

# SRN: Side-output Residual Network for Object Symmetry Detection in the Wild

Wei Ke<sup>\*1,2</sup>, Jie Chen<sup>2</sup>, Jianbin Jiao<sup>1</sup>, Guoying Zhao<sup>2</sup> and Qixiang Ye<sup>†1</sup>

<sup>1</sup>University of Chinese Academy of Sciences, Beijing, China

<sup>2</sup>CMVS, University of Oulu, Finland

{kewei11}@mailsucas.ac.cn, {jiechen, gyzhao}@ee.oulu.fi, {jiaojb, qxxy}@ucas.ac.cn

## Abstract

In this paper, we establish a baseline for object symmetry detection in complex backgrounds by presenting a new benchmark and an end-to-end deep learning approach, opening up a promising direction for symmetry detection in the wild. The new benchmark, named Sym-PASCAL, spans challenges including object diversity, multi-objects, part-invisibility, and various complex backgrounds that are far beyond those in existing datasets. The proposed symmetry detection approach, named Side-output Residual Network (SRN), leverages output Residual Units (RUs) to fit the errors between the object symmetry ground-truth and the outputs of RUs. By stacking RUs in a deep-to-shallow manner, SRN exploits the ‘flow’ of errors among multiple scales to ease the problems of fitting complex outputs with limited layers, suppressing the complex backgrounds, and effectively matching object symmetry of different scales. Experimental results validate both the benchmark and its challenging aspects related to real-world images, and the state-of-the-art performance of our symmetry detection approach. The benchmark and the code for SRN are publicly available at <https://github.com/KevinKecc/SRN>.

## 1. Introduction

Symmetry is pervasive in visual objects, both in nature creatures like trees and birds, and artificial objects like aircrafts and oil pipes in aerial images. Symmetric parts and their connections constitute a powerful part-based decomposition of shapes [19, 25], providing valuable cue for the task of object recognition. With symmetry constrained, the performance of image segmentation [24], foreground extraction [5], object proposal [10], and text-line detection [29] could be significantly improved.

The early symmetry detection, named skeleton extraction, usually involves only binary images [8, 18]. In recent

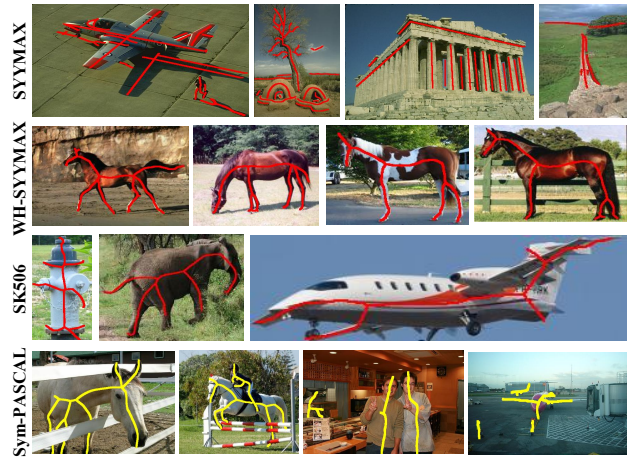


Figure 1: We propose a new benchmark, named Sym-PASCAL, for object symmetry detection in the wild. Compared with SYMMAX [26], WH-SYMMAX [21], and SK506 [22], our Sym-PASCAL spans challenges including object diversity, multi-objects, part-invisibility and various complex backgrounds. (Best viewed in color)

years, symmetry detection tends to process color images [13, 14], but still limited to cropped image patches with little background. This limitation is partially due to the lack of fundamental benchmarks, considering that most existing symmetry detection datasets, *e.g.*, SYMMAX [26], WH-SYMMAX [21], and SK506 [22], lack either object-level annotation or the in-the-wild settings, *i.e.*, multi-objects, part-invisibility, and various complex backgrounds.

In this paper, we present a new challenging benchmark with complex backgrounds, and an end-to-end deep symmetry detection approach that processes in-the-wild images, and target at opening up a promising direction for practical applications of symmetry. The new benchmark, named Sym-PASCAL, is composed of 1453 natural images with 1742 objects derived from the PASCAL-VOC-2011 [4] segmentation dataset. Such a benchmark is more close to practical applications with challenges far beyond

<sup>\*</sup>This work was supported in part by the CSC, China.

