Learning to Discover Reflection Symmetry via Polar Matching Convolution

Ahyun Seo*   Woohyeon Shim*   Minsu Cho

Pohang University of Science and Technology (POSTECH), South Korea

http://cvlab.postech.ac.kr/research/PMCNet

Abstract

The task of reflection symmetry detection remains challenging due to significant variations and ambiguities of symmetry patterns in the wild. Furthermore, since the local regions are required to match in reflection for detecting a symmetry pattern, it is hard for standard convolutional networks, which are not equivariant to rotation and reflection, to learn the task. To address the issue, we introduce a new convolutional technique, dubbed the polar matching convolution, which leverages a polar feature pooling, a self-similarity encoding, and a systematic kernel design for axes of different angles. The proposed high-dimensional kernel convolution network effectively learns to discover symmetry patterns from real-world images, overcoming the limitations of standard convolution. In addition, we present a new dataset and introduce a self-supervised learning strategy by augmenting the dataset with synthesizing images. Experiments demonstrate that our method outperforms state-of-the-art methods in terms of accuracy and robustness.

1. Introduction

The world is built on symmetry. From the physical structures of nature, the biological patterns of life, to the artifacts of human, symmetries reveal themselves almost everywhere. Perception of such symmetries plays a crucial role at different levels of human vision [38]; it provides humans with pre-attentive cues for early visual analysis and also acts as an integral part for 3D object perception under perspective distortion. Among common groups of symmetry, the most basic and popular form is reflection, mirror, or bilateral symmetry, which is the focus of this work. The task of reflection symmetry detection is to discover reflective patterns from images by detecting their axes of symmetry. Despite the apparent simplicity of the mathematical concept [40] and the long history of research [24], the problem remains challenging due to significant variations and ambiguities of symmetry patterns in the wild.
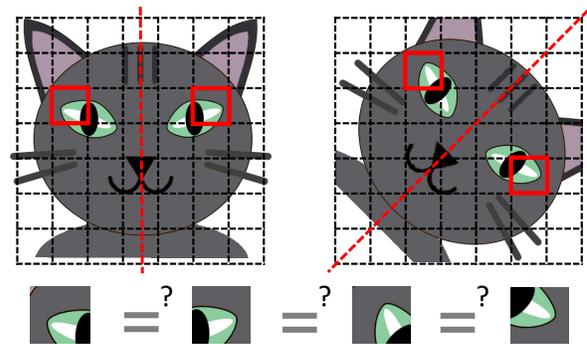

Figure 1. Feature matching in reflection symmetry. Perception of reflection symmetry requires matching features of corresponding regions in reflection with respect to its symmetry axis. Grids represent feature maps, and red squares denote corresponding regions.

One of the most promising directions for tackling the challenge would be to learn from data [15]. While deep convolutional neural networks (CNNs) have made remarkable progress in a wide range of computer vision problems, there has been little work for learning symmetry detection on real-world images. Previous work [14] adapts a CNN [3] to regress a dense heatmap of symmetry axes for symmetry detection. While the results demonstrate the effectiveness of learning, the method does not consider the limitation of conventional CNNs in learning symmetry. As illustrated in Fig. 1, discovering symmetry requires recognizing matching pairs of local regions in reflection, and this may make it hard for CNNs, which are neither invariant nor equivariant to reflection [8, 20], to learn the task. Furthermore, the rotational freedom of the symmetry axis makes it even harder since conventional CNNs are not invariant to rotation either. Standard convolution alone is not effective in learning the required properties solely from data.

In this work, we introduce a new convolutional technique, dubbed the *polar matching convolution* (PMC), which leverages a polar feature pooling, a self-similarity encoding, and a systematic kernel design for axes of different angles. On the basis of symmetrically matched feature pairs in a polar structure, the polar matching kernel computes the confidence of symmetry by exploiting both local similarities

and geometric layouts of local patterns. The proposed convolution neural network learns to discover symmetry patterns with the high-dimensional kernel that effectively compares corresponding features in reflection with respect to the possible candidate axes of symmetry. We also present a new symmetry detection dataset and introduce a self-supervised learning strategy using synthesized images. The experimental evaluation on the SDRW [23] benchmark and our dataset shows that the proposed method, PMCNet, outperforms state-of-the-art methods in accuracy and robustness.

2. Related Work

Reflection symmetry algorithms can be broadly classified into two classes depending on whether the symmetry axes are detected using keypoint matching [4, 7, 26, 28, 39] or heatmap prediction [11, 12, 14, 36].

Keypoint matching. Early work on symmetry detection [4, 26] constructs the symmetry axes using the sparse set of symmetry matches obtained by matching the local features of the keypoints. Keypoint matching is done by comparing the original features and their mirrored counterparts. The descriptors need to be equivariant under reflection on images. Loy and Eklundh [26] use the SIFT [25] descriptor so they can compute the reflected counterpart of the descriptor by reordering its elements. For each pair, a candidate of a symmetry axis is the line perpendicularly passing through the mid-point of the symmetry pair. Patraucean et al. [30] develop a validation scheme using contrario theory on top of the work [26] to find the best mirror-symmetric image patches. Cicconet et al. [6] propose a pairwise voting scheme based on the symmetry coefficient computed with the tangents. Elawady et al. [10] transform the voting problem into the kernel density estimation and handle the displacement and orientation information with a linear-directional kernel-based voting representation. Cho and Lee [4] propose a symmetry-growing method to exploit further information than the local symmetry regions. The initial symmetry feature pairs are used as seeds to be merged and refined with geometric consistency and photometric similarity. The methods [7, 28] group the pixel correspondences using a randomized algorithm and registration. Cicconet et al. [7] propose a Mirror Symmetry via Registration (MSR) framework to perform registration between the original and reflected patterns using normalized cross-correlation matches. They obtain the optimal symmetry plan with the reflection and registration mappings. Cornelius et al. [17] consider detecting reflection symmetries on planar surfaces and locate them using a hough-voting scheme with feature quadruplets. They also improve this framework efficiently with the local affine frame [18] that takes a single symmetry pair to hypothesize the symmetry axis. Sinha et al. [34] adopt a camera coordinate system to

assume planar symmetry and use random sample consensus (RANSAC) algorithm to extract the camera parameters and the representative symmetry. Our convolutional technique, PMC, is inspired by these keypoint matching schemes and designed to detect deformed real-world symmetries in the images.

