The Role of Shape for Domain Generalization on Sparsely-Textured Images

Narges Honarvar Nazari
University of Pittsburgh
nah114@pitt.edu

Adriana Kovashka
University of Pittsburgh
kovashka@cs.pitt.edu

Abstract

State-of-the-art object recognition methods do not generalize well to unseen domains. Work in domain generalization has attempted to bridge domains by increasing feature compatibility, but has focused on standard, appearance-based representations. We show the potential of shape-based representations to increase domain robustness. We compare two types of shape-based representations: one trains a convolutional network over edge features, and another computes a soft, dense medial axis transform. We show the complementary strengths of these representations for different types of domains, and the effect of the amount of texture that is preserved. We show that our shape-based techniques better leverage data augmentations for domain generalization, and are more effective at texture bias mitigation than shape-inducing augmentations. Finally, we show that when the convolutional network in state-of-the-art domain generalization methods is replaced with one that explicitly captures shape, we obtain improved results.

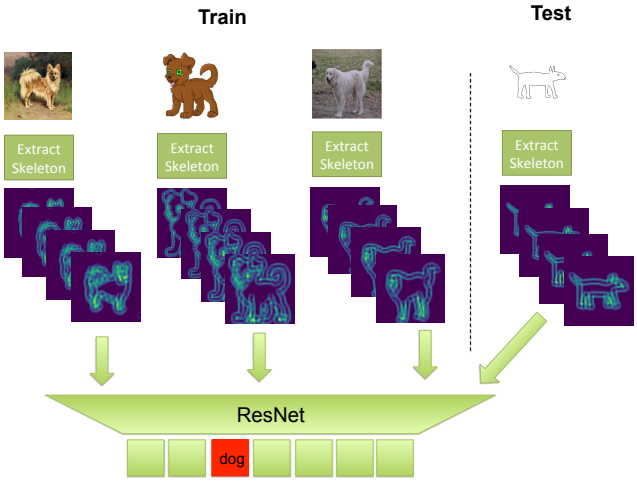

Figure 1. Objects of the same category share common shape. The dogs from paintings, cartoons and photos have varying texture across domains, but their skeletons are similar (e.g. in the legs and tails). We test two shape-representations, one of which computes skeletons in a soft way, and use it in a domain generalization setting, where test images are from a disjoint domain (sketches).

1. Introduction

Appearance-based convolutional representations have advanced visual recognition, but the community has primarily focused on the setting where training and test sets belong to the same distribution. It is now well-known that models trained on one dataset do not generalize well to others [16, 48, 11, 28, 29, 15]. This is problematic because in many real-world cases, we do not have access to plentiful data from the domains our model will be applied on. In contrast to computer vision models, humans have little trouble recognizing object categories across domains: children easily recognize animals in cartoons of different drawing styles, with zero/few training samples.

Prior research shows that *shape* is largely important for *human* vision [27, 14, 1, 50]. On the other hand, prior work in *computer* vision shows that convolutional representations are biased towards texture [17, 2, 23]. However, as shown in Fig. 1, shape is more robust to domain shifts than texture: the shape of the legs and tails of dogs is similar, even though appearance and texture vary across domains. Prior work [9] shows that later layers of a network are less domain-specific than earlier ones, and it is namely later layers that capture larger, more global, shape-like patterns, implicitly.

We compare the potential of two representations that explicitly capture shape, for boosting domain generalizations results. We focus on sparsely-textured objects, such as those portrayed in cartoons and sketches. Many real-world documents contain sparsely-textured imagery, e.g. comic books. Importantly, these are the domains that domain generalization approaches typically struggle with (i.e. where they achieve lowest results).

First, we compare two mechanisms to capture shape: both rely on first converting the image to an edge map that partly removes texture. One representation directly learns a convolutional network on top of the edge map, which has already suppressed texture to some degree. The other relies on new, specialized shape-responsive circular filters akin to Laplacian of Gaussians. These filters create an approxima-

tion of the Medial Axis Transform [4] of an object, which is a way to represent the inner skeleton and shape of the object. Importantly, our representation is denser than the edge map, and we show this is important for several domains where the original images contain sparse information (are not highly textured). We feed our representation to a standard, trainable ResNet to compute a hierarchy of shape-based features. We finally fuse either or both of these shape representations, with a standard, appearance-based convolutional representation (ResNet-18). We show that shape representations are very effective at domain generalization, especially for sparse images. We boost the performance of methods that explicitly tackle domain generalization but use appearance-only representations [15, 29, 8, 16].

Our second contribution is extensive analysis of the factors that affect how shape boosts domain robustness. We compare the impact of edge quality, including blurring before edge detection, and the impact of different data augmentations. We show that simpler edge extraction methods work better for sparsely-textured images, and that more blurring before edge extraction is helpful for densely-textured images. We test various shape representations on the PACS, Office-Home, and DomainNet datasets. We also compare to a prior technique [23] that discovers some data augmentations reduce texture bias, but our method works better than this prior texture bias mitigation technique.

To summarize, our contributions are as follows: (1) extensive comparisons of edge- and medial axis-based shape representations, with different edge extraction mechanisms; (2) tests of the sensitivity of multiple shape representations to edge quality, for domain generalization; (3) detailed examination of the contribution of data augmentation; (4) improved performance of statistical domain generalization methods by using shape-based representations; and (5) a new mechanism for a convolutional network to more explicitly capture shape, which is robust to domain shifts, especially for sparsely-textured objects.

2. Related work

Prior work [17, 23] shows that convolutional networks are more likely to use texture as a cue for classification, rather than shape. A few recognition methods rely on shape explicitly, and we describe some of these below. We next discuss work on domain generalization; none of this work has combined a shape-based representation with statistical bridging of features across domains, as we do.