<sup>†</sup>Corresponding author

those in existing datasets: (1) *diversity of objects*: multi-class objects with different illuminations and viewpoints; (2) *multi-object co-occurrence*: multiple objects exist in a single image; (3) *part-invisibility*: objects are partially occluded; and (4) *complex backgrounds*: the scenes where object located could be contextually cluttered.

For the in-the-wild symmetry detection problem, we explore the deep Side-output Residual Network (SRN) that directly outputs response image about object symmetry. SRN roots in the Holistically-nested Edge Detection (HED) network [28] but updates it by stacking multiple Residual Units (RUs) on the side-outputs. The Residual Unit (RU) is designed to fit the error between the object symmetry ground-truth and the outputs of RUs, which is computationally easier as it pursuits the minimization of residuals among scales rather than only struggles to combine multi-scale features to fit the object symmetry ground-truth. The RU we defined not only significantly improves the performance of SRN, but also solves the learning convergence problem left by the baseline HED method. By stacking multiple RUs in a deep-to-shallow manner, the receptive fields of stacked RUs could adaptively match the scale of symmetry. The contributions of this paper include:

- A new object symmetry benchmark that spans challenges of diversity, multi-objects, part-invisibility, and various complex backgrounds, promoting the symmetry detection research to in-the-wild scenes.
- A Side-output Residual Network that can effectively fit the errors between ground-truth and the outputs of the stacked RUs, enforcing the modeling capability to symmetry in complex backgrounds, achieving state-of-the-art symmetry detection performance in the wild.

## 2. Related Works

For the applicability and beauty, symmetry has attracted much attention in the past decade. The targets of symmetry detection evolve from binary images to color object images, while the symmetry detection approaches update from hand-crafted to learning based.

**Benchmarks:** In the early research, symmetry extraction algorithms are qualitatively evaluated on quite limited binary shapes [8]. Such shapes are selected from the MPEG-7 Shape-1 dataset for subjective observation [2]. Later, Liu *et al.* [13] use very a few real-world images to perform symmetry detection competitions. To be honest, SYMMAX [26] could be regarded as an authentic benchmark that contains hundreds of training/testing images with local symmetry annotation. But the local reflection symmetry it defined mainly focuses on low-level image edges and contours, missing the high-level concept of objects. WH-SYMMAX [21] and SK506 [22] are recently proposed benchmarks with annotation of object skeletons. Nevertheless, WH-SYMMAX is simply com-

posed of side-view horses while SK506 consists objects with little background. Neither of them involves multiple objects in complex backgrounds, leaving a plenty of room for developing new object symmetry benchmarks.

**Methods:** Early symmetry detection methods, also named skeleton extraction [8, 18], are mainly developed for the binary images by leveraging morphological image operations. When processing color images, they usually need a contour extraction or an image segmentation step as pre-processing. Considering that segmentation of in-the-wild images remains a research problem, the integration of color image segmentation and symmetry detection not only increases the complexity but also accumulates the errors.

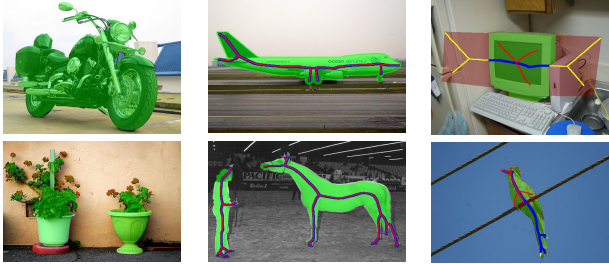
Researchers have tried to extract symmetry in color images based on multi-scale super-pixels. One hypothesis is that the object symmetry axes are the subsets of lines connecting the center points of super-pixels [11]. Such line subsets are explored from the super-pixels using a sequence of deformable disc models extracting the symmetry paths [9]. Their consistence and smoothness are enforced with spatial filters, e.g., a particle filter, which link local skeleton segments into continuous curves [27]. Due to the lack of object prior and the learning module, however, these methods are still limited to handle the images with simple backgrounds.

More effective symmetry detection approaches root in powerful learning methods. On the SYMMAX benchmark, the Multiple Instance Learning (MIL) [26] is used to train a curve symmetry detector with multi-scale and multi-orientation features. To capture diversity of symmetry patterns, Teo *et al.* [24] employ the Structured Random Forest (SRF) and Shen *et al.* [21] use subspace MIL with the same feature. Nevertheless, as the pixel-wise hand-craft feature is computationally expensive and representation limited, these methods are intractable to detect object symmetry in complex backgrounds.