Heatmap prediction. Other than comparing between features, recently proposed methods directly predict the symmetry for each pixel. Tsogkas et al. [36] assign each pixel a bag of features at all orientations and scales to obtain the symmetry probability map. These symmetry structures are found with multiple instance learning. Nagar et al. [28] leverage PatchMatch [1] to produce dense SymmMap comprising the displacement field to the mirror reflection and the confidence score for each pixel. SymmMap is first initialized with the sparse set of symmetry points and iteratively updated by searching neighbors which maximize the confidence scores. Fukushima et al. [11, 12] introduce a 4-layer neural network rather than using hand-crafted features. The network extracts and blurs the edges from the image to produce a dense heatmap of the symmetry axes.

Funk et al. [14] is the first to use deep CNN to regress the symmetry heatmap directly. This work [14] employs a segmentation-based model [3] and trains with $\ell2$-loss over ground-truth heaptmaps. While the simple CNN model outperforms the methods using the local feature descriptors, it often fails to discover real-world symmetries with deformed axes since conventional CNNs are neither invariant nor equivariant to reflection and rotation. To mitigate these difficulties, we extend PMC to use the self-similarity descriptor for detecting symmetries.

3. Proposed Approach

The proposed method discovers reflection symmetry patterns by learning convolution with high-dimensional kernels for reflective feature matching. Figure 2 briefly illustrates the overall architecture. Given an input image $\mathbf{I}$, a base feature $\mathbf{F}$ is computed by a feature encoder ENC. We transform the base feature $\mathbf{F}$ to a polar self-similarity descriptor $\mathbf{P}$. Then, the polar matching convolutions, $\mathrm{PMC}^{\mathrm{F}}$ and $\mathrm{PMC}^{\mathrm{P}}$, compute the symmetry scores, $\mathbf{S}^{\mathrm{F}}$ and $\mathbf{S}^{\mathrm{P}}$, for the features, $\mathbf{F}$ and $\mathbf{P}$, respectively. The final prediction is obtained by applying a convolutional block $g(\cdot)$ after combining the scores $\mathbf{S}^{\mathrm{F}}$ and $\mathbf{S}^{\mathrm{P}}$, and the base feature $\mathbf{F}$. In the following, we elaborate the polar matching convolutions $\mathrm{PMC}^{\mathrm{F}}$ and $\mathrm{PMC}^{\mathrm{P}}$ and then describe the final output and the training objective for our model, the polar matching convolution network.

3.1. Polar Matching Convolution (PMC)

The convolution operation detects local patterns by multiplying shared kernels with features at different positions.
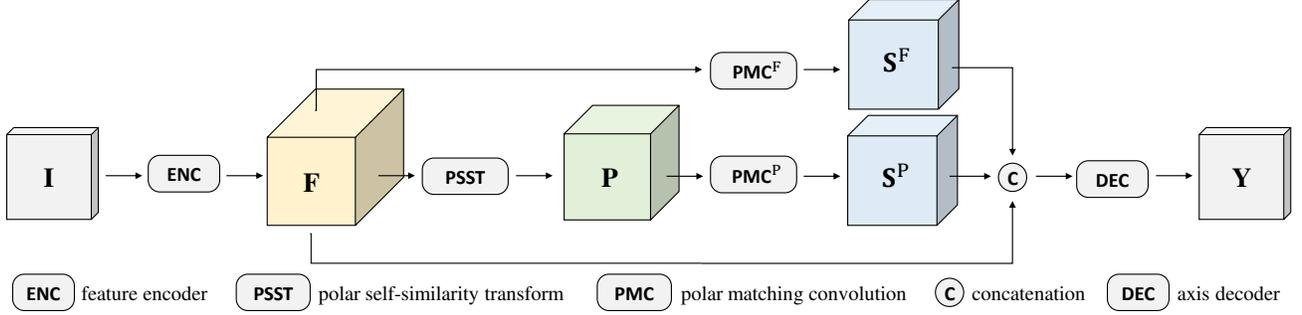
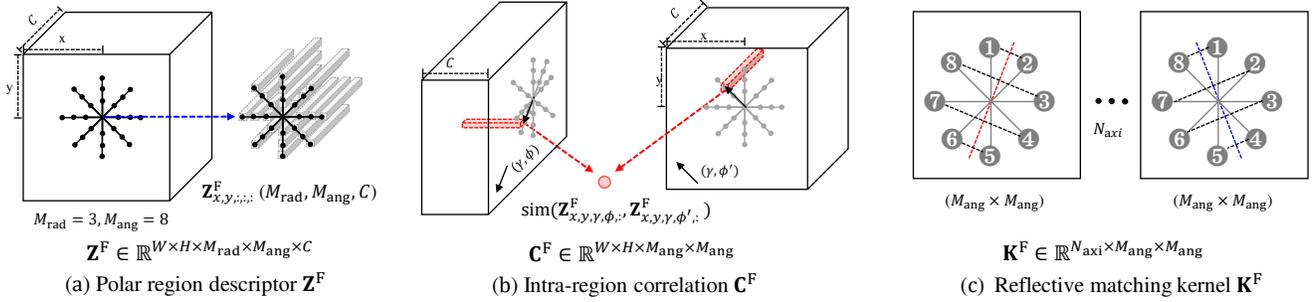Figure 2. Overview of our proposed method. For details, see Sec. 3.

ENC feature encoder   PSST polar self-similarity transform   PMC polar matching convolution   C concatenation   DEC axis decoder

(a) Polar region descriptor $\mathbf{Z}^{\mathrm{F}}$

(b) Intra-region correlation $\mathbf{C}^{\mathrm{F}}$

(c) Reflective matching kernel $\mathbf{K}^{\mathrm{F}}$

Figure 3. Illustration of the Polar Matching Convolution (PMC). (a) The polar region descriptor $\mathbf{Z}^{\mathrm{F}}$ is sampled from the given base feature $\mathbf{F}$ with the number of sampling angles M_{ang} and radii M_{rad}. (b) The intra-region correlation $\mathbf{C}^{\mathrm{F}}$ are computed using the polar region descriptor $\mathbf{Z}^{\mathrm{F}}$. (c) The reflective matching kernel $\mathbf{K}^{\mathrm{F}}$ is applied to $\mathbf{C}^{\mathrm{F}}$ to compute the symmetry score tensor $\mathbf{S}^{\mathrm{F}}$. The matching feature pairs are indicated with black dotted lines. Note that N_{axi} is the number of the candidate axes.

The standard convolution in neural networks is trained to learn the kernels according to the target objective without specific constraints in convolution. For effective symmetry detection, we propose the polar matching convolution (PMC) that is designed to learn to extract symmetry patterns *w.r.t.* the axes of symmetry. The operations of PMC are illustrated in Fig. 3 and detailed in the following.