Shape representations. While shape modeling and reconstruction methods exist [44, 31], shape has been used for recognition to a very limited extent. Shape-based 2D image recognition includes [3, 46, 26] but these do not utilize hierarchical convolutional representations. More recently, researchers have formulated approaches for extracting the skeleton of objects [47, 60, 53, 6], e.g. for image reconstruction, based on the well-known Medial Axis Transform [4]. These typically rely on skeleton annotations, while we do not use supervision, and they do not show skeletons' applications for object classification. [39] train a standard CNN on top of a weighted skeleton representation for scene recognition. However, a scene contains many objects hence the contour map is dense, while an image in a domain generalization dataset [28, 49] contains a single object, hence a contour image is more sparse, especially for sketches and cartoons. Therefore, a CNN may not have enough signal to learn a useful representation; we compare against [39].

Shape in retrieval. A number of prior works model sketches and learn cross-modal photo-sketch spaces for retrieval [41, 58, 43, 35, 36, 40]. Many of these require paired photo-sketch data (same object instances per pair) e.g. from [41], which is more challenging to obtain than *unpaired* samples from different domains, as in standard domain generalization datasets. We also tackle a *different task* (classification rather than retrieval), in the setting where no target domain samples are available at training time, and we focus on a broader set of domains beyond photo and sketch. Radenovic et al. [38] use learnable parameters to extract a filtered edge representation, with soft thresholding of weaker edges. Their network is fine-tuned using *paired* photo-edge data automatically generated through a specialized 3D reconstruction framework. As a side task to retrieval, they also show results for domain generalization. However, [38] only compare to a CNN trained with multiple source domains (the simplest baseline in our results), not to dedicated domain generalization methods, and do not combine their edge method with such generalization methods. We show that our skeleton-based representation exceeds a variant of [38] when details in the image are sparse. We are not aware of prior work that softly extracts object skeletons and computes a dense, convolutional shape representation.

Shape bias. Geirhos et al. [17] explore the role of texture and shape in CNN-based models and humans. They construct images to assess texture and shape biases: e.g. silhouettes, edge images, texture images (small patch of elephant skin). CNN models outperform humans for texture, but underperform for edge images and silhouettes. These results demonstrate that appearance-based CNN models do not capture shape well, and motivate our work. *While [17, 23] cope with texture bias through data augmentation, we explicitly capture shape through new filters, and achieve improved results on domain generalization.* We examine how augmentation impacts robustness to domain shifts for different domains and methods.

Sketch representations. One of the domains we test our shape representation on is the sketch domain, which implicitly contains strong information about shape. While we consider a static representation of sketches, they have also been represented as temporal constructs [55], using transformers

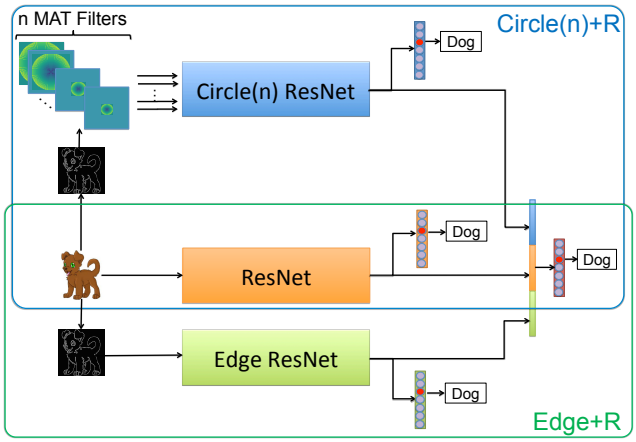

Figure 2. Main method. We combine explicit shape representations (through special shape filters, top, or an edge map, bottom) with a standard texture-based CNN representation (middle).

[56], recurrent [19] or text convolutional networks [57].

Domain generalization. The discrepancy of performance across domains is studied in work on domain adaptation (DA) and generalization (DG). One common technique is to minimize the discrepancy between domains in feature space [48, 30, 5, 16, 45]. Researchers have proposed to align domains at the pixel or patch level, align moments and prototypical samples, align class relations, adaptively tune parts of the network, explicitly expose a network to domain mismatches at training time, suppress dominant features at training time, use auxiliary signal from self-supervised tasks, separately compute batch statistics per domain, etc. [11, 52, 29, 34, 18, 25, 8, 15, 51, 10, 61, 42, 24]. The main limitation of prior DA/DG methods is *they have used similar types of representations as those used by methods that aim to "overfit" to dataset characteristics*, i.e. responses to appearance filters. DA/DG methods have failed to explore shape as an intuitive representation humans rely on as they naturally perform generalization. We compare to three prior works in this realm [15, 29, 8, 16]. When our medial axis-based shape features are used instead of ResNet features in [29] or [15], performance improves. In recent work [33] published after ours commenced, shock graphs (related to skeletons) are used for domain generalization, but the representation is not dense, and is used in the somewhat unrealistic setting of no ImageNet pretraining.

Corruption robustness. Recent work [21, 22, 59] proposes representations robust to synthetic corruptions (e.g. Gaussian noise, motion blur, snow) while we examine real dataset shifts (e.g. cartoon vs sketch), and focus on shape as a novel technique to increase generalization.

3. Shape representations with edges and MAT

We capture shape through (1) a convolutional network trained using edge maps as inputs, or (2) circular filters that approximate a Medial Axis Transform (MAT) skeleton [4], but compute a dense representation, to increase the amount of signal that a network taking the skeleton as input can capture. We train the shape networks jointly with a standard CNN, with shape and texture branches combined via a fully-connected layer. We experiment with CIRCLE(N)+R (top two branches, which uses n circular filters), EDGE+R (bottom two) and CIRCLE(N)+EDGE+R (all three).

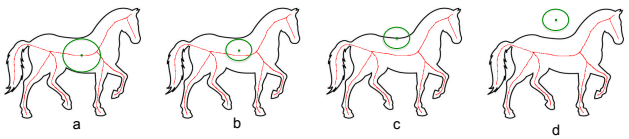

Figure 3. Different situations of the overlap (or lack of overlap) of filters and the MAT of a horse. Filter responses should be large for (a), medium for (b), and zero for (c, d).