Most recently, a deep learning approach, Fusing Scale-associated Deep Side-outputs (FSDS) [22], is shown to be capable of learning unprecedentedly effective object skeleton representations on WH-SYMMAX [21] and SK506 [22]. FSDS takes the architecture of HED [28] and supervises its side-outputs with scale-associated ground-truth. Despite of its state-of-the-art performance, it needs the intensive annotations of the scales for each skeleton point, which means that it uses much more human effort than other approaches when preparing the training data. Compared with FSDS, our proposed SRN can adaptively match the scales of symmetry, without using scale-level annotation.

## 3. The Sym-PASCAL Benchmark

Symmetry annotation involves pixel-level fine details, and is time consuming. We thus leverage the semantic segmentation ground-truth and a skeleton generation algorithm to aid the annotation of symmetry [20].



(a) Unavailable (b) Available (easy) (c) Available (hard)

Figure 2: Object symmetry annotation. The green masks are annotated semantic segmentation ground-truth. The brown masks are extended from the segmentation. The red lines are the skeletons of semantic segmentation masks. The yellow and blue lines are the skeletons corresponding to the extended masks. The blue lines are the object symmetry ground-truth. (Best viewed in color)

### 3.1. Categorization and Annotation

Sym-PASCAL is derived from the PASCAL-VOC-2011 segmentation dataset [4] which contains 1112 training images and 1111 testing images from 20 object classes including: person, bird, cat, cow, dog, horse, sheep, aero plane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, and tv/monitor.

We categorize the 20 classes of objects into symmetry-available and symmetry-unavailable, Fig. 2. The objects that contain lots of discontinuous parts in the segmentation masks are symmetry-unavailable, specifically potted plant, dining table, motorbike, bicycle, chair and sofa, are not selected, Fig. 2a. The other 14 object classes are symmetry-available. Some of objects are slender and thus easy to annotate, Fig. 2b, and others with small length-width ratio or occlusion are difficult to annotate, Fig. 2c. In total, 648/787 images are selected and annotated from the PASCAL-VOC-2011 training and testing sets. Among these images, 31.3% are with multi-object and 45.6% are with part-invisibility.

For the images where object symmetry is obvious, *i.e.*, objects are composed of slender parts that are easy to annotate, we directly extract symmetry on the object segmentation masks using a skeleton extraction algorithm [20], Fig. 2b. For such objects, the object symmetry (marked with blue curves) and their skeleton (marked with red curves) are consistent. For the images where object symmetry is not obvious, we manually extend the semantic segmentation masks and annotate symmetry on them, Fig. 2c. For wide object as shown on the top of Fig. 2c, we extend the mask along the direction of the long axis of the object and choose the long axis as ground-truth. For occluded objects as shown at the bottom of Fig. 2c, we need to manually fill the missed parts of segmentation masks. For the pictures that contain partial objects, we empirically imagine the occluded parts to extend the segmentation masks. With

these processing above, the skeleton extraction algorithm [20] is used to extract symmetry on the object segmentation masks. The object symmetry ground-truth is set as the skeleton points within the segmentation masks, shown as the blue curves in Fig. 2c.

### 3.2. Discussion

In what follows, we compare the proposed benchmark with three other representative ones, SYMMAX [26], WH-SYMMAX [21], and SK506 [22].

SYMMAX is derived from BSDS300 [1], which contains 200/100 training and testing images. It's annotated with local reflection symmetry on both foreground and background. Considering that most computer vision tasks focus on the foreground, it's more meaningful to use object symmetry instead of the symmetry about the whole image. WH-SYMMAX is developed for object skeletons, but it is made up of only cropped horse images, which are not comprehensive for general object symmetry. SK506 involves skeletons about 16 classes of objects. Nevertheless, their backgrounds are too simple to represent in-the-wild images.

As shown in Tab. 1, the proposed benchmark involves more training and testing images. Particularly, these images involve complex backgrounds, multiple objects and/or occlusions. It is developed for end-to-end object symmetry in-the-wild, providing the protocol to evaluate whether or not an algorithm can detect symmetry without using additional object detectors. In Sym-PASCAL, the images for each class are more balanced than other datasets, Fig. 3b, except that the number of human objects is larger than others. In contrast, in SK506 the objects from different classes have more unbalance, Fig. 3a.

