depending on class label.

$$\mathcal{L}_{contrastive} = \sum_{k \in C} \frac{1}{n_k} \left(\sum_{i=1}^{n} -log \left(\frac{e^{\phi_i^v \cdot \phi_+^s / \tau}}{\sum_{j=0}^{M} e^{\phi_i^v \cdot \phi_j^s / \tau}} \right) \right) \tag{3}$$

where, C denotes the number of classes, n_k denotes the number of images in each class, τ is a temperature hyperparameter, and the sum in the denominator is over one positive and M negative samples. ϕ^v and ϕ_+^s need not be extracted from the same input image, but ϕ_+^s should be a shape feature of the same class as ϕ^v. Optimizing InfoNCE loss between image and shape features maximizes mutual information between the image and its contextual shape. As the role of query-key pairs is very significant in the InfoNCE loss, minimizing this loss with image-shape pairs imbibes invariance in the learned representations by maximising the similarity between image and shape representations. For a given image feature ϕ_i^v (anchor), the easy semi-hard miner [48] finds M hardest shape features such they are further away from the selected positive shape feature ϕ_+^s.

The final shape bias loss is:

$$\begin{aligned} \mathcal{L}_{shape-bias} = \lambda_m \cdot \mathcal{L}_{mse} + \lambda_p \cdot \mathcal{L}_{pairwise} \\ + \lambda_c \cdot \mathcal{L}_{contrastive} \end{aligned} \tag{4}$$

During training, $\mathcal{L}_{shape-bias}$ is minimized along with cross-entropy loss. The coefficients were computed empirically using hyperparameter optimization *i.e.* (10, 1, 0.01, 0.001) for ($\mathcal{L}_{contrastive}$, $\mathcal{L}_{cross-entropy}$, $\mathcal{L}_{mse}$, $\mathcal{L}_{pairwise}$) respectively. For *Unseen categories*, the training and test partitions have distinct categories. Thus, to classify novel categories, we use nearest neighbour with pre-trained ResNet as feature extractor to compute mean feature for each category. The model is evaluated by varying the number of new categories (k-way) and the number of images used to compute prototype for a category (k-shot) [45] using a pre-trained ResNet. It is important to note that *at inference, the model doesn't have access to shape features* rather only images are fed for classification.

4 EXPERIMENTS

We test if our proposed shape bias boosts image classification compared to a variety of baselines (Sec. 4.1), under *Unseen images*, *Unseen instances* of an object (Sec. 4.2) and *Unseen categories* (Sec. 4.3). We compare two forms of shape domains, PC and DT (Sec. 4.5). We find that PC are superior over other methods for the *Unseen images* and *Unseen instances*, while DT are especially successful for *Unseen categories*. Finally, we evaluate if contrastive learning is beneficial in aligning the image-to-shape feature representations (Sec. 4.6).

Datasets and Metrics We evaluate our models on the recent Toys4K dataset [40] for image classification. The dataset consists of 3D shape point cloud data as well as image renders of 4,179 object instances in 105 categories. We partition the dataset into 70:15:15 splits as train, validation and test. We depict the evaluation tasks in Fig. 4. First, we test model's ability to generalize to *unseen images*. The dataset partitions are disjoint, but they contain the same categories and instances. Next, we test along the instance dimension such that dataset partitions has the same categories, but the partitions have disjoint instances, called *unseen instances*. Finally, we test generalization to *unseen categories*. The tasks are arranged in increasing order of complexity assessing higher levels of generalization.

We further evaluate robustness of different models on unseen image manipulations *i.e.* novel textures with the same semantic content.

Table 1: Evaluation of shape-biased models (names in bold) on *unseen images*. As SIN requires finetuning with pretraining on Style, we follow the same procedure for our shape-biased models (bottom).

Modality	Toys4K Acc. ↑	Style Acc. ↑	Cue Conflict Acc. ↑	Average Acc. ↑	Scrambled Acc. ↓
Original Images	**93.48 ± 0.12**	15.09 ± 0.41	36.52 ± 0.51	48.36	16.26 ± 0.58
Edge Images	88.87 ± 0.19	21.73 ± 0.58	**47.17 ± 0.41**	52.59	**8.84 ± 0.74**
STDNN [26]	75.07 ± 0.38	31.23 ± 0.36	40.15 ± 0.47	48.82	9.25 ± 0.41
InfoDropout [38]	90.36 ± 0.14	5.58 ± 0.10	38.15 ± 0.24	44.70	11.95 ± 0.18
MixStyle [52]	92.41 ± 0.12	19.68 ± 0.46	20.12 ± 0.48	44.07	15.33 ± 0.68
DistanceT Biased	86.79 ± 0.21	29.93 ± 0.49	38.01 ± 0.55	51.58	13.77 ± 0.61
PointCloud Biased	92.21 ± 0.20	**31.49 ± 0.45**	40.82 ± 0.46	**54.84**	9.86 ± 0.41
SIN [11]	87.94 ± 0.20	**80.81 ± 0.27**	37.96 ± 0.46	68.90	12.38 ± 0.59
DistanceT Biased	**89.62 ± 0.15**	80.27 ± 0.30	**41.19 ± 0.48**	**70.36**	8.47 ± 0.43
PointCloud Biased	84.81 ± 0.22	74.41 ± 0.31	37.28 ± 0.46	65.50	**6.54 ± 0.24**