3.1. Convolutions over an edge map

Our first shape representation simply relies on edges. First, we convert the image to grayscale, and use Canny edge detection [7] to extract the object edges. We set the lower and upper thresholds in Canny's classic algorithm by visual inspection on the training set, and show sensitivity to different amounts of blurring before edge extraction, in Table 4. Blurring is a classic way to reduce the amount of detail (texture) and obtain coarser contour edges (i.e. just the boundary of a shirt, rather than the patterns on a shirt). We also use a more recent edge detection method, namely Holistically-Nested Edge Detection (HED) [54]. We achieve the best results across both shape methods when filtering edge values less than 0.5 and binarizing the resulting image. Unlike Canny, HED requires supervised training which could result in domain-*specific* edge detection models. We find HED consistently does worse than Canny for the cartoon domain which is different from the photos HED was trained on. We then feed the edge map to a standard, trainable ResNet-18 convolutional network, which learns to capture shape since the edge input removes texture to a certain degree. In experiments, we show that the best domain generalization results are achieved when sufficient blurring is applied. While this approach is simple, *no prior work has extensively evaluated the impact of the type of edge representation in domain generalization,* and combined shape with state-of-the-art DG techniques.

3.2. Specialized circular MAT filters

Our second approach is to pass the edge image through a convolutional layer with specialized filters. We construct up to 16 filters, which are circles of different sizes and mimic the process of creating MAT skeletons. We use the created filters as kernel weights in a convolutional layer, followed by batch normalization and RELU. We feed the output from

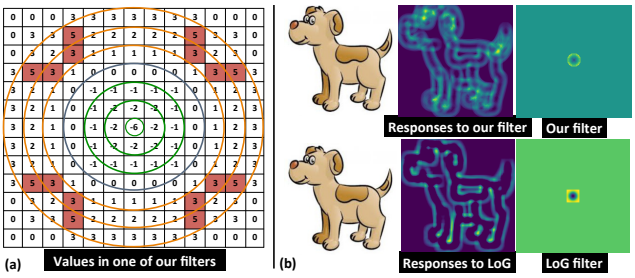

Figure 4. (a) One of our circular filters used to compute a MAT. For a line crossing the filter in the middle, the response would be zero as desired. (b) Comparison of one of our filters vs a comparably-sized LoG filter. We show larger filters in Fig. 5.

our shape-based layer to a trainable ResNet-18, to compute a hierarchy of shape-based features. We also combine the shape filter branch with a standard CNN, using a FC layer.

Background: The object skeleton or medial axis (MA) [4] of an object is the set of points inside the object boundary that represent the overall shape. Every point in the MA (shown in red in Fig. 3) is located at the center of the largest circle which can be inscribed within the object boundary. The set of centers and radii of all maximally inscribed (MI) circles represents the Medial Axis Transform (MAT). We use MAT because the medial axis represents the shape of the object without considering color or texture.

Desired behavior of a MAT filter: In Fig. 3, black pixels show the object boundary, red pixels show the object's medial axis, and green circles are filters. Fig. 3 (a) shows a filter that touches the boundaries of the object in more than one point and its center is located on the medial axis. To be a filter that computes a MAT skeleton in a soft way, a filter should have high value at this position. Fig. 3 (b) shows a filter which touches only one of the boundaries so its center is not located on the medial axis and the filter response should be less than case (a) but larger than zero. Fig. 3 (c) shows a filter whose *center* is located on the object boundary and Fig. 3 (d) shows a filter which does not overlap with any boundary. For (c-d), we want the filter response to be zero. *Note that no prior work proposes specialized filters to compute a MAT skeleton in a soft, dense way.*

Construction: We experiment with circular filters of different sizes, using Alg. 1. We compute the number of circles which can fit in a given square; this number is proportional to the size of the filter. We find the pixels whose coordinates overlap with the equation of each circle. We enforce that the sum of all values inside the filter should be zero. To mimic MAT, the outer circles need to be positive and inner circles need to be negative and of the same absolute values. We show our proposed setting for one filter in Fig. 4 (a). The inner circles with negative values are shown in green. The middle circle with zero value is in blue, and outer circles with positive values in orange. Some pixels

Algorithm 1 Pseudocode for circular filters. *circID* returns the circle ID on which the pixel is located, or *-1* if the pixel is not on a circle, e.g. corners of Fig. 4 (a).

for $iteration = 2, \dots numFilters + 1$ **do**
  $filtSize = 4 \times iteration + 1$
  $numCirc = \lfloor \frac{filtSize}{2} \rfloor + 1$
  $circVal = zeros(numCirc)$
  $filter = zeros([filtSize, filtSize])$
  for $i = 0, \dots numCirc - 1$ **do**
    $circVal[i] = \lfloor \frac{filtSize}{4} \rfloor - i$
  end for
  $circVal[numCirc-1] = circVal[numCirc-1] \times 2$
  for $i = 0, \dots filtSize - 1$ **do**
    for $j = 0, \dots filtSize - 1$ **do**
      $circNum = \text{CIRCID}(i, j, filtSize)$
      if $circNum \neq -1$ **then**
        $filter[i, j] += circVal[circNum]$
      end if
    end for
  end for
end for

overlap with more than one circle (in red) and we need to sum the values for these pixels. The value of the center pixel is twice the highest values of circles because we want the sum of the pixels along the diameter to be zero: if a line crosses the center of filter, the filter response should be zero.

Filter bank: To find the maximally-inscribed circles, the original MAT creates a "vocabulary" of filters with different radii. Similarly, we generate filters to cover a subset of all possible circle sizes. We use filters of size $4m+1$ where $2 \leq m \leq (n+1)$, n is number of filters, and $n \in \{1, 2, 4, 8, 16\}$ (we show different values in our experiments). $4m + 1$ is capped at the image size 224. The pixel values of filters are proportional to the filter size. Filters are resized to the same largest size (padding with zeros as needed).