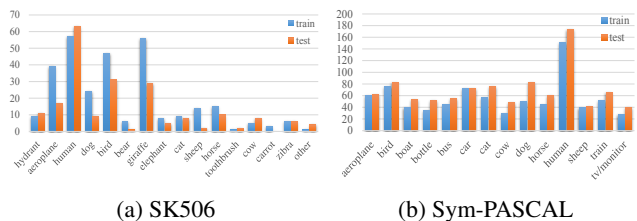


Figure 3: Object-class distributions of the SK506 and Sym-PASCAL datasets.

	SYMMAX	WH-SYMMAX	SK506	Sym-PASCAL
Data type	local symmetry	object skeleton	object skeleton	object symmetry
Image type	in-the-wild image	simple image	simple image	in-the-wild image
#object	–	1	16	14
#training	200	228	300	648
#testing	100	100	206	787

Table 1: Comparison of four symmetry detection datasets.

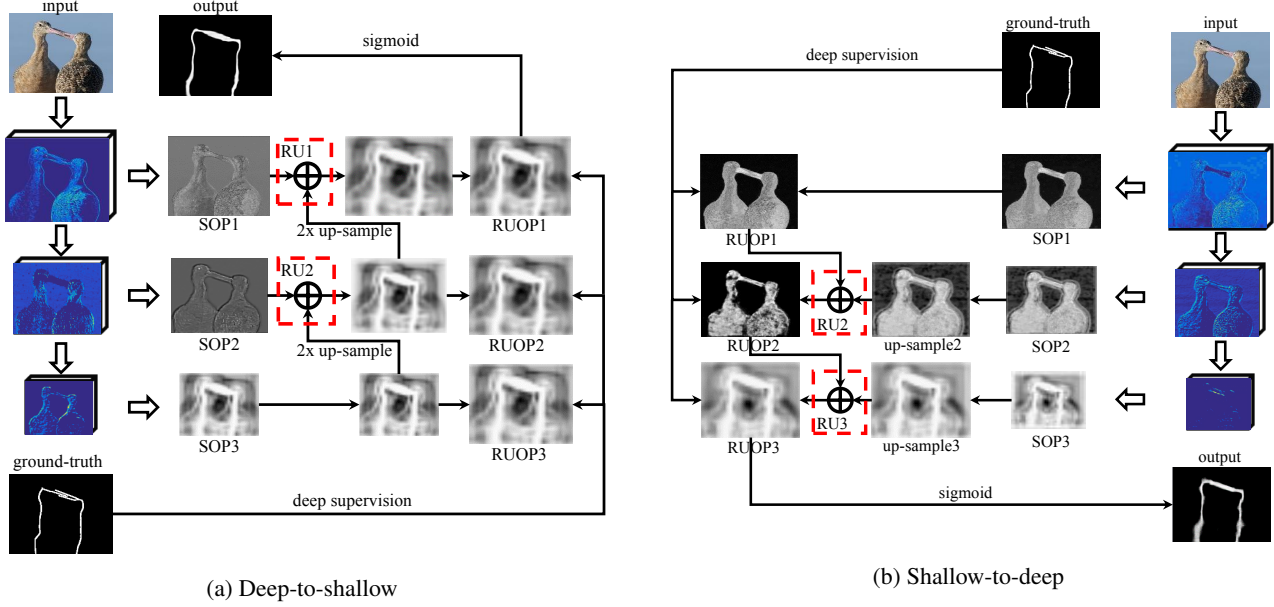


Figure 4: The architectures of proposed Side-Output Residual Network (SRN) by stacking Residual Units (RUs) in (a) deep-to-shallow and (b) shallow-to-deep strategies. The RUs are marked with dashed boxes. With the deep supervision both on the input and output of RU, the residual between the ground-truth and output of RU (RUOP) is computed hierarchically. Along the stacking orientation, the residual decreases so that the RUOP is closer to ground-truth.

## 4. Side-output Residual Network

The proposed Side-output Residual Network (SRN) roots in the well-designed output Residual Unit (RU) and a deep-to-shallow learning strategy. Given the symmetry ground-truth, the SRN is learned in an end-to-end manner.