Polar region descriptor. As a basic unit for local matching, we use a polar region descriptor (Fig. 3a). For each spatial position (x, y) of feature tensor $\mathbf{F} \in \mathbb{R}^{W \times H \times C}$, we collect the features of the neighborhood points sampled with a polar grid centered at (x, y). We set the maximum radius as M_{len}, and the number of sampling angles M_{ang} and radii M_{rad}. A polar region descriptor $\mathbf{Z}^{\mathrm{F}} \in \mathbb{R}^{W \times H \times M_{\mathrm{rad}} \times M_{\mathrm{ang}} \times C}$ is designed to contain the local polar window for all spatial position (x, y) sampled with bilinear interpolation from $\mathbf{F}$:

$$\mathbf{Z}^{\mathrm{F}}_{x,y,\gamma,\phi,c} = \mathbf{F}_{x-\gamma\cos\phi, y+\gamma\sin\phi, c}, \qquad (1)$$

where ϕ indicates the offset angle and γ offset radius from the center of each polar region. The offsets ϕ and γ are evenly distributed in $[0, \frac{M_{\mathrm{len}}}{M_{\mathrm{rad}}}]$ and $[0, \frac{2\pi}{M_{\mathrm{ang}}}]$, respectively.

Intra-region correlation. To capture the symmetry pattern within each polar region, we compute an intra-region

correlation tensor $\mathbf{C}^{\mathrm{F}} \in \mathbb{R}^{W \times H \times M_{\mathrm{ang}} \times M_{\mathrm{ang}}}$ (Fig. 3b) as

$$\mathbf{C}^{\mathrm{F}}_{x,y,\phi,\phi'} = \sum_{\gamma} \mathrm{sim}(\mathbf{Z}^{\mathrm{F}}_{x,y,\gamma,\phi,:}, \mathbf{Z}^{\mathrm{F}}_{x,y,\gamma,\phi',:}), t \qquad (2)$$

to contain the similarities across different angles. For the similarity function, the cosine-similarity is used. The colon (:) represents the whole elements in the specified axis.

Reflective matching kernel. From the intra-region correlation $\mathbf{C}^{\mathrm{F}}$, the symmetry axes are detected by the reflective matching kernel $\mathbf{K}^{\mathrm{F}} \in \mathbb{R}^{N_{\mathrm{axi}} \times M_{\mathrm{ang}} \times M_{\mathrm{ang}}}$ where N_{axi} is the number of the candidate axes. The kernel associates with all the pairs in the polar region by construction. As shown in Fig. 3c, if the region is symmetry with respect to the axis of the red-dotted line, the (unordered) feature pairs $\{(1,2), (3,8), (4,7), (5,6)\}$ should have high correlations. On the other hand, for the axis of the blue-dotted line, the matching feature pairs should be $\{(1,8), (2,7), (3,6), (4,5)\}$. Likewise, we establish a matching set for each candidate axis. In training, we learn the parameters of kernel $\mathbf{K}^{\mathrm{F}}$ involved in the matching sets only and discard the other entries. This results in a kernel with $N_{\mathrm{axi}} M_{\mathrm{ang}}$ trainable parameters in total. The symmetry score $\mathbf{S}^{\mathrm{F}} \in \mathbb{R}^{W \times H \times N_{\mathrm{axi}}}$ is computed by convolution with

$N_{\mathrm{rad}} = 3, N_{\mathrm{ang}} = 8$ $M_{\mathrm{rad}} = 3, M_{\mathrm{ang}} = 8$ $(M_{\mathrm{rad}} \times M_{\mathrm{ang}} \times N_{\mathrm{rad}} \times N_{\mathrm{ang}})$

$\mathbf{F} \in \mathbb{R}^{W \times H \times C}$ $\mathbf{P} \in \mathbb{R}^{W \times H \times N_{\mathrm{rad}} \times N_{\mathrm{ang}}}$ $\mathbf{Z}^{\mathrm{P}} \in \mathbb{R}^{W \times H \times M_{\mathrm{rad}} \times M_{\mathrm{ang}} \times N_{\mathrm{rad}} \times N_{\mathrm{ang}}}$

(a) Polar self-similarity region descriptor $\mathbf{Z}^{\mathrm{P}}$

$(M_{\mathrm{ang}} \times M_{\mathrm{ang}} \times N_{\mathrm{ang}} \times N_{\mathrm{ang}})$ $(M_{\mathrm{ang}} \times M_{\mathrm{ang}})$ $(N_{\mathrm{ang}} \times N_{\mathrm{ang}})$

$\mathbf{K}^{\mathrm{P}} \in \mathbb{R}^{N_{\mathrm{axi}} \times M_{\mathrm{ang}} \times M_{\mathrm{ang}} \times N_{\mathrm{ang}} \times N_{\mathrm{ang}}}$

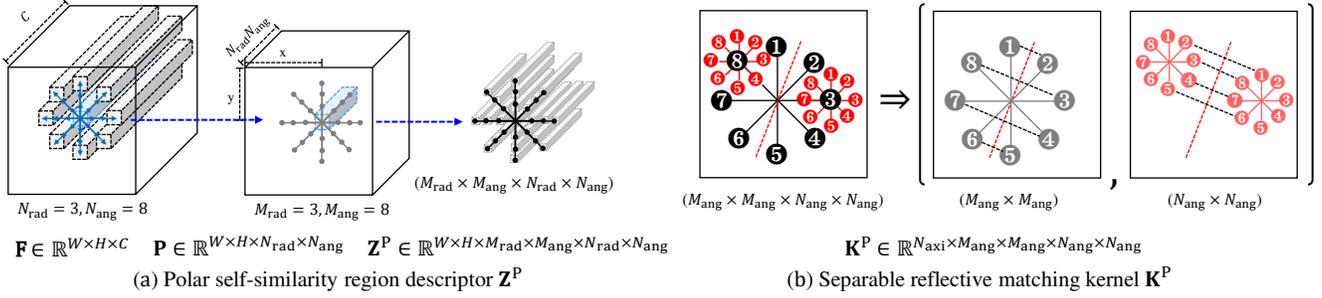(b) Separable reflective matching kernel $\mathbf{K}^{\mathrm{P}}$