Discussion: Our filters share some similarity with Laplacian of Gaussian (LoG) filters which also compute circular differences. However, if the filter crosses a line in the middle, the response using LoG would not necessarily be zero, so LoG would not have the benefits of accurately capturing the well-studied MAT. Further, as shown in Fig. 4 (b), our filters result in a denser image compared to LoG, which facilitates the learning of the network. An alternative would be to learn what the circular filters should be (circles are just equations). However, it is not clear whether the circles that are learned will be domain-specific. Instead, we rely on the subsequent learnable layers to inform the model which hand-constructed filters are useful. Our representation offers some robustness to rotation, translation, skew etc. compared to a *binary, sparse* skeleton. For example, if an object was at (x, y) and shifts by (a, b), the

binary skeleton that had value 1 at (x, y), would now have 0. Our representation is softer/denser, so there will be some response at both (x, y) and (x+a, y+b). We found ellipse filters (to cover different orientations) had limited benefits.

Implementation details: We initialize the ResNet part in each branch (whether operating over skeleton outputs, edge maps, or pixels) to those trained on ImageNet. Each branch has a fully-connected (FC) layer which is used to separately train it for the classification task. We combine the branches using another FC layer. We allow the ResNet convolutional, FC layers of each branch and the FC layer for combination of branches, to train. We trained our models and all baselines with Stochastic Gradient Descent (SGD), initial learning rate of 0.001, 30 epochs and batch size of 128. The learning rate decreases to 0.0001 after 24 epochs. Batches contain samples from each of the source domains.

4. The role of shape: Experiments

We make four key findings. First, we show a shape-based representation is greatly superior to an appearance-based one, for domain generalization (Table 1, 6, 7), and that relative performance is consistent regardless of the edge extraction mechanism (Table 1). The MAT-based representation is especially useful for the sparse domains, i.e. cartoons and sketches. Second, we demonstrate a combined shape and appearance representation is superior to an appearance-only one for two domain generalization methods (Table 2, 6). Third, we show that the strong performance of shape methods persists when data augmentation is applied, with the combination of CIRCLE(N)+EDGE+R being strongest, followed by CIRCLE(N)+R (Table 3). We also show that we can better leverage augmentations that generally increase texture bias, if using them in conjunction with a shape-based method. Fourth, we demonstrate the domain dependence of the optimal amount of blurring applied before edge extraction (Table 4) and optimal number of circles (Table 5).

We describe the methods and setup in Sec. 4.1, our results on PACS in Sec. 4.2, and on OfficeHome and DomainNet in Sec. 4.3. *We focus on domains with low texture, namely cartoons, sketches and clipart. Results are lowest for the baseline for these sparsely-textured domains, indicating larger need to develop improved representations.*

4.1. Experimental setup

Methods compared: We use ResNet-18 [20] for all of the following methods, pretrained using ImageNet [13], as in [8]. The **simplest, but strong** baseline is:

- DEEP ALL: Pools together data from different source domains and does not explicitly apply domain regularization/generalization techniques.

We test **four methods for representing shape:**

- EDGE: We train a ResNet using edge maps. Because edge images have much of their texture removed (if using a sufficiently large threshold), this method captures shape. The initial weights are borrowed from a model trained on ImageNet (averaging over the RGB channels for the first layer because edges are grayscale). This baseline is inspired by [39] that captures shape for scene rather than object classification. We do not weight contour saliency as in [39], but the results are far below DEEP ALL and saliency weighting would not compensate the gap.
- EDGE+R: The edge map representation is combined with a standard appearance ResNet, as described in Sec. 3.1. This method is similar to [38], but [38] did not study the impact of edge quality on generalization performance, and did not combine DG and shape methods.
- CIRCLE(N)+R: We compute a soft representation of an object's skeleton using n circular filters of different sizes, as described in Sec. 3.2. The number of input channels is n. We use the mean over RGB of the first conv layer from a model pre-trained on ImageNet, with small random noise added to obtain different weights per channel.
- CIRCLE(N)+EDGE+R: This architecture combines the circular filter stream CIRCLE(N), edge stream EDGE and appearance stream R, and corresponds to the full Fig. 2.

We compare to four **domain generalization methods** which use standard appearance representations:

- JIGSAW: Carlucci et al. [8] propose an approach that encourages generalization by asking networks to solve a self-supervision task (of predicting how pieces of an image were shuffled) in addition to classification. We use 31 permutations to shuffle. On PACS, 90% are ordered and 10% are shuffled, and on OfficeHome, 70% are ordered and 30% shuffled (best setting we could find).
- EPISODIC training by Li et al. [29] exposes a network to domain shifts via three strategies. It trains multiple domain-specific networks, but uses a single network at test time, and has 4x the parameters of DEEP ALL. We also combine EPISODIC with our shape representation using the combination of features extracted from CIRCLE(2) and RESNET instead of just RESNET.
- MASF by Dou et al. [15] exposes the training process to domain shift by splitting the training data into meta train and meta test sets. In addition to hard label classification, it aligns the soft label distribution between meta train and meta test by minimizing their symmetrized Kullback–Leibler (KL) divergence and forces the model to learn a domain-invariant representation. We combine MASF with shape by using the MASF training paradigm on CIRCLE(2) and RESNET independently and combining the extracted features using a FC layer.
- GRADREV by Ganin et al. [16] trains a domain classifier and negates (reverses) the gradient, to ensure similar features are extracted from images in different domains. We perform reversal over the source domains.

Data, splits, metrics: We primarily use PACS, but

also include results on the Office-Home and DomainNet datasets. **PACS** [28] contains 9,991 images from seven classes and four modalities: cartoon, sketch, art painting and photo. **Office-Home** [49] contains 15,500 images from 65 different categories and four modalities: clipart, art, product and real-world. **DomainNet** [37] contains 569,010 images from 345 classes and 6 modalities: clipart, quickdraw, sketch, infograph, painting, and real. For DomainNet only, we focus on three domains that are sparsely-textured, and use a subset of the images to reduce computational cost. One target domain is used to test in each experiment, and the remaining domains are used as sources to train on (using a 90%/10% split for train/val). We report standard top-1 accuracy for object classification. For every experiment, we train each model 3 times and report the average accuracy.