### 4.1. Output Residual Unit

Given training images, the end-to-end symmetry learning pursuits deep network parameters that best fit the symmetry ground-truth. Such a learning objective is different from that of learning a classification network [7]. The RU defined for output, Fig. 5, is essentially different from that in the residual network defined for features [7]. With the deep supervision both on the input and output of RUs, the residual of the ground-truth is computed. Formally, denoting the input of RU as  $r$  and the additional mapping as  $\mathcal{F}(y)$ , the deep supervision is written as:

$$\begin{cases} r \approx y \\ r + \mathcal{F}(y) \approx y \end{cases}, \quad (1)$$

where  $r$  and  $r + \mathcal{F}(y)$  are the input and output of the RU, respectively.  $\mathcal{F}(y)$  is regarded as the residual estimation of  $y$ . RUs provide shortcut connections between the ground-truth and outputs from different scales, which implies a functional module for the ‘flow’ of errors among different scales, and thus make it easier to fit complex outputs with higher adaptivity. To the extreme, if an input  $r$  is optimal, it would be easier to push the residual to zero than to fit the additional mapping  $\mathcal{F}(y)$ .

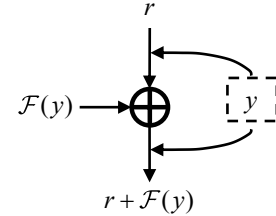


Figure 5: The output Residual Unit (RU). By supervision both on the input and output of RU, the additional mapping  $\mathcal{F}(y)$  estimates the residual of  $y$ .

### 4.2. Network Architectures

By stacking the RUs defined, we implement a kind of new side-output deep network, named Side-output Residual Network (SRN), which incorporates the advantages of both the scale adaptability and residual learning. For SRN, the input of the first RU can be chosen as the shallowest side-output or deepest side-output, which derives two versions of SRN, Fig. 4. In what follows, the RU is numbered as the side-output (SOP) index, and the output of the  $i$ -th RU is denoted as  $RUOP_i$ , for short.

**Deep-to-shallow.** In this SRN architecture, RUs are stacked from deep to shallow, Fig. 4a. Assume that  $s_i$  is the  $i$ -th side-output, and  $r_{i+1}, r_i$  are the input and output of  $i$ -th RU respectively. For the first stacked RU2, the input is set as the deepest SOP3, i.e.,  $r_3 = s_3$ . And SOP2 is used to learn the residual between  $RUOP_3$  and the ground-truth, which updates  $RUOP_3$  to  $RUOP_2$ . The RUs are stacked in order until the shallowest side-output, in other words, the inputs of which are set as the output of the former one. Sigmoid is used as classifier on the output of the last stacked

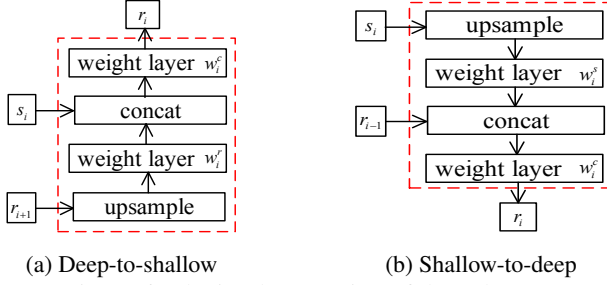


Figure 6: The implementation of the  $i$ -th RU.

RU to generate the final output image.

The implementation of RU in the deep-to-shallow architecture is shown in Fig. 6a. It's noting that the output size of RU in this architecture is same as the side-output rather than the input image. Therefore, a Gaussian deconvolution layer is introduced to the output of RU. As the up-sampling is non-linear transformation, a weight layer is stacked to improve the scale adaptability. Instead of adding up-sampled  $r_{i+1}$  and  $s_i$  directly, a  $1 \times 1$  convolutional layer is utilized to generate  $r_i$ . The RU is formulated,

$$r_i = w_i^c(s_i + w_i^r r_{i+1}), \quad (2)$$

where  $w_i^c, w_i^r$  are the convolutional weights of concatenation layer and the up-sampled  $r_{i+1}$ . With Eqs. (1) and (2), the output residual  $\mathcal{F}_i(y)$  is computed,

$$\mathcal{F}_i(y) = w_i^c \cdot s_i + (w_i^r w_i^c - 1)r_{i+1}. \quad (3)$$

When  $w_i^r \cdot w_i^c$  approximates 1.0, the residual is related to only the side-output. To the extreme, along the stacking orientation of RUs, the residual  $\mathcal{F}(y)$  approximates 0.0.