Figure 4. Illustration of PMC with self-similarity (PMC$^{\mathrm{P}}$). (a) The polar self-similarity $\mathbf{P}$ contains self-similarity values of the neighborhood pixels sampled with N_{rad} and N_{ang}. The polar region descriptor $\mathbf{Z}^{\mathrm{P}}$ is then sampled from the polar self-similarity $\mathbf{P}$ with the sampling radius M_{rad} and the sampling angle M_{ang}. (b) The reflective matching kernel $\mathbf{K}^{\mathrm{P}}$ extracts the relevant angle-wise relation pairs for N_{axi} candidate axes. The kernel can be decomposed to the lower-dimensional kernels to tackle the relations within the polar region descriptor and the polar self-similarity descriptor.
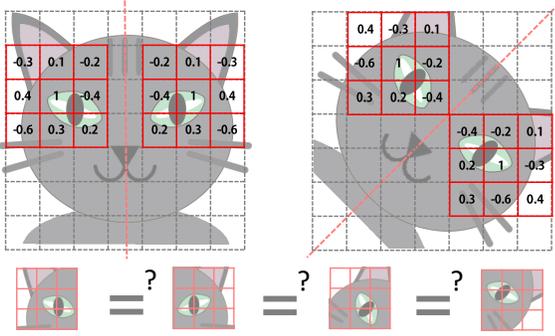

Figure 5. Matching with self-similarity in reflection symmetry. The red grid represents a self-similarity descriptor where each position encodes its similarity value to the center. Between the two corresponding regions in reflection, the self-similarity descriptor is preserved in reflection if each similarity value is invariant to reflection and rotation. See text for details.

the reflective matching kernel as

$$\mathbf{S}^{\mathrm{F}}_{x,y,k} = \sum_{\phi,\phi'} \mathbf{K}^{\mathrm{F}}_{k,\phi,\phi'} \mathbf{C}^{\mathrm{F}}_{x,y,\phi,\phi'}. \qquad (3)$$

3.2. PMC with self-similarity

Since CNNs are neither invariant nor equivariant to rotation and reflection, PMC may still have difficulty in learning to detect symmetry. For example, in the case of Fig. 1, PMC effectively detects reflection if the base feature is entirely invariant for both reflection and rotation. However, it may be an excessive requirement for the base feature extractor, *i.e.*, lower convolutional layers, to achieve. To tackle this issue, we relax the requirement by extending PMC with the self-similarity descriptor [31]. Figure 5 illustrates the basic idea. Let us represent each region as the descriptor of self-similarity values to the neighbor regions. The self-similarity descriptor is preserved in reflection if the pairwise similarity values are invariant to reflection and rotation. In other

words, the region descriptor would be reflection-equivariant if similarity between two regions is invariant to reflection and rotation. Note that this invariance requirement on pairwise similarities is weaker than the original invariance requirement on individual features. To this end, we transform the base feature into self-similarity by using the neighborhood in a polar grid. The details of the extended PMC$^{\mathrm{P}}$ are illustrated in Fig. 4 and described in the following.

Polar self-similarity descriptor. We adopt self-similarity with a polar-shaped local window to encode relational information with varying orientations. Given an base feature $\mathbf{F} \in \mathbb{R}^{W \times H \times C}$, polar self-similarity $\mathbf{P} \in \mathbb{R}^{W \times H \times N_{\mathrm{rad}} \times N_{\mathrm{ang}}}$ is computed as

$$\mathbf{P}_{x,y,r,\theta} = \mathrm{sim}(\mathbf{F}_{x,y,:}, \mathbf{F}_{x-r\cos\theta, y+r\sin\theta, :}), \qquad (4)$$

where x and y indicate the spatial position, and r and θ denote the offsets for a spatial position in polar coordinate. r and θ are evenly distributed between $[0, \frac{N_{\mathrm{len}}}{N_{\mathrm{rad}}}]$ and $[0, \frac{2\pi}{N_{\mathrm{ang}}}]$, respectively, where N_{rad} is the number of sampled radii, N_{ang} the number of angles, and N_{len} the maximum radius. For the similarity function, the rectified cosine-similarity, which is the cosine similarity followed by ReLU. The polar region descriptor $\mathbf{Z}^{\mathrm{P}} \in \mathbb{R}^{W \times H \times M_{\mathrm{rad}} \times M_{\mathrm{ang}} \times N_{\mathrm{rad}} \times N_{\mathrm{ang}}}$ is constructed by sampling the polar self-similarity descriptor $\mathbf{P}$ with bilinear interpolation:

$$\mathbf{Z}^{\mathrm{P}}_{x,y,\gamma,\phi,r,\theta} = \mathbf{P}_{x-\gamma\cos\phi, y+\gamma\sin\phi, r, \theta}. \qquad (5)$$

Note that we preserve the structure of the polar-shaped local windows of $\mathbf{P}$ so that we construct the match *w.r.t.* the polar self-similarity descriptor $\mathbf{P}$.

Intra-region correlation. From the polar region descriptor $\mathbf{Z}^{\mathrm{P}}$, the intra-region correlation tensor $\mathbf{C}^{\mathrm{P}} \in \mathbb{R}^{W \times H \times M_{\mathrm{ang}} \times M_{\mathrm{ang}} \times N_{\mathrm{ang}} \times N_{\mathrm{ang}}}$ is computed as

$$\mathbf{C}^{\mathrm{P}}_{x,y,\phi,\phi',\theta,\theta'} = \sum_{\gamma,r} \mathbf{Z}^{\mathrm{P}}_{x,y,\gamma,\phi,r,\theta} \mathbf{Z}^{\mathrm{P}}_{x,y,\gamma,\phi',r,\theta'}. \qquad (6)$$

The operation above is equivalent to the outer product of N_{ang}-dimensional polar self-similarity vectors in polar region, results in $(N_{\mathrm{ang}} \times N_{\mathrm{ang}})$ element-wise relations of the polar self-similarity descriptor $\mathbf{P}$.

Separable reflective matching kernel. To detect symmetry axes from the intra-region correlation $\mathbf{C}^{\mathrm{P}}$, the reflective matching kernel $\mathbf{K}^{\mathrm{P}} \in \mathbb{R}^{N_{\mathrm{axi}} \times M_{\mathrm{ang}} \times M_{\mathrm{ang}} \times N_{\mathrm{ang}} \times N_{\mathrm{ang}}}$ is required. The symmetry score $\mathbf{S}^{\mathrm{P}} \in \mathbb{R}^{W \times H \times N_{\mathrm{axi}}}$ is computed by convolution with the reflective matching kernel as

$$\mathbf{S}^{\mathrm{P}}_{x,y,k} = \sum_{\phi,\phi',\theta,\theta'} \mathbf{K}^{\mathrm{P}}_{k,\phi,\phi',\theta,\theta'} \mathbf{C}^{\mathrm{P}}_{x,y,\phi,\phi',\theta,\theta'}. \quad (7)$$