For each of these tasks, we create Style and CueConflict corruption variants of Toys4K dataset, following Geirhos *et al*. [11]. To generate Style dataset we carry out style-transfer on images using AdaIN [17] with paintings [34] as texture images, and set the stylization coefficient α=0.5. CueConflict images have conflicting texture and semantics. To this end, we first use a non-parametric example-based image quilting to construct a texture image [35] from a random image from the Toys4K dataset. The texture image is used to fill the silhouette of the original image. For both Style and CueConflict, the label assigned to final image is same as that of original image. As shown in Fig. 1, the labels assigned is same as the original image *i.e*. apple and banana respectively, even though the texture may be atypical. Following Mummadi *et al*. [32], we evaluate a model's susceptibility to local texture-bias. We obfuscate the semantics of the original image by scrambling pixel blocks. A model which properly captures object semantics and shape shouldn't classify the image as the original object because mere presence of texture segments should not correspond to existence of the object. To evaluate robustness to unseen image manipulations, in most experiments we focus on zero-shot generalization *i.e*. we evaluate a model's robustness without any retraining on any of the Style, CueConflict and Scrambled datasets.

4.1 Methods Tested

To evaluate shape-texture bias, we compare with recent baselines which claim to improve the shape bias in deep learning models. We also compare our approach to data augmentation techniques to explore if the inclusion of explicit shape bias has advantages over improvement on corruption robustness. We choose ResNet18 [14] as the default architecture for all models. **Original Images** represent an off-the-shelf ResNet18 trained on on the original Toys4K as a baseline. **Edge Images** represent the performance of ResNet18 trained on Canny edge maps of the input image ($\sigma = 1$). **DistanceT Biased** is the proposed shape-biased model, which uses DT of the input image as shape data. **PointCloud Biased** is another variant of our shape-biased model, which uses PC corresponding to the input image (available in Toys4K) as shape data. **STDNN** [26] proposes soft label assignment and combination of stylized images from random classes to learn a debiased network. We compare STDNN for *Unseen images* & *Unseen instances* but not in the low-shot setting for *Unseen categories* as their proposed technique uses label information. Similar to STDNN, **MixStyle** [52] proposes to combine stylized images in an implicit manner by mixing instance-level feature statistics of training samples. **InfoDropout** [38] alleviates texture information from the image by adopting a Dropout-like algorithm based on local self-information in the image *i.e*. regions containing more contrasting and distinctive information than their surroundings. **SIN** [11] is trained on stylized images and then finetuned on the original dataset keeping the feature backbone intact. The work is driven by style transfer [9] which was used to generate *uninformative-styled* images which can be in-turn used to eliminate over-reliance on common texture of objects for classification. For comparison to other baselines (such STDNN, InfoDrop and MixStyle), we evaluate model performance without any retraining. However, to have a fair comparison to SIN, we create variants of our two proposed shape-biased models where we follow the same procedure, *i.e*. finetuing on Toys4K preceded by pre-training on Style. We do not include this model in the low-shot setting, where the model sees disjoint set of classes in training and inference.

4.2 Can shape-bias enhance supervised models?

Domain generalization is primarily associated with recognizing novel data distribution during inference. Going beyond, we combine atypical distributions such as Style and CueConflict with various tasks as shown in Fig. 4. Tables 1 and 2 show the results on *Unseen images* and *Unseen instances* respectively. *Unseen images* emulate a general machine learning setting, where we expect the model to generalize to novel views if it has already seen the object during the training phase. In contrast, *Unseen instances* emulates a harder task: the model may have seen a yellow Wright flyer, but is expected to recognize a purple jet. As the models have very diverse performance on Style and CueConflict datasets (Tables 1 and 2), we evaluate robustness by observing the mean accuracy across the Original, Style and CueConflict variations.

First, in the top of Tables 1 and 2, we note that the performance of ResNet trained on Original Images drops from 93.48% to 15.09% and 70.70% to 8.79% for *Unseen images* and *Unseen instances* on

Table 2: Evaluation of shape-biased models (names in bold) on *unseen instances*. As SIN requires finetuning with pretraining on Style, we follow the same procedure for our models (bottom).