Settings: It is well-known that implementation details contribute greatly to performance differences between methods; see [12, 32]. In the results for "Deep All", [8] observe the numbers reported in different papers vary by up to 6% (65.3% reported for "Deep All" in one paper, and 71.5% in another, using the same architecture). To isolate the effect of different implementation details and compare methods on equal footing, we compute the performance for each method in our own environment, using code from the original papers [8, 29, 15]. We train the batch normalization layers for all methods, as done in the majority of prior work. An alternative is to freeze the batchnorm layer; [29] show this may help even for the basic "Deep All" method, but the effect is not well studied for other baselines, and it depends on the distribution of samples from different modalities per batch. We do not use weight decay. We use Canny edge detection in most tables except Table 1.

Data augmentation: In most tables, we do not use data augmentation, because prior work [23] shows it has complex effects on shape and texture biases and we want to isolate these effects. [29] also do not use augmentation. In Table 3, we study the impact of different types of augmentation on generalization performance, using our implementations for Gaussian and Sobel and the PyTorch implementations for the rest of augmentations. Color Jittering changes the brightness, contrast, saturation and hue of the image. Grayscale converts the image to grayscale. Random Resized Crop crops the image to a random size and aspect ratio. Random Horizontal Flip randomly flips the image. Gaussian adds Gaussian noise to the image. Sobel applies the Sobel filter on the image. We observe the benefit of shape methods over the DEEP ALL baseline is generally maintained. *With data augmentation, the relative contribution of our MAT-based representations (*CIRCLE(N)+R *and* CIRCLE(N)+EDGE+R*) over* EDGE+R *increases.*

Method / Target	Cartoon	Sketch	Painting	Photo	Avg
DEEP ALL [20]	0.7031	0.6181	0.7155	0.9509	0.7469
Canny					
EDGE [39]	0.6445	0.6980	0.5262	0.7046	0.6433
EDGE+R [38]	0.7088	0.6953	**0.7282**	**0.9383**	**0.7677**
CIR(2)+R	**0.7122**	0.6933	0.7217	0.9313	0.7646
CIR(2)+E+R	0.7120	**0.7116**	0.6910	0.9220	0.7592
HED					
EDGE+R	0.6883	0.7008	**0.7411**	0.9373	0.7669
CIR(2)+R	**0.7106**	0.6960	0.7381	**0.9389**	**0.7709**
CIR(2)+E+R	0.6981	**0.7303**	0.7228	0.9293	0.7701

Table 1. Comparison of representations on PACS. The best method in each group in **bold**; the best per column also underlined. We focus on the domains shown in blue. The best method per group is consistent for the first three domains across both Canny and HED, and *shape consistently outperforms appearance except on photos*.

Method / Target	Cartoon	Sketch	Painting	Photo	Avg
GRADREV [16]	0.7078	0.6534	0.6981	0.9387	0.7495
JIGSAW [8]	0.7061	0.6402	0.7264	0.9545	0.7568
EPISODIC: R [29]	**0.7108**	0.6936	0.7798	**0.9042**	0.7721
EPIS: CIR(2)+R	**0.7108**	**0.7218**	**0.7803**	0.8814	**0.7736**
MASF: R [15]	0.7092	0.6543	**0.7939**	**0.9291**	0.7716
MASF: CIR(2)+R	**0.7443**	**0.7082**	0.7510	0.9178	**0.7803**

Table 2. Comparison against domain generalization baselines on PACS. The first three and the fifth methods use a ResNet-18 representation, while others use our CIR(2)+R. The better result in each of the last two groups is shown in bold. *Combining DG methods with shape boosts results* on cartoons, sketches, and on average.

4.2. Results: PACS

Shape improves upon appearance-only: In Table 1, we observe that the two shape methods we described in Sec. 3, outperform the DEEP ALL baseline by about 7.5% on Sketch, 0.5-1% on Cartoon, and 0.5-1.5% on Painting, using Canny. As observed in other domain generalization literature, DEEP ALL is the strongest method on Photo, but note that classification on photos is *not the goal of domain generalization, because large photo datasets are much easier to find* than cartoon/sketch/painting datasets. Our new MAT-based filters CIR(2)+R are strongest on Cartoons, and the combination CIR(2)+E+R is strongest on Sketch, while EDGE+R is strongest on Painting. In other words, on the two most challenging domains (cartoon and sketch, with weakest performance by DEEP ALL), our novel MAT-based filters are *more helpful than the edge-based method alone: when an image has limited texture, our soft, dense skeleton representation helps provide signal to the subsequent convolutional network.* EDGE alone, without the additional ResNet branch, is much weaker. HED is stronger than Canny on Painting and Photo for most methods, and weaker than Canny for Cartoon, for all methods. This is expected because HED was trained on photo data, and paintings are often photorealistic, but other domains are most distinct. Because our focus is on sparse domains like cartoons and sketches, we only use Canny in the remaining experiments.

Shape boosts DG methods: Table 2 shows the per-

Method / Target	Cartoon	Sketch	Painting	Photo	Avg
DEEP ALL	*0.7598*	0.7378	**0.8130**	**0.9549**	0.8164
EDGE+R	0.7561	0.7561	*0.8084*	0.9481	0.8172
CIR(2)+R	0.7561	*0.7784*	0.8071	*0.9533*	*0.8237*
CIR(2)+E+R	**0.7658**	**0.7841**	0.7990	0.9515	**0.8251**
DEEP ALL					
No augm	0.7031	0.6181	0.7155	0.9509	0.7469
Color jit	0.7117	0.6645	0.7676	0.9611	0.7762
Grayscale	0.7044	0.6122	0.7428	0.9511	0.7526
Crop	0.7302	0.6226	0.7236	0.9511	0.7569
Hor Flip	0.7188	0.6651	0.7392	0.9551	0.7696
Gaussian	0.7275	0.6719	0.7570	0.9545	0.7777
Sobel	0.7282	0.6581	0.7204	0.9525	0.7648
Jigsaw augm	0.7547	0.6841	0.7809	0.9617	0.7954
All augm	0.7598	0.7378	0.8130	0.9549	0.8164
CIR(2)+R					
No augm	0.7122	0.6933	0.7217	0.9313	0.7646
Color jit	0.7349	0.7290	0.7591	0.9575	0.7951
Grayscale	0.7290	0.7021	0.7326	0.9365	0.7751
Crop	0.7398	0.7035	0.7214	0.9491	0.7785
Hor Flip	0.7366	0.7002	0.7278	0.9459	0.7776
Gaussian	0.7477	0.7262	0.7692	0.9487	0.7978
Sobel	0.7302	0.7246	0.7271	0.9373	0.7798
Jigsaw augm	0.7635	0.7277	0.7873	0.9581	0.8092
All augm	0.7561	0.7784	0.8071	0.9533	0.8237