Alternatively, the kernel $\mathbf{K}^{\mathrm{P}}$ can be computed by sequentially applying the kernels $\mathbf{K}^{\mathrm{P_M}} \in \mathbb{R}^{N_{\mathrm{axi}} \times M_{\mathrm{ang}} \times M_{\mathrm{ang}}}$ and $\mathbf{K}^{\mathrm{P_N}} \in \mathbb{R}^{N_{\mathrm{axi}} \times N_{\mathrm{ang}} \times N_{\mathrm{ang}}}$ as illustrated in Fig. 4b. If the image is symmetry *w.r.t.* the axis of the red-dotted line, the (unordered) feature pairs $\{(1,2),(3,8),(4,7),(5,6)\}$ and the element pairs $\{(1,2),(3,8),(4,7),(5,6)\}$ should be matched. Different from Sec. 3.1, the matching set of PMC$^{\mathrm{P}}$ emcompasses two levels of matching pairs. When $M_{\mathrm{ang}} = N_{\mathrm{ang}}$, the self-similar pattern enables the kernels $\mathbf{K}^{\mathrm{P_M}}$ and $\mathbf{K}^{\mathrm{P_N}}$ to share the kernel parameters. We denote the shared kernel by $\mathbf{K}^{\mathrm{P_S}} \in \mathbb{R}^{N_{\mathrm{axi}} \times N_{\mathrm{ang}} \times N_{\mathrm{ang}}}$. As in the reflective matching kernel of Sec. 3.1, we only use and train $N_{\mathrm{axi}} N_{\mathrm{ang}}$ elements among $N_{\mathrm{axi}} N_{\mathrm{ang}} N_{\mathrm{ang}}$ parameters of the kernel. We can thus rewrite Eq. (7) as

$$\tilde{\mathbf{C}}^{\mathrm{P}}_{x,y,\theta,\theta'} = \sum_{\phi,\phi'} \mathbf{K}^{\mathrm{P_S}}_{k,\phi,\phi'} \mathbf{C}^{\mathrm{P}}_{x,y,\phi,\phi',\theta,\theta'}, \quad (8)$$

$$\mathbf{S}^{\mathrm{P}}_{x,y,k} = \sum_{\theta,\theta'} \mathbf{K}^{\mathrm{P_S}}_{k,\theta,\theta'} \tilde{\mathbf{C}}^{\mathrm{P}}_{x,y,\theta,\theta'}. \quad (9)$$

3.3. Reflection symmetry detection

We combine $\mathbf{S}^{\mathrm{P}}$, $\mathbf{S}^{\mathrm{F}}$, and $\mathbf{F}$ as the input to the decoder (DEC). To feed the positional confidence to the decoder as an additional input, we construct the maximum score map $\tilde{\mathbf{S}} \in \mathbb{R}^{W \times H \times 1}$ by pooling the maximum score for each spatial position and then replace $\mathbf{S}$ with $[\tilde{\mathbf{S}}||\mathbf{S}]$. The final prediction $\mathbf{Y}$ is the output of the decoder $g(\cdot)$:

$$\mathbf{Y} = g([\mathbf{S}^{\mathrm{P}}||\mathbf{S}^{\mathrm{F}}||\mathbf{F}]), \quad (10)$$

where $||$ denotes the concatenation operation along the channel. The decoder $g(\cdot)$ consists of two blocks of $(3 \times 3 \text{ conv-BN-DO-ReLU})$ and a 1×1 conv layer with sigmoid, where BN and DO denote batch normalization and dropout, respectively.

3.4. Training objective

Our network basically performs pixel-wise binary classification for detection of symmetry axes. Since the symmetry axes occupy only a tiny part of the image, we adopt
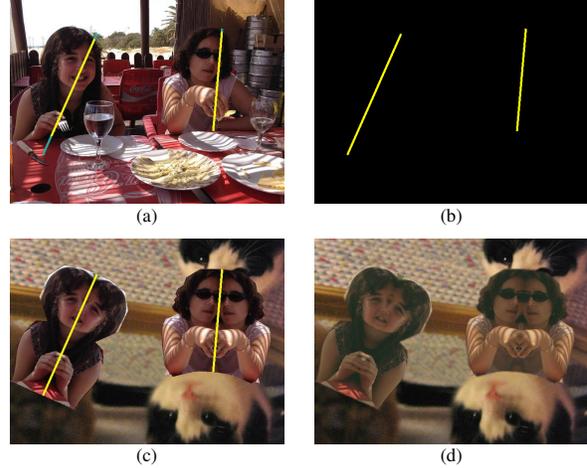

(a) (b)

(c) (d)

Figure 6. An example of a synthesized image. (a) A foreground image with randomly sampled axes. (b) Ground-truth symmetry axes. (c) The result of simple paste of the mirror-flipped foreground instances to the background image. (d) A final training sample with a blending procedure.

α-variant of the focal loss [21] to down-weight the non-axes region in training:

$$\mathcal{L} = \sum_{x,y} -\alpha_t(x,y)(1 - p_t(x,y))^\beta \log(p_t(x,y)), \quad (11)$$

where $\alpha_t(x,y)$ balances the importance of axes/non-axes region and $p_t(x,y)$ is a confidence of symmetry, which adjust the rate between the easy and the hard samples with the focusing parameter β [21]. Moreover, we soft-weight $p_t(x,y)$ and $\alpha_t(x,y)$ after smoothing the corresponding label with gaussian blur with radius 5 following [14],

$$p_t(x,y) = \mathbf{Y}_{x,y}\mathbf{M}_{x,y} + (1 - \mathbf{Y}_{x,y})(1 - \mathbf{M}_{x,y}), \quad (12)$$

$$\alpha_t(x,y) = \alpha\mathbf{M}_{x,y} + (1 - \alpha)(1 - \mathbf{M}_{x,y}), \quad (13)$$

where $\mathbf{Y}$ and $\mathbf{M}$ are the predicted heatmap and the ground-truth (GT) label heatmap and α is a pre-specified scalar that typically set by inverse class frequency.

4. Experiments

4.1. Datasets

Existing datasets. SDRW [23] is used in *Symmetry Detection from Real World Images Competition '13* which contains 51/70 images for training and testing purposes. NYU [5] consists of 176 single-symmetry and 63 multiple-symmetry images. DSW [13] contains 100 images for each single and multiple symmetries and is used in the challenge *Detecting Symmetry in the Wild '17*. Sym-COCO [13] is the dataset built on images selected from COCO [22] containing 250 training and 240 test images for reflection symmetry. BPS [14] is a dataset with 1,202 images collected

from COCO to train the deep neural networks. The test set of Sym-COCO and BPS are the same. Unfortunately, both DSW and BPS datasets are not available at this point. Therefore, we use the two available datasets, SDRW and NYU, and construct a new dataset, LDRS, and its synthetic augmentation, LDRS-*synth*, as follows.