Modality	Toys4K Acc. ↑	Style Acc. ↑	Cue Conflict Acc. ↑	Average Acc. ↑	Scrambled Acc. ↓
Original Images	**70.70 ± 0.01**	8.79 ± 0.24	33.98 ± 0.27	37.82	13.09 ± 0.84
Edge Images	70.12 ± 0.02	18.75 ± 0.59	**42.27 ± 0.40**	43.71	10.35 ± 0.94
STDNN [26]	62.78 ± 0.64	30.30 ± 0.56	38.65 ± 0.55	43.91	9.75 ± 0.37
InfoDropout [38]	68.59 ± 0.29	8.75 ± 0.34	33.44 ± 0.32	36.93	12.03 ± 0.34
MixStyle [52]	70.51 ± 0.08	14.26 ± 0.51	35.16 ± 0.05	39.98	13.09 ± 1.05
DistanceT Biased	58.44 ± 0.60	21.56 ± 5.74	36.56 ± 0.57	38.85	**7.03 ± 0.46**
PointCloud Biased	68.95 ± 0.18	**33.20 ± 0.81**	35.55 ± 0.32	**45.90**	7.42 ± 0.59
SIN [11]	70.12 ± 0.40	61.72 ± 0.21	36.91 ± 0.35	56.25	10.35 ± 0.89
DistanceT Biased	71.48 ± 0.21	64.84 ± 0.21	35.35 ± 0.13	57.22	**9.38 ± 0.54**
PointCloud Biased	**73.24 ± 0.46**	**73.05 ± 0.16**	**40.62 ± 0.01**	**62.30**	15.82 ± 0.56

the Style dataset respectively. This severe drop in accuracy suggests that high performance gains in off-the-shelf CNNs are limited to datasets with similar texture characteristics as the training dataset. Despite not having the benefit of advanced techniques such as soft labels (STDNN [26]), self-information (InfoDrop [38]), or mixing feature statistics (MixStyle [52]), training a ResNet on edge maps provide a relatively stable model as it suppresses variations in texture and enhances boundary information explicitly. However, using edge maps is not effective in case of Style images, which corrupts the edge information of original images. In contrast, PC-biased model achieves much better overall performance. The DT-biased model achieves significant improvements over Edge-based ResNet on Style images in Table 1 and on both Style & Scrambled images in Table 2. We can infer that augmenting the feature space by deliberate alignment of image-to-shape feature alignment leads to discriminative embedding space robust to unseen image manipulations. We also observe that gaps between DT and PC are small in multiple settings, e.g. Style (1.56%) and CueConflict (2.81%) in Table 1, CueConflict (1.01%) and Scrambled (0.39%) in Table 2.

We evaluate the performance on Scrambled images to judge the susceptibility of a model to local texture by obfuscating semantics (lower performance is better). We observe that PC-biased models are less susceptible to recognize the object for *unseen images* while DT-biased models are more suitable for *unseen instances*. The results suggest that incorporating shape bias leads to relatively robust features which can capture shape holistically and can thus bear atypical transformations of the original object.

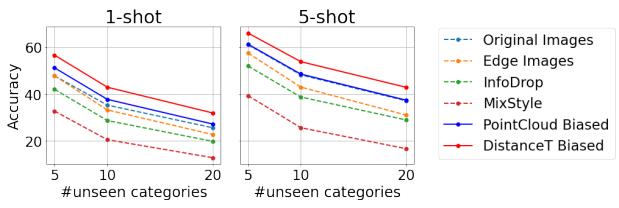

Figure 5: Average performance on Original, Style and CueConflict datasets for *unseen categories*

At the bottom rows of Tables 1 and 2, we observe that SIN is able to perform efficiently on Style images but at a significant cost of training the model again on the Style images. This would not be feasible at scale. Following the same procedure as adopted by SIN, we retrain our shape-biased models *i.e.* PC-biased and DT-biased models, and see these models achieve better accuracy than SIN in many settings, indicating that inclusion of shape bias can improve effectiveness of deep learning models. DT performs better than or comparable to PC on Toys4K, Style and CueConflict in Table 1 (bottom).

4.3 Can shape bias aid in learning new categories?

Table 3 compares the performance of shape-biased models to alternative methods on *Unseen categories*. Fig. 5 provides a visual summary of Table 3 by indicating the mean value across Original, Style and CueConflict datasets. Observe that the DT-biased model outperforms all the baselines in all settings for *Unseen categories* as shown in Fig. 5, and PC-biased model follows subsequently. However, for 5-shot setting the performance of PC-biased model is very similar to Original images. We observe that benefit of shape bias is more prominent for 1-shot both for DT as well as PC. The evidence clearly demonstrates the significance of using shape for rapidly learning novel categories from limited examples, even with unseen image manipulations.

Unlike *Unseen images* and *Unseen instances*, we observe that Edge-based ResNet has inferior performance to ResNet trained on *Original images*. Thus, rejecting all the texture hampers performance. Edge-based ResNet can be viewed as an extreme shape-biased model. Observing the performance on Original images and Edge images, we can infer that the extreme ends of appearance and shape lead to inferior generalization.