Table 3. The impact of data augmentation. Top: Full augmentation; best result in **bold**, second-best in *italics*. Below: Individual augmentations (light gray, best two augmentations in red) and combinations (dark gray, blue denotes the better combination).

formance of domain generalization methods which seek to statistically bridge domain gaps. All DG methods outperform DEEP ALL from Table 1 on average by varying amounts (0.3% for GRADREV, 1% for JIGSAW, 2.5% for EPISODIC and MASF). EPIS: CIR(2)+R boosts performance on Sketch by **2.8%** compared to EPISODIC: R, and 0.15% on average. MASF: CIR(2)+R boosts MASF: R by **3.5%** on Cartoon, **5.4%** on Sketch, and **0.9%** on average. Note that our numbers for the DG methods are lower than numbers in some of the original papers due to implementation and setting differences; we show data augmentation in Table 3. Small improvements on average are consistent with the original papers, e.g. [8] only reports a 1.5% improvement over DEEP ALL, and in [29], EPISODIC only achieves 1% better results than a DEEP ALL ensemble, and 2.5% better than a single DEEP ALL model.

Benefit of shape persists with data augmentation: In Table 3, we show the effect of data augmentation. At the top, we apply all six augmentations that we described in Sec. 4.1, to the baseline and three shape-based methods. Note that the last two of these augmentations were not used in prior work, but we generally see they boost results (discussed later). We see that the benefit of each shape method over DEEP ALL persists on average, and for the Cartoon and Sketch domains. (If only the standard four Jigsaw augmentations are used, benefits persist for Painting as well). *Our* CIR(2)+E+R *is the strongest method, followed by* CIR(2)+R *and* EDGE+R. In other words, our new MAT representation is most helpful in the data-augmented setting. CIR(2)+E+R achieves *comparable results with several very recent domain generalization methods*, namely [51] (0.8215 on average) and [10] (0.8146) vs 0.8251 for CIR(2)+E+R, *simply through the use of shape.* As shown in Table 2, shape is complementary to statistical domain generalization, so we can further boost [51, 10] by replacing ResNet-18 in these with CIR(2)+R+E.

Impact of different augmentations: At the bottom of Table 3, we show results from individual augmentation strategies (lighter shading), as well as combinations (darker shading), for two methods. We see that the two best individual augmentations (in red) differ by domain. For example, random **crops** are consistently among the best for Cartoons, but not for other domains, and **color jit**tering is primarily helpful for Painting and Photo. Prior work [23] shows that while color jittering, Gaussian blur, and Sobel filtering reduce texture bias of ResNet, random crops *increase* texture bias. Our results put this previous observation in a new light: *data augmentations that reduce texture bias are more helpful than other augmentations in the domain generalization setting*. Yet even the result from crop augmentations improves when using CIR(2)+R rather than DEEP ALL (1% vs 1.4% gain from augmentation, respectively). Compared to the augmentations applied in JIGSAW [8], adding Gaussian and Sobel (resulting in **All augm**) performs much better, and for some domains, boosts performance even beyond recent domain generalization methods, e.g. for Painting and DEEP ALL, the 0.8130 from **All augm** is higher than the 0.8029 in MASF [15], but this is not the case with the original **Jigsaw augm**'s 0.7873. Further, CIR(2)+E+R *boosts results beyond the gain from data augmentation, on average and especially for Sketch* (0.7378 to 0.7841, i.e. 4.6% gain).

Relation to texture/shape bias: We show that *our method works better than mitigating texture bias from prior work* [23], for improving domain robustness. [23] show that Gaussian blur, Sobel filtering, and color jittering, reduce texture bias. In Table 3, we show that our method achieves further gains (1.5-2% stronger) beyond reducing texture bias through these augmentations: compare the Avg column for DEEP ALL with Color jit, Gaussian and Sobel (0.7762, 0.7777, 0.7648) vs CIR(2)+R with the same augmentations (0.7951, 0.7978, 0.7798).

Quality of Canny edges: We next show an ablation (Table 4) where we evaluate the impact of the amount of smoothing applied before edge extraction. Larger values of sigma result in more smoothing, hence coarser edges. We see that the optimal amount of smoothing varies per target domain. Cartoon and Sketch require less smoothing to achieve optimal results than Painting/Photo. This demonstrates quantitatively that Cartoon and Sketch contain coarser/sparser structures, highlights the difference between those two groups of domains, and the importance of devel-

σ / Target	Cartoon	Sketch	Painting	Photo	Avg
CIR(2)+R					
1	0.7170	**0.6929**	0.7078	0.9351	0.7632
2	**0.7291**	0.6901	0.7137	0.9361	**0.7660**
3	0.7132	0.6899	0.7196	**0.9395**	0.7655
4	0.6954	0.6555	**0.7226**	0.9371	0.7527
argmax	2	1	4	3	2
stdev	0.0139	0.0178	0.0066	0.0019	0.0100
EDGE+R					
1	0.7194	**0.7026**	0.7201	0.9407	**0.7707**
2	**0.7211**	0.6882	0.7116	0.9403	0.7653
3	0.7043	0.6841	0.7251	0.9381	0.7629
4	0.7100	0.6372	**0.7353**	**0.9429**	0.7564
argmax	2	1	4	4	1
stdev	0.0079	0.0283	0.0068	0.0020	0.0113

Table 4. Impact of amount of smoothing before edge extraction.