Our dataset (LDRS). We construct the LDRS dataset, which is named after the acronym of our work, following the dataset collection protocol of BPS [14] in terms of the scale and the characteristics of the images. First, we filter the images from COCO [22] dataset by excluding the instances with the area less than $\frac{hw}{16}$ given image height h and width w and discarding the images with less than three instances. Among them, we manually select 1,500 candidate images with the valid reflection symmetries for the train/val split. For the test split, we selected 240 images from the 250 training images of Sym-COCO [13] as they were part of the test split of the COCO dataset. We mainly follow the annotation guidelines of the previous datasets [5, 23, 14] except we do not consider the 3D prior-based or semantic symmetries. Four human annotators labeled the symmetry axes as line segments using the annotation tool labelMe [37]. The train/val/test split contains 1,110/127/240 images, respectively, and all of the images contain at least one ground-truth symmetry axis.

Synthetic augmentation. To overcome the limited real-world training data, we adopt a self-supervised learning strategy of using synthesized images from the existing dataset. We use COCO [22] images and annotations utilized for instance segmentation. The images are generated online during training, and we denote this set of images as LDRS-*synth*. We pre-select the valid foreground images (6,526) while using the whole training split (83,000) for the background. An example is shown in Fig. 6. The synthesis process takes a pair of images, one for the foreground and the other for the background. Given a foreground image and its mask, we select the top-5 instances by area. For each instance, we randomly select two angles ψ_1 and ψ_2 to rotate the masks. We assign vertical axes for each instance and mirror one of the two parts of the target instance. Then, we rotate the image reverse so that each axis results in the angle of $-\psi_1$ and $-\psi_2$. Among ten candidate instances, the combinations with the largest sum of the length of the axes are selected for foreground instances. The instances are blurred with a radius of 5 and matched with the statistics of the background image after normalization for blending. The detailed algorithm is stated in the supplementary material.

4.2. Experimental setup

Training and testing data. For training, we use the training splits of SDRW and LDRS along with the whole sets
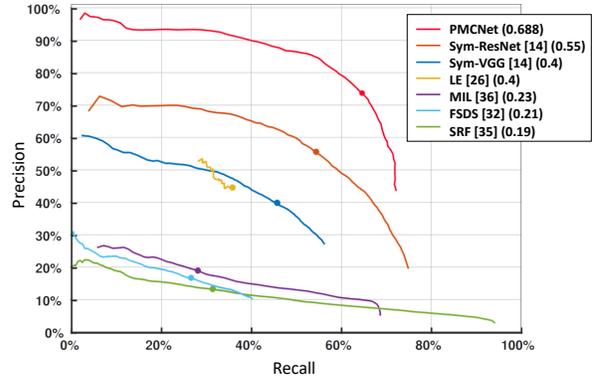

Figure 7. Precision-Recall curve on the SDRW [23] symmetry detection dataset. The (recall, precision) point of the maximum F1-score is indicated with dots.

of NYU and LDRS-*synth*. For evaluation, we use the test splits of SDRW and LDRS.

Evaluation metric. Since the ground-truth axes are pixel-width, we follow the common procedure for evaluating the boundaries by applying the morphological thinning [10, 27] to the output heatmap and performing the pixel matching algorithm between the output and the ground-truth boundaries. We set the maximum distance as 0.01 of the image diagonal for the pixel matching process. We use 100 evenly distributed thresholds to compute the F1 score, the harmonic mean of precision and recall, following [10]. We report the maximum F1 score (%) for the test split of both SDRW and our LDRS datasets for ablation studies.

Implementation details. We employ ResNet-101 [16] pre-trained on ImageNet [9] as the backbone for the base feature $\mathbf{F}$. To aggregate multi-resolution features, we apply the Atrous Spatial Pyramid Pooling module (ASPP) [2]. The dropout [33] rates of two blocks in the decoder $g(\cdot)$ are 0.5 and 0.1, respectively. The hyperparameters for the polar descriptors are set as M_{rad}=5, M_{ang}=8, N_{rad}=5, N_{ang}=8, M_{len}=8, N_{len}=4, and N_{axi}=8. When N_{axi}=8, the 4 axes lie on the polar descriptor, and the other 4 axes lie in between the polar descriptor. The focal loss is applied with the weighting factor α=0.95 and the focusing parameter β=2. The model is trained end-to-end for 100 epochs with an initial learning rate 0.001 using Adam optimizer [19] while the learning rate is decayed by the factor of 0.1 at 50th and 75th epochs. The training batch is randomly composed of 16 images from the different datasets. The images are augmented with random rotation and color jittering. During training, we evaluate our model every 5 epochs in our validation set with 10 thresholds. The input images are resized to (417, 417) for both training and testing, while we resize the output back to its original size for evaluation. We use

descriptor		test dataset	
polar region	polar self-similarity	SDRW	LDRS
✓		65.5	35.3
	✓	66.2	33.7
✓	✓	**68.3**	**35.9**

Table 1. Ablation on descriptors.

kernel		test dataset	
learnable	axis-aware	SDRW	LDRS
	✓	65.4	34.6
✓		67.5	35.3
✓	✓	**68.8**	**37.3**

Table 2. Ablation on kernels.

decoder input		test dataset	
score ($\mathbf{S}^P$, $\mathbf{S}^F$)	base feature ($\mathbf{F}$)	SDRW	LDRS
✓		62.7	32.5
	✓	66.8	36.2
✓	✓	**68.8**	**37.3**

Table 3. Ablation on decoder inputs.

training strategy		test dataset	
pre-training	fine-tuning	SDRW	LDRS
	✓	47.8	25.5
✓		67.2	32.7
✓	✓	**68.8**	**37.3**

Table 4. Ablation on training strategies.

train dataset		test dataset	
LDRS-*synth*	SDRW + LDRS + NYU	SDRW	LDRS
✓		37.6	14.2
	✓	61.6	34.8
✓	✓	**68.8**	**37.3**

Table 5. Ablation on training datasets.

PyTorch [29] framework to implement our model.

4.3. Ablation study

Different types of descriptors. We introduce two types of polar descriptors in Sec. 3.2: the *polar region* descriptor and the *polar self-similarity* descriptor. While the PMC^F extracts the *polar region* descriptor from the base feature, the PMC^P first computes the *polar self-similarity* descriptor using the base feature, then constructs the *polar region* descriptor. To investigate the impact of the proposed polar descriptors, we conduct experiments using decoder input of $(\mathbf{S}||\mathbf{F})$, which contains a single score computed by the corresponding PMC. Without the *polar region* descriptor, the region is a single pixel. As shown in Tab. 1, using both polar descriptors are effective.