4.4 Can shape-bias aid in generalization?

In domain generalization, the target domain has distinct characteristics in contrast to the training datasets. To corroborate the advantages of the proposed image-to-shape feature alignment, we evaluate the robustness to texture variations on the PACS dataset. A good model should be able to capture the object even from distinctive domain

Table 3: Evaluation of shape-biased models (names in bold) for mitigating texture bias on *unseen categories*. Shape bias boosts performance in almost all settings on the original Toys4K and all manipulated versions. k-shot denotes number of images (supports) used for prototype and k-way denotes number of unseen categories at inference.

(a) Model performance on the original Toys4K dataset

Modality	1-shot 5-way Acc. ↑	5-shot 5-way Acc. ↑	1-shot 10-way Acc. ↑	5-shot 10-way Acc. ↑	1-shot 20-way Acc. ↑	5-shot 20-way Acc. ↑
Original Images	62.97 ± 0.34	78.62 ± 0.26	49.38 ± 0.25	66.82 ± 0.21	37.54 ± 0.15	55.15 ± 0.15
Edge Images	59.36 ± 0.36	72.87 ± 0.29	44.74 ± 0.24	59.44 ± 0.21	32.92 ± 0.15	46.68 ± 0.14
MixStyle [52]	38.31 ± 0.30	47.97 ± 0.29	25.55 ± 0.18	33.62 ± 0.19	16.73 ± 0.10	23.18 ± 0.11
InfoDropout [38]	57.82 ± 0.33	75.62 ± 0.27	43.98 ± 0.24	63.74 ± 0.21	33.17 ± 0.47	52.11 ± 0.45
DistanceT Biased	**70.93 ± 0.34**	**84.08 ± 0.23**	**57.80 ± 0.25**	**74.19 ± 0.19**	**45.67 ± 0.17**	**63.19 ± 0.14**
PointCloud Biased	63.09 ± 0.34	77.82 ± 0.26	49.53 ± 0.24	66.37 ± 0.20	37.90 ± 0.15	54.80 ± 0.14

(b) Model performance on Style images

Modality	1-shot 5-way Acc. ↑	5-shot 5-way Acc. ↑	1-shot 10-way Acc. ↑	5-shot 10-way Acc. ↑	1-shot 20-way Acc. ↑	5-shot 20-way Acc. ↑
Original Images	32.79 ± 0.28	39.11 ± 0.30	22.20 ± 0.19	26.46 ± 0.19	15.14 ± 0.11	17.73 ± 0.12
Edge Images	37.87 ± 0.26	43.43 ± 0.25	23.20 ± 0.15	28.23 ± 0.15	14.22 ± 0.08	18.15 ± 0.08
InfoDropout [38]	28.24 ± 0.20	30.82 ± 0.22	15.57 ± 0.11	17.19 ± 0.12	8.55 ± 0.19	9.51 ± 0.21
MixStyle [52]	32.83 ± 0.25	39.51 ± 0.25	20.18 ± 0.15	24.90 ± 0.14	12.13 ± 0.08	15.61 ± 0.08
DistanceT Biased	**39.64 ± 0.26**	**45.21 ± 0.24**	**25.70 ± 0.15**	**30.71 ± 0.15**	16.40 ± 0.09	20.36 ± 0.09
PointCloud Biased	38.44 ± 0.25	43.24 ± 0.23	25.52 ± 0.15	30.23 ± 0.16	**16.68 ± 0.09**	**20.54 ± 0.19**

(c) Model performance on Cue-Conflict images

Modality	1-shot 5-way Acc. ↑	5-shot 5-way Acc. ↑	1-shot 10-way Acc. ↑	5-shot 10-way Acc. ↑	1-shot 20-way Acc. ↑	5-shot 20-way Acc. ↑
Original Images	47.55 ± 0.37	65.65 ± 0.33	34.10 ± 0.24	51.29 ± 0.24	23.89 ± 0.15	38.24 ± 0.15
Edge Images	46.14 ± 0.30	56.25 ± 0.27	31.55 ± 0.19	41.16 ± 0.19	20.72 ± 0.11	27.99 ± 0.11
MixStyle [52]	26.89 ± 0.21	30.41 ± 0.22	15.56 ± 0.12	18.14 ± 0.13	9.26 ± 0.07	10.89 ± 0.07
InfoDropout [38]	39.94 ± 0.25	49.53 ± 0.27	26.32 ± 0.16	35.12 ± 0.17	17.39 ± 0.29	24.93 ± 0.33
DistanceT Biased	**59.32 ± 0.31**	**68.95 ± 0.25**	**45.21 ± 0.22**	**56.59 ± 0.19**	**33.46 ± 0.14**	**44.91 ± 0.13**
PointCloud Biased	52.08 ± 0.31	62.90 ± 0.27	38.04 ± 0.21	48.95 ± 0.19	26.72 ± 0.12	36.60 ± 0.12