#filt / Target	Cartoon	Sketch	Painting	Photo	Avg
0	0.7088	0.6953	**0.7282**	**0.9383**	0.7677
1	0.7070	**0.7182**	*0.7032*	0.9194	0.7620
2	0.7120	0.7116	0.6910	0.9220	0.7592
4	0.7080	0.7100	0.7018	0.9154	0.7588
8	*0.7152*	*0.7131*	0.7023	*0.9232*	0.7635
16	**0.7157**	0.7097	0.7031	0.9216	0.7625
1st-3rd best	16, 8, 2	1, 8, 2	0, 1, 16	0, 8, 2	

Table 5. Ablation for number of filters used for CIR(N)+E+R.

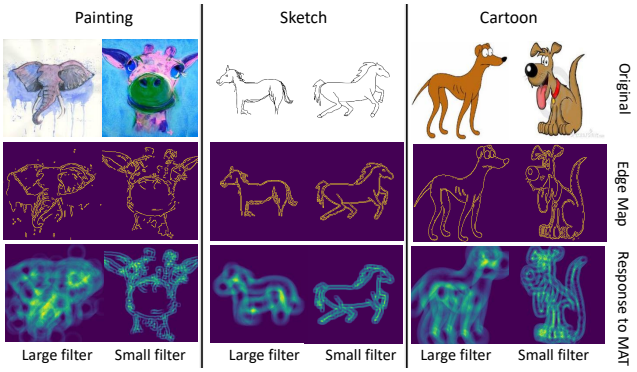

Figure 5. Example images our shape methods classified correctly.

oping methods for sparse images specifically. Other experiments use the OpenCV default of 1.5, which is close to the optimum on average. Finally, CIR(2)+R's standard deviation over values of sigma is lower compared to EDGE+R. *No prior work has extensively tested the importance of the quality of edges for domain generalization.*

Optimal number of shape filters: In Table 5, we show the impact of varying the number of circles n for CIR(N)+E+R, where CIR(0)+E+R = EDGE+R. For Cartoon and Sketch, we improve performance if we add circular filters, but the optimal number of filters varies by domain (16 for Cartoon, 1 for Sketch), although the accuracies themselves are not very different, i.e. stdev is 0.004, 0.008, 0.012, 0.008. We could treat n as a hyperparameter but we find the best n on the val and test sets are very different due to the disjoint domains, thus in all other experiments, we use

Method / Target	Clipart	Art	Product	Real-W	Avg
DEEP ALL [20]	0.4610	**0.5801**	**0.7294**	**0.7550**	0.6314
EDGE+R	0.5000	0.5698	0.7213	0.7520	**0.6358**
CIR(2)+E+R	0.5070	0.5613	0.7150	0.7467	0.6325
CIR(8)+E+R	**0.5117**	0.5620	0.7207	0.7465	0.6352
GRADREV	0.4699	0.5766	**0.7259**	0.7479	0.6301
JIGSAW	0.4581	0.5639	0.7185	0.7422	0.6207
EPISODIC: R	0.4845	**0.5843**	0.7202	0.7482	0.6342
EPIS: CIR(2)+R	**0.5052**	0.5806	0.7157	**0.7489**	**0.6376**

Table 6. Office-Home results. Best method per group (top/bottom) in bold. We focus on Clipart as it contains sparse texture.

Method / Target	Clipart	Quickdraw	Sketch	Avg
DEEP ALL [20]	0.4903	0.1026	0.3396	0.3108
CIR(2)+R	**0.5040**	**0.1039**	**0.3556**	**0.3212**

Table 7. DomainNet results; see text for details.

$n = 2$ as the intersection of the top-3 performers on Cartoon and Sketch (but not absolute best on either domain).

Qualitative result: Fig. 5 shows six images from PACS, correctly classified by our shape methods and incorrectly by JIGSAW. Small filters have high responses in the smaller areas like dog ears in cartoons, horse legs in sketches, and giraffe horn in paintings. Large filters highlight large areas like elephant ears in paintings, and dog and horse torsos in cartoons and sketches.

4.3. Results: Office-Home and DomainNet

Finally, we show results on two additional datasets. In Table 6, we show all OfficeHome domains. Methods perform similarly in this setting, consistent with prior literature (methods within 1% of each other in [8]). The lowest performance is on the sparsely-textured Clipart domain, which is also the one we focus on. CIR(N)+E+R *outperforms all other methods (top part) on Clipart, and all shape methods outperform* DEEP ALL *by a large margin (**4-5%**).* When combining EPISODIC with our shape representation, performance improves for Clipart (**2%**) and on average (0.3%).

In Table 7, we show results on just the sparse domains from DomainNet, using 10,000 images per domain. The reduced data explains low performance, but we see that *shape-based methods outperform the baseline in all cases.*

5. Conclusions and Future Work

We examined the role of shape for improving domain generalization. We tested sensitivity to the quality of edges, and adapted skeletons into a dense representation. These shape representations boost performance in multiple domain generalization tasks. For sparsely-textured domains, skeletons outperform edges. Statistical domain generalization approaches benefit from shape.

Acknowledgements: This work was supported by National Science Foundation Grants No. 2006885 and 1566270, a gift from Google, and a Univ. of Pittsburgh CRDF grant.

References

[1] Vladislav Ayzenberg, Yunxiao Chen, Sami R. Yousif, and Stella F. Lourenco. Skeletal representations of shape in human vision: Evidence for a pruned medial axis model. *Journal of Vision*, 19(6):6–6, 06 2019.

[2] Pedro Ballester and Ricardo Matsumura Araujo. On the performance of googlenet and alexnet applied to sketches. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[3] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (4):509–522, 2002.

[4] Harry Blum et al. *A transformation for extracting new descriptors of shape*, volume 4. MIT press Cambridge, 1967.

[5] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Advances in neural information processing systems (NeurIPS)*, pages 343–351, 2016.

[6] Charles-Olivier Dufresne Camaro, Morteza Rezanejad, Stavros Tsogkas, Kaleem Siddiqi, and Sven Dickinson. Appearance shock grammar for fast medial axis extraction from real images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14382–14391, 2020.

[7] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[8] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019.

[9] Lluis Castrejon, Yusuf Aytar, Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Learning aligned cross-modal representations from weakly aligned data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2940–2949, 2016.