As we know, the deep layers of CNNs contain features that ignore the image details but capture high-level representations. Therefore, a deep layer SOP3 is expected to be closer to the optimal training solution. RU2 pushes the residual to zero and the response map RUOP2 is similar with the response map RUOP3. In the deep-to-shallow architecture, the deepest side-output is used as a good initialization for the ground-truth, therefore, the deep-to-shallow architecture contributes better results than the shallow-to-deep one, as shown in Sec. 5.2.1.

**Shallow-to-deep.** The architecture is shown in Fig. 4b and the RU in Fig. 6b. The side-outputs are up-sampled by the Gaussian deconvolution layer so that their size is consistent with the input image. Similar with Eq. (3), the residual is computed,

$$\mathcal{F}_i(y) = w_i^s w_i^c \cdot s_i + (w_i^c - 1)r_{i+1}, \quad (4)$$

where  $w_i^s$  is weight parameter of the up-sampled  $s_i$ . Fig. 4b indicates that the shallowest RUOP1 has lots of false positive pixels compare to ground-truth as SOP1 represents local structure of the input image. Along the stacking

orientation, the RU3 reduces the residual so that the outputs of RU3, *i.e.*, RUOP3, are closer to ground-truth compared to RUOP2.

### 4.3. Learning

Given the object symmetry detection training dataset  $S = \{(X_n, Y_n)\}_{n=1}^N$  with  $N$  training pairs, where  $X_n = \{x_j^{(n)}, j = 1, \dots, T\}$  and  $Y_n = \{y_j^{(n)}, j = 1, \dots, T\}$  are the input image and the ground-truth binary image with  $T$  pixels, respectively.  $y_j^{(n)} = 1$  denotes the symmetry pixel and  $y_j^{(n)} = 0$  denotes non-symmetry pixel. We subsequently drop the subscript  $n$  for notational simplicity, since we consider each image independently. We denote  $\mathbf{W}$  as the parameters of the base network. Supposing the network has  $M$  side-outputs, the  $M$ -th side-output is set as the basic output and  $M - 1$  RUs are used. We use the architecture of Fig. 4a as example, in which  $M = 3$  and RUOP3 is the basic output. Fig. 4b has similar formulation. For the basic output, the loss is computed,

$$\begin{aligned} \mathcal{L}_b(\mathbf{W}, w_b) = & -\beta \sum_{j \in Y_+} \log \Pr(y_j = 1 | X; \mathbf{W}, w_b) \\ & - (1 - \beta) \sum_{j \in Y_-} \log \Pr(y_j = 0 | X; \mathbf{W}, w_b), \end{aligned} \quad (5)$$

where  $w_b$  is the classifier parameter for the basic output.  $Y_+$  and  $Y_-$  respectively denote the symmetry and non-symmetry ground-truth label sets. The loss weight  $\beta = |Y_+|/|Y|$ , and  $|Y_+|$  and  $|Y_-|$  denote the symmetry and non-symmetry pixel number, respectively.  $\Pr(y_j = 1 | X; \mathbf{W}, w_b) \in [0, 1]$  is the sigmoid prediction of the basic output that measures how likely the point to be on the symmetry axis. For the  $i$ -th RU,  $i = M - 1, \dots, 1$ , the loss is computed,

$$\begin{aligned} \mathcal{L}_i(\mathbf{W}, \theta_i, w_i) = & -\beta \sum_{j \in Y_+} \log \Pr(y_j = 1 | X; \mathbf{W}, \theta_i, w_i) \\ & - (1 - \beta) \sum_{j \in Y_-} \log \Pr(y_j = 0 | X; \mathbf{W}, \theta_i, w_i) \end{aligned} \quad (6)$$

where  $\theta_i = (w_i^c, w_i^s)$  is the convolutional parameter of the concatenation layers and side-output layers after the  $i$ -th RU.  $w_i$  is the classifier parameter for the output of  $i$ -th RU. The loss function for all the stacked RUs is obtained by

$$\mathcal{L}(\mathbf{W}, \theta, w) = \alpha_M \mathcal{L}_b(\mathbf{W}, w_b) + \sum_{i=M-1}^1 \alpha_i \mathcal{L}_i(\mathbf{W}, \theta_i, w_i). \quad (7)$$

Finally, we obtain the optimal parameters,

$$(\mathbf{W}, \theta, w)^