(d) Model performance on Scrambled images

Modality	1-shot 5-way Acc. ↓	5-shot 5-way Acc. ↓	1-shot 10-way Acc. ↓	5-shot 10-way Acc. ↓	1-shot 20-way Acc. ↓	5-shot 20-way Acc. ↓
Original Images	38.93 ± 0.33	54.06 ± 0.33	24.79 ± 0.20	38.68 ± 0.23	15.55 ± 0.12	26.60 ± 0.13
Edge Images	32.88 ± 0.23	36.91 ± 0.23	19.16 ± 0.13	22.65 ± 0.13	10.86 ± 0.07	13.20 ± 0.07
MixStyle [52]	29.01 ± 0.23	34.68 ± 0.22	16.86 ± 0.13	20.95 ± 0.13	9.61 ± 0.07	12.47 ± 0.07
InfoDropout [38]	35.59 ± 0.25	42.04 ± 0.26	21.84 ± 0.15	26.88 ± 0.16	12.87 ± 0.27	17.02 ± 0.29
DistanceT Biased	**27.36 ± 0.19**	**28.16 ± 0.18**	**15.35 ± 0.11**	**15.75 ± 0.10**	**8.53 ± 0.06**	**8.73 ± 0.05**
PointCloud Biased	28.88 ± 0.21	30.47 ± 0.02	16.80 ± 0.12	17.97 ± 0.11	9.65 ± 0.07	10.37 ± 0.06

capturing a wide variety of domains. The images typically used as training datasets are reflective of a real-world setting in computer vision. Therefore, we choose the *Photo* domain for training. The trained model is expected to capture abstract objects even if the domain is maximally distinct from real-world picture because the shape of the object is still consistent. We use the *Art paintings, Sketch* and *Cartoon* domains for evaluation. Fig. 6 shows that the DT-biased model achieves best average accuracy on these abstract domains, providing empirical evidence that it is able to capture shape from natural images which is robust to corruptions like paintings and cartoons. Edge-based ResNet has superior performance on *Sketch* dataset, due to similarity between Sketch and Edge images, but it shows an inferior performance on other domains. The experiment provides empirical evidence that rejecting texture completely may hamper generalization. Rather, a model should use both appearance and shape for better generalization.

4.5 Distance Transform vs. Point Cloud

As explained in Section 3.1, we compute a mean feature representation for each instance, by averaging over multiple shape representations (computed over different views) of the same instance. PC provides a permutation-invariant representation for spatial structure. As DGCNN processes PC data directly, it extracts features which are invariant to viewpoint. However, DT is subject to viewpoint. Therefore, shape representations computed by DGCNN for PC are much more discriminative as compared to those computed by ResNet18 for DT. In *Unseen images* and *Unseen instances*, cross-entropy is dominant, and the model focuses on discriminating between the classes. As PC features are more discriminative, using PC leads to superior performance as compared to DT. We observe that Point Cloud-biased model achieves a marginal boost (3.3% and 7.05% for *Unseen images* and *Unseen instances*, respectively) in comparison to DT-biased model but at the cost of 3D data acquisition and processing.

However, the performance on *Unseen categories* is much more decisive to understand the advantages of using shape, as a model with stronger shape bias has a greater potential to recognize new categories effectively. In a low-shot setting (*Unseen categories*), the cross-entropy loss is absent, and the model emphasizes shape-bias loss *i.e.* image-to-shape alignment only. We observe that it is easier to impose constraints on features for contrastive image-to-shape feature alignment, when both shape and image features are similar, *i.e.* both image and DT are extracted via same architecture (ResNet18). However, aligning image features from two distinct distributions *i.e.* image (ResNet18) and PCs (DGCNN) is more challenging. Therefore, we observe that DT-biased model attains substantial boost in accuracy. The superior performance of the Distance Transform-biased model in *Unseen categories* shows that shape information can be exploited efficiently by using the simpler DT.

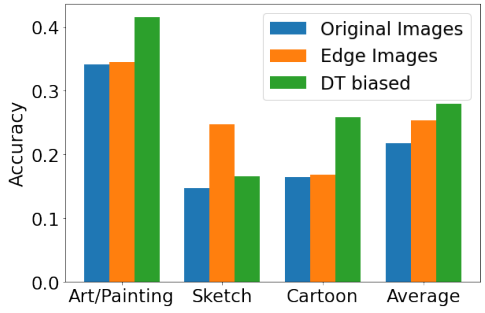

Figure 6: Comparative evaluation of DT-biased model on PACS dataset. The photo domain is used for training, as it reflects training datasets in a real-world setting.