[10] Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. Learning to balance specificity and invariance for in and out of domain generalization. In *European Conference on Computer Vision*, pages 301–318. Springer, 2020.

[11] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018.

[12] Junsuk Choe, Seong Joon Oh, Seungho Lee, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim. Evaluating weakly supervised object localization methods right. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.

[14] Sven Dickinson and Zygmunt Pizlo. *Shape perception in human and computer vision*. Springer, 2015.

[15] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[16] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning (ICML)*, 2015.

[17] 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 *International Conference on Learning Representations (ICLR)*, 2019.

[18] Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris. Spottune: transfer learning through adaptive fine-tuning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4805–4814, 2019.

[19] David Ha and Douglas Eck. A neural representation of sketch drawings. In *International Conference on Learning Representations*, 2018.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[21] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

[22] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations*, 2020.

[23] Katherine Hermann, Ting Chen, and Simon Kornblith. The origins and prevalence of texture bias in convolutional neural networks. *Advances in Neural Information Processing Systems*, 33, 2020.

[24] Zeyi Huang, Haohan Wang, and Eric P Xing. Self-challenging improves cross-domain generalization. In *European Conference on Computer Vision*. Springer, 2020.

[25] Taekyung Kim, Minki Jeong, Seunghyeon Kim, Seokeon Choi, and Changick Kim. Diversify and match: A domain adaptive representation learning paradigm for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12456–12465, 2019.

[26] Kin Chung Kwan, Lok Tsun Sinn, Chu Han, Tien-Tsin Wong, and Chi-Wing Fu. Pyramid of arclength descriptor for generating collage of shapes. *International Conference on Computer Graphics and Interactive Techniques*, 35(6):229, 2016.

[27] Barbara Landau, Linda B Smith, and Susan S Jones. The importance of shape in early lexical learning. *Cognitive development*, 3(3):299–321, 1988.

[28] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[29] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M. Hospedales. Episodic training for domain generalization. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[30] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 136–144, 2016.

[31] Sanjeev Muralikrishnan, Vladimir G Kim, Matthew Fisher, and Siddhartha Chaudhuri. Shape unicode: A unified shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3790–3799, 2019.

[32] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision (ECCV)*, 2020.

[33] Maruthi Narayanan, Vickram Rajendran, and Benjamin Kimia. Shape-biased domain generalization via shock graph embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1315–1325, 2021.

[34] Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2239–2247, 2019.

[35] Kaiyue Pang, Ke Li, Yongxin Yang, Honggang Zhang, Timothy M. Hospedales, Tao Xiang, and Yi-Zhe Song. Generalising fine-grained sketch-based image retrieval. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[36] Kaiyue Pang, Yongxin Yang, Timothy M. Hospedales, Tao Xiang, and Yi-Zhe Song. Solving mixed-modal jigsaw puzzle for fine-grained sketch-based image retrieval. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[37] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1406–1415, 2019.

[38] Filip Radenovic, Giorgos Tolias, and Ondrej Chum. Deep shape matching. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[39] Morteza Rezanejad, Gabriel Downs, John Wilder, Dirk B. Walther, Allan Jepson, Sven Dickinson, and Kaleem Siddiqi. Scene categorization from contours: Medial axis based salience measures. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[40] Leo Sampaio Ferraz Ribeiro, Tu Bui, John Collomosse, and Moacir Ponti. Sketchformer: Transformer-based representation for sketched structure. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[41] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (TOG)*, 35(4):1–12, 2016.

[42] Seonguk Seo, Yumin Suh, Dongwan Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization. In *European Conference on Computer Vision*. Springer, 2020.

[43] Jifei Song, Qian Yu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In *Proceedings of the IEEE international conference on computer vision*, pages 5551–5560, 2017.

[44] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2974–2983, 2018.

[45] Chris Thomas and Adriana Kovashka. Artistic object recognition by unsupervised style adaptation. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2018.

[46] Alexander Toshev, Ben Taskar, and Kostas Daniilidis. Shape-based object detection via boundary structure segmentation. *International Journal of Computer Vision*, 99(2):123–146, 2012.

[47] Stavros Tsogkas and Sven Dickinson. Amat: Medial axis transform for natural images. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[48] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

[49] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[50] Dirk B Walther, Barry Chai, Eamon Caddigan, Diane M Beck, and Li Fei-Fei. Simple line drawings suffice for functional mri decoding of natural scene categories. *Proceedings of the National Academy of Sciences*, 108(23):9661–9666, 2011.

[51] Shujun Wang, Lequan Yu, Caizi Li, Chi-Wing Fu, and Pheng-Ann Heng. Learning from extrinsic and intrinsic supervisions for domain generalization. In *European Conference on Computer Vision*, pages 159–176. Springer, 2020.

[52] Tao Wang, Xiaopeng Zhang, Li Yuan, and Jiashi Feng. Few-shot adaptive faster r-cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7173–7182, 2019.

[53] Yukang Wang, Yongchao Xu, Stavros Tsogkas, Xiang Bai, Sven Dickinson, and Kaleem Siddiqi. Deepflux for skeletons in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5287–5296, 2019.

[54] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[55] Peng Xu, Timothy M Hospedales, Qiyue Yin, Yi-Zhe Song, Tao Xiang, and Liang Wang. Deep learning for free-hand sketch: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[56] Peng Xu, Chaitanya K Joshi, and Xavier Bresson. Multi-graph transformer for free-hand sketch recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[57] Peng Xu, Zeyu Song, Qiyue Yin, Yi-Zhe Song, and Liang Wang. Deep self-supervised representation learning for free-hand sketch. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(4):1503–1513, 2020.

[58] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen-Change Loy. Sketch me that shoe. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 799–807, 2016.

[59] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019.

[60] Kai Zhao, Wei Shen, Shanghua Gao, Dandan Li, and Ming-Ming Cheng. Hi-fi: Hierarchical feature integration for skeleton detection. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2018.

[61] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *European Conference on Computer Vision*, pages 561–578. Springer, 2020.