Reflective matching kernels. We show the effectiveness of the kernel design in PMC in Tab. 2. If the kernel is not *axis-aware*, we use all parameters consisting of the kernels. If the kernel is not *learnable*, we fix the values and do not update during training. The *learnable* and *axis-aware* kernel results in the best performance. Interestingly, the *axis-aware* kernel increases the max F1 score while using only $\frac{1}{N_{axi}}$ of the total learnable kernel parameters.

Decoder inputs. We compare the different inputs to the decoder of our model in Tab. 3. The *score* denotes the combination of $\mathbf{S}^P$ and $\mathbf{S}^F$ and the *base feature* indicates $\mathbf{F}$. Without the *base feature*, the model scores lower than the final model because of the lack of semantic information from the CNN features. The model with *base feature* becomes a simple segmentation network similar to [14]. Since the *base feature* does not contain the axis-aware pixel matching information, using *score* improves the max F1 score.

Training strategies. The backbone network of our model is initialized with the ImageNet pre-trained weights and fine-tuned during training. Tab. 4 shows comparisons to results (1) w/o pre-training on the ImageNet and (2) w/o

fine-tuning the feature extractor. The performance drops significantly without fine-tuning, but not in the case without pre-training. While adapting the whole network to polar matching kernels is crucial, the pre-trained weights still act as a good initial prior for computing pixel-level similarities.

Training datasets. We investigate the effectiveness of the self-supervised learning strategy of using synthetically augmented images using the different combinations of the train datasets in Tab. 5. For the real-image datasets, we combine SDRW-*train* [23], NYU [5], and LDRS-*train* each consisting of 51, 239 and 1,110 images, respectively. For the augmented images, we use LDRS-*synth*, which is generated using COCO images. Training only with LDRS-*synth* suffers from poor generalization ability, but it is helpful when used together with the real-image datasets as when the amount of the real images is far less than that of the augmented images.

4.4. Comparison with the state-of-the-art methods

We compare our method with the state-of-the-art methods in Fig. 7. Our method achieves the F1 score of 68.8% and outperforms the state-of-the-art method Sym-ResNet [14] with a large margin. Notably, our model and LE [26] show robust performance against dense detection methods [14, 32, 35, 36], which supports our claim that the reflection-equivariance of the feature descriptor is essential in detecting symmetry patterns.

The qualitative results are shown in Fig. 8. We compare our PMCNet with MIL [36] and Sym-ResNet [14] on (a) SDRW [23] and (b) LDRS (Ours) datasets. SDRW exhibit well-defined symmetries while LDRS include deformed real-world symmetries. MIL works well at the simple cases such as Fig. 8a-(i) and Fig. 8b-(i), but it mostly fails with the complicated and textured objects. Sym-ResNet fails by missing the apparent symmetries in Fig. 8a-(i), adding irrelevant regions in Fig. 8b-(i), falsely detecting the nearby relevant regions in Fig. 8a-(ii) and Fig. 8b-(ii). In contrast, PMCNet successfully detects the well-defined symmetries in Fig. 8a-(ii-iv) and the complex cases in Fig. 8b-(ii-iv) where the other methods fail. Furthermore, PM-
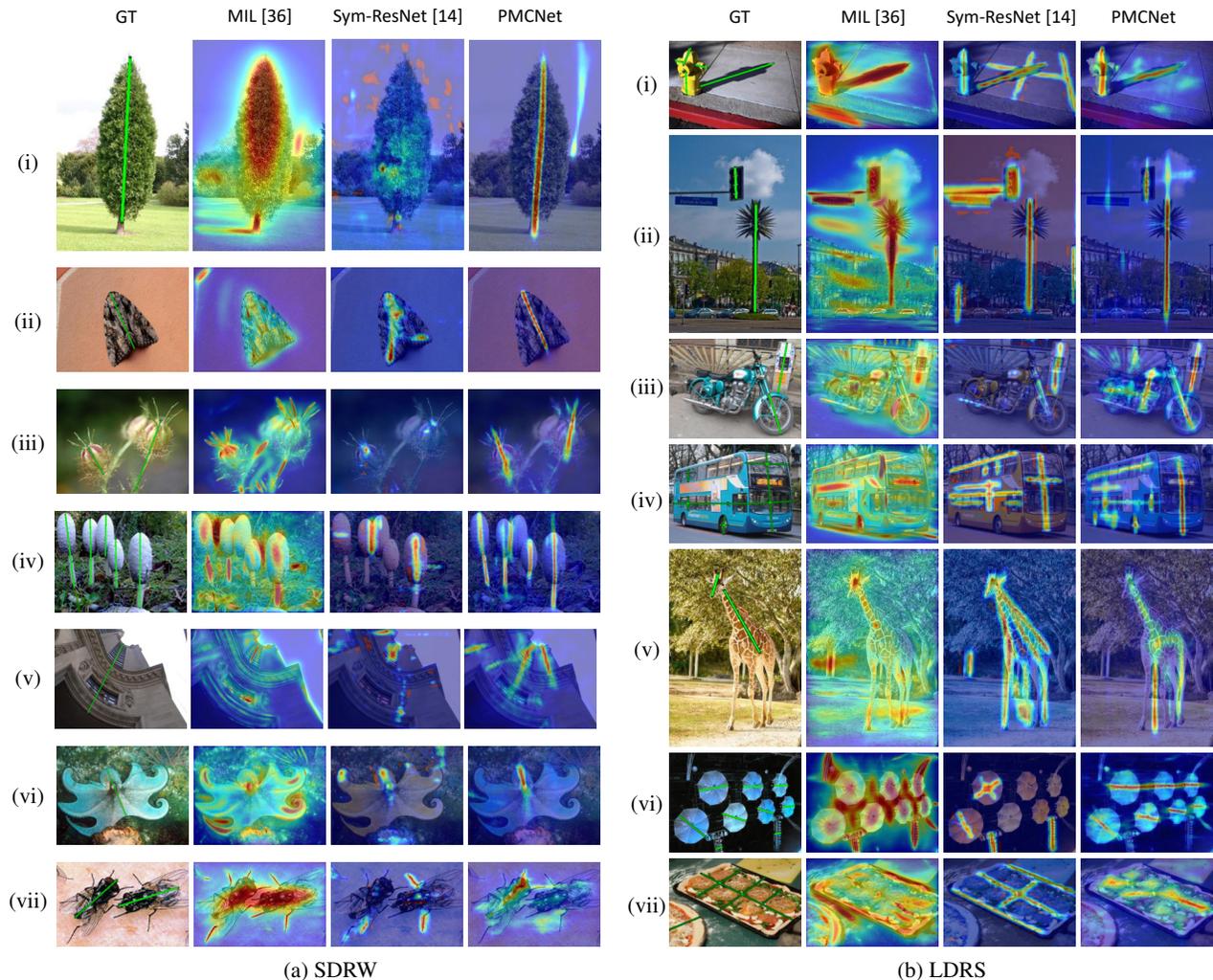
Figure 8. Qualitative results from the test splits of (a) SDRW and (b) LDRS datasets.