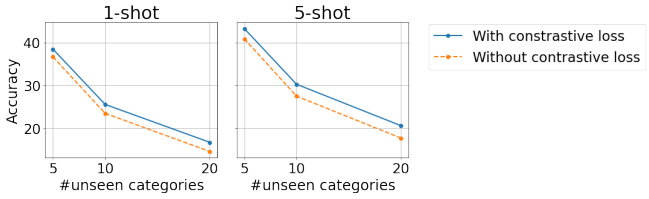

Figure 7: Ablation study showing impact of contrastive loss for PointCloud-biased model for *unseen categories* task.

Table 4: Ablation study for contrastive loss. Blue cells show the top-3 models

MSE	Pairwise	Contrastive	Toys4K	Style	CueConflict	Average	Scrambled
✓			0.3208	0.2363	0.2088	0.2553	**0.1754**
	✓		0.4797	0.1865	0.2940	0.3201	0.2182
		✓	0.4904	0.2296	0.3229	0.3477	0.2288
✓	✓		0.4638	0.1815	0.2871	0.3108	0.2238
	✓	✓	0.5368	0.2291	0.3567	0.3742	0.2505
✓		✓	0.4937	0.2008	0.3327	0.3424	0.2430
✓	✓	✓	**0.5431**	**0.2405**	**0.3600**	**0.3812**	0.2554

4.6 Impact of contrastive loss

In this paper, we present a novel use of contrastive loss for aligning image-to-shape features motivated by cross-modal contrastive approaches on text-to-image alignment. To this end, we incorporate InfoNCE with an easy-semihard miner. Stojanov *et al.* [40] uses mean squared error (MSE) and pairwise losses to align image-to-shape features. However, our experiments (Fig. 7) suggest that such alignment is not optimal. Therefore, we add contrastive loss to penalize the relative distance between corresponding image and shape features directly. As described in Sec. 3, we use image features as an anchor and shape features as a positive/negative samples with an easy-semihard miner. We evaluate the impact of contrastive loss on performance a PC biased model for the *Unseen categories* task as shown in Fig. 7. We also conduct exhaustive ablation study to understand the advantages of individual components in shape-bias loss as shown in Table 4. We see in Table 4 that the top-3 models include contrastive loss. Observe that even using contrastive loss alone (row 3) has a better performance than the model which uses mean squared error and pairwise loss as suggested by Stojanov *et al.* [40]. Furthermore, Fig. 7 clearly demonstrates that using contrastive loss boosts the performance of shape-biased model beyond that which is attained by MSE and pairwise losses.

5 CONCLUSIONS

In this work, we demonstrated the advantages of using shape bias enforced via a contrastive loss, in conjunction with classical CNNs. The benefits of shape invariance is achieved through two representations, namely 3D point cloud and 2D distance transform. Our models learn representations robust to unseen image manipulations such as Style, & CueConflict images. We show our proposed method works better than existing methods which focus on data augmentation as a mechanism to encourage a CNN to capture shape. We empirically demonstrate that using shape-bias leads to superior generalization in both supervised and unsupervised scenarios without necessitating retraining on unseen domains. Finally, we show that benefits of using shape is not restricted to popular representations like 3D point clouds, rather significant benefits can be extracted using simpler distance transform for out-of-domain generalization.

Acknowledgement: This material is based upon work supported by the National Science Foundation under Grant No. 2006885.

REFERENCES

[1] HB Barlow. 1983. Vision: A computational investigation into the human representation and processing of visual information: David Marr. San Francisco: WH Freeman, 1982. pp. xvi+ 397.

[2] Ilaria Bartolini, Paolo Ciaccia, and Marco Patella. 2005. WARP: Accurate Retrieval of Shapes Using Phase of Fourier Descriptors and Time Warping Distance. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 1 (2005), 142–147. https://doi.org/10.1109/TPAMI.2005.21

[3] Harry Blum et al. 1967. *A transformation for extracting new descriptors of shape*. Vol. 43. MIT press Cambridge, MA.

[4] Charles-Olivier Dufresne Camaro, Morteza Rezanejad, Stavros Tsogkas, Kaleem Siddiqi, and Sven J. Dickinson. 2020. Appearance Shock Grammar for Fast Medial Axis Extraction From Real Images. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 14370–14379. https://doi.org/10.1109/CVPR42600.2020.01439

[5] Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. 2020. Learning to Balance Specificity and Invariance for In and Out of Domain Generalization. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IX (Lecture Notes in Computer Science, Vol. 12354)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 301–318. https://doi.org/10.1007/978-3-030-58545-7_18

[6] Sven J. Dickinson, Alex Pentland, and Azriel Rosenfeld. 1992. From volumes to views: An approach to 3-D object recognition. *CVGIP Image Underst.* 55, 2 (1992), 130–154. https://doi.org/10.1016/1049-9660(92)90013-S

[7] Ricardo Fabbri, Luciano da Fontoura Costa, Julio C. Torelli, and Odemir Martinez Bruno. 2008. 2D Euclidean distance transform algorithms: A comparative survey. *ACM Comput. Surv.* 40, 1 (2008), 2:1–2:44. https://doi.org/10.1145/1322432.1322434

[8] Jacob Feldman, Manish Singh, Erica Briscoe, Vicky Froyen, Seha Kim, and John Wilder. 2013. An integrated Bayesian approach to shape representation and perceptual organization. In *Shape perception in human and computer vision*. Springer, 55–70.

[9] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2017. Texture and art with deep neural networks. *Current opinion in neurobiology* 46 (Oct. 2017), 178–186. https://doi.org/10.1016/j.conb.2017.08.019

[10] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nat. Mach. Intell.* 2, 11 (2020), 665–673. https://doi.org/10.1038/s42256-020-00257-z

[11] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. 2019. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=Bygh9j09KX

[12] Robert Geirhos, Carlos R. Medina Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge, and Felix A. Wichmann. 2018. Generalisation in humans and deep neural networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 7549–7561. https://proceedings.neurips.cc/paper/2018/hash/0937fb5864ed06ffb59ae5f9b5ed67a9-Abstract.html

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 770–778. https://doi.org/10.1109/CVPR.2016.90

[15] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. 2020. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=S1gmrxHFvB

[16] Katherine L. Hermann, Ting Chen, and Simon Kornblith. 2020. The Origins and Prevalence of Texture Bias in Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/db5f9f42a7157abe65bb145000b5871a-Abstract.html

[17] Xun Huang and Serge J. Belongie. 2017. Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 1510–1519. https://doi.org/10.1109/ICCV.2017.167

[18] Zeyi Huang, Haohan Wang, Eric P. Xing, and Dong Huang. 2020. Self-challenging Improves Cross-Domain Generalization. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 12347)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 124–140. https://doi.org/10.1007/978-3-030-58536-5_8

[19] Md. Amirul Islam, Matthew Kowal, Patrick Esser, Sen Jia, Björn Ommer, Konstantinos G. Derpanis, and Neil D. B. Bruce. 2021. Shape or Texture: Understanding Discriminative Features in CNNs. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=NcFEZOi-rLa

[20] Yakov Keselman and Sven J. Dickinson. 2005. Generic Model Abstraction from Examples. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 7 (2005), 1141–1156. https://doi.org/10.1109/TPAMI.2005.139

[21] Nikolaus Kriegeskorte. 2015. Deep Neural Networks: A New Framework for Modeling Biological Vision and Brain Information Processing. 1, 1 (nov 2015), 417–446. https://doi.org/10.1146/annurev-vision-082114-035447

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger (Eds.). 1106–1114. https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

[23] Barbara Landau, Linda B. Smith, and Susan S. Jones. 1988. The importance of shape in early lexical learning. 3 (1988), 299–321. https://doi.org/10.1016/0885-2014(88)90014-7

[24] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. 521, 7553 (may 2015), 436–444. https://doi.org/10.1038/nature14539

[25] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M. Hospedales. 2019. Episodic Training for Domain Generalization. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 1446–1455. https://doi.org/10.1109/ICCV.2019.00153

[26] Yingwei Li, Qihang Yu, Mingxing Tan, Jieru Mei, Peng Tang, Wei Shen, Alan L. Yuille, and Cihang Xie. 2021. Shape-Texture Debiased Neural Network Training. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=Db4yerZTYkz

[27] C. C. Lin and R. Chellappa. 1987. Classification of Partial 2-D Shapes Using Fourier Descriptors. PAMI-9, 5 (sep 1987), 686–690. https://doi.org/10.1109/tpami.1987.4767963

[28] Bria Long and Talia Konkle. 2018. The role of textural statistics vs. outer contours in deep CNN and neural responses to objects. https://doi.org/10.32470/ccn.2018.1118-0

[29] Diego Macrini, Sven J. Dickinson, David J. Fleet, and Kaleem Siddiqi. 2011. Bone graphs: Medial shape parsing and abstraction. *Comput. Vis. Image Underst.* 115, 7 (2011), 1044–1061. https://doi.org/10.1016/j.cviu.2010.12.011

[30] David Marr. 2010. Representing Shapes for Recognition. , 295-328 pages. https://doi.org/10.7551/mitpress/9780262514620.003.0006

[31] Carolyn B. Mervis. 1987. Child-basic object categories and early lexical development. *U. Neisser (Ed.), Concepts and conceptual development: Ecological and intellectual factors in categorization* (1987), 201–233.

[32] Chaithanya Kumar Mummadi, Ranjitha Subramaniam, Robin Hutmacher, Julien Vitay, Volker Fischer, and Jan Hendrik Metzen. 2021. Does enhanced shape bias improve neural network robustness to common corruptions?. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=yUxUNaj2Sl

[33] Maruthi Narayanan, Vickram Rajendran, and Benjamin Kimia. 2021. Shape-Biased Domain Generalization via Shock Graph Embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1315–1325.

[34] Kiri Nichol. 2018. Painter by Numbers, Does every painter leave a fingerprint? https://www.kaggle.com/c/painter-by-numbers/.

[35] Anastasia Opara. 2019. More Like This, Please! Texture Synthesis and Remixing from a Single Example. https://github.com/EmbarkStudios/texture-synthesis.

[36] Morteza Rezanejad and Kaleem Siddiqi. 2013. Flux graphs for 2D shape analysis. In *Shape perception in human and computer vision*. Springer, 41–54.

[37] Ehud Rivlin, Sven J. Dickinson, and Azriel Rosenfeld. 1994. Recognition by functional parts [function-based object recognition]. In *Conference on Computer Vision and Pattern Recognition, CVPR 1994, 21-23 June, 1994, Seattle, WA, USA*. IEEE, 267–274. https://doi.org/10.1109/CVPR.1994.323839

[38] Baifeng Shi, Dinghuai Zhang, Qi Dai, Zhanxing Zhu, Yadong Mu, and Jingdong Wang. 2020. Informative Dropout for Robust Representation Learning: A Shape-bias Perspective. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 8828–8839. http://proceedings.mlr.press/

v119/shi20e.html

[39] Kaleem Siddiqi, Ali Shokoufandeh, Sven J. Dickinson, and Steven W. Zucker. 1998. Shock Graphs and Shape Matching. In *Proceedings of the Sixth International Conference on Computer Vision (ICCV-98), Bombay, India, January 4-7, 1998*. IEEE Computer Society, 222–229. https://doi.org/10.1109/ICCV.1998.710722

[40] Stefan Stojanov, Anh Thai, and James M. Rehg. 2021. Using Shape To Categorize: Low-Shot Learning With an Explicit Shape Bias. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 1798–1808. https://openaccess.thecvf.com/content/CVPR2021/html/Stojanov_Using_Shape_To_Categorize_Low-Shot_Learning_With_an_Explicit_Shape_CVPR_2021_paper.html

[41] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 1–9. https://doi.org/10.1109/CVPR.2015.7298594

[42] Stavros Tsogkas and Sven J. Dickinson. 2017. AMAT: Medial Axis Transform for Natural Images. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2727–2736. https://doi.org/10.1109/ICCV.2017.295

[43] Cristina Vasconcelos, Hugo Larochelle, Vincent Dumoulin, Rob Romijnders, Nicolas Le Roux, and Ross Goroshin. 2021. Impact of Aliasing on Generalization in Deep Convolutional Networks. *CoRR* abs/2108.03489 (2021). arXiv:2108.03489 https://arxiv.org/abs/2108.03489

[44] Shujun Wang, Lequan Yu, Caizi Li, Chi-Wing Fu, and Pheng-Ann Heng. 2020. Learning from Extrinsic and Intrinsic Supervisions for Domain Generalization. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IX (Lecture Notes in Computer Science, Vol. 12354)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 159–176. https://doi.org/10.1007/978-3-030-58545-7_10

[45] Yan Wang, Wei-Lun Chao, Kilian Q. Weinberger, and Laurens van der Maaten. 2019. SimpleShot: Revisiting Nearest-Neighbor Classification for Few-Shot Learning. *CoRR* abs/1911.04623 (2019). arXiv:1911.04623 http://arxiv.org/abs/1911.04623

[46] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. 2019. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.* 38, 5 (2019), 146:1–146:12. https://doi.org/10.1145/3326362

[47] Yongchao Xu, Yukang Wang, Stavros Tsogkas, Jianqiang Wan, Xiang Bai, Sven J. Dickinson, and Kaleem Siddiqi. 2021. DeepFlux for Skeleton Detection in the Wild. *Int. J. Comput. Vis.* 129, 4 (2021), 1323–1339. https://doi.org/10.1007/s11263-021-01430-6

[48] Hong Xuan, Abby Stylianou, and Robert Pless. 2020. Improved embeddings with easy positive triplet mining. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2474–2482.

[49] Zecong Ye, Zhiqiang Gao, Xiaolong Cui, Yaojie Wang, and Nanliang Shan. 2022. DuFeNet: Improve the Accuracy and Increase Shape Bias of Neural Network Models. *SIViP* (2022). https://doi.org/10.1007/s11760-021-02065-3

[50] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. 2019. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 6022–6031. https://doi.org/10.1109/ICCV.2019.00612

[51] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=r1Ddp1-Rb

[52] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. 2021. Domain Generalization with MixStyle. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=6xHJ37MVxxp