CNet even detects symmetries of hierarchical patterns in Fig. 8a-(v) and thin object in Fig. 8b-(v), which were missed in the ground-truth annotations. PMCNet fails at detecting the ambiguous symmetries in Fig. 8a-(vi), multi-angled symmetries in Fig. 8b-(vi), and the symmetries at multiple scales in Fig. 8a-(vii) and Fig. 8b-(vii).

PMCNet remains some limitations. The discrete and sparse bins of the polar descriptors might not fully represent the rotational freedom of the symmetry axis. Also, a single-level PMC with fixed polar window sizes is not capable of detecting the complex multi-scale symmetries. Improving the model considering aforementioned issues and collecting the larger dataset will be interesting for future work.

5. Conclusion

We have introduced the polar matching convolution (PMC) to discover reflection symmetry patterns. It effec-

tively learns to detect symmetry patterns by leveraging the high-dimensional matching kernel with the relational descriptors, achieving the state of the art on the SDRW reflection symmetry benchmark. The ablation studies demonstrate the effectiveness of the components of PMCNet - a polar feature pooling, a self-similarity encoding, and a systematic kernel design for axes of different angles. Further research on this direction can benefit different symmetry detection tasks and a wide range of problems related to symmetries.

References

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 28(3):24, 2009. 2

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. 6

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018. 1, 2

[4] Minsu Cho and Kyoung Mu Lee. Bilateral symmetry detection via symmetry-growing. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 1–11. Citeseer, 2009. 2

[5] M. Cicconet, V. Birodkar, M. Lund, M. Werman, and D. Geiger. A convolutional approach to reflection symmetry, 2016. New York. 5, 6, 7

[6] Marcelo Cicconet, Davi Geiger, Kristin C Gunsalus, and Michael Werman. Mirror symmetry histograms for capturing geometric properties in images. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2981–2986, 2014. 2

[7] Marcelo Cicconet, David GC Hildebrand, and Hunter Elliott. Finding mirror symmetry via registration and optimal symmetric pairwise assignment of curves. In *IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 1749–1758, 2017. 2

[8] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2990–2999. PMLR, 2016. 1

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 6

[10] Mohamed Elawady, Olivier Alata, Christophe Ducottet, Cécile Barat, and Philippe Colantoni. Multiple reflection symmetry detection via linear-directional kernel density estimation. In *Proceedings of the International Conference on Computer Analysis of Images and Patterns (CAIP)*, pages 344–355. Springer, 2017. 2, 6

[11] Kunihiko Fukushima. Use of non-uniform spatial blur for image comparison: Symmetry axis extraction. *Neural Network*, 18(1):23–32, Jan. 2005. 2

[12] Kunihiko Fukushima and Masayuki Kikuchi. Symmetry axis extraction by a neural network. *Neurocomputing*, 69(16):1827–1836, 2006. 2

[13] Christopher Funk, Seungkyu Lee, Martin R Oswald, Stavros Tsogkas, Wei Shen, Andrea Cohen, Sven Dickinson, and Yanxi Liu. 2017 iccv challenge: Detecting symmetry in the wild. In *IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 1692–1701, 2017. 5, 6

[14] Christopher Funk and Yanxi Liu. Beyond planar symmetry: Modeling human perception of reflection and rotation symmetries in the wild. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 793–803, 2017. 1, 2, 5, 6, 7

[15] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009. 1

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 6

[17] Cornelius Hugo and Loy Gareth. Detecting bilateral symmetry in perspective. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2006. 2

[18] Cornelius Hugo, Perdoch Michal, Matas Jiri, and Loy Gareth. Efficient symmetry detection using local affine frames. In *Scandinavian Conference on Image Analysis (SCIA)*, 2007. 2

[19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 6

[20] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2747–2755. PMLR, 2018. 1

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2980–2988, 2017. 5

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 5, 6

[23] Jingchen Liu, George Slota, Gang Zheng, Zhaohui Wu, Minwoo Park, Seungkyu Lee, Ingmar Rauschert, and Yanxi Liu. Symmetry detection from realworld images competition 2013: Summary and results. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 200–205, 2013. 2, 5, 6, 7

[24] Yanxi Liu, Hagit Hel-Or, and Craig S Kaplan. *Computational symmetry in computer vision and computer graphics*. Now publishers Inc, 2010. 1

[25] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004. 2

[26] Gareth Loy and Jan-Olof Eklundh. Detecting symmetry and symmetric constellations of features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 508–521. Springer, 2006. 2, 7

[27] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(5):530–549, 2004. 6

[28] Rajendra Nagar and Shanmuganathan Raman. Symmmap: Estimation of the 2-d reflection symmetry map and its applications. In *IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 1715–1724, 2017. 2

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035. Curran Associates, Inc., 2019. 7

[30] Viorica Patraucean, Rafael Grompone von Gioi, and Maks Ovsjanikov. Detection of mirror-symmetric image patches. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 211–216, 2013. 2

[31] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007. 4

[32] Wei Shen, Kai Zhao, Yuan Jiang, Yan Wang, Zhijiang Zhang, and Xiang Bai. Object skeleton extraction in natural images by fusing scale-associated deep side outputs. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 222–230, 2016. 7

[33] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. 6

[34] Sinha Sudipta N., Ramnath Krishnan, and Szeliski Richard. Detecting and reconstructing 3d mirror symmetric objects. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 586–600. Springer, 2012. 2

[35] Ching L Teo, Cornelia Fermuller, and Yiannis Aloimonos. Detection and segmentation of 2d curved reflection symmetric structures. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1644–1652, 2015. 7

[36] Stavros Tsogkas and Iasonas Kokkinos. Learning-based symmetry detection in natural images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 41–54. Springer, 2012. 2, 7

[37] Kentaro Wada. labelme: Image Polygonal Annotation with Python. https://github.com/wkentaro/labelme, 2016. 6

[38] Johan Wagemans. Detection of visual symmetries. *Spatial vision*, 9(1):9–32, 1995. 1

[39] Zhaozhong Wang, Zesheng Tang, and Xiao Zhang. Reflection symmetry detection using locally affine invariant edge correspondence. *IEEE Transactions on Image Processing (TIP)*, 24(4):1297–1301, 2015. 2

[40] Hermann Weyl. Symmetry princeton university press. *Princeton, New Jersey*, page 17, 1952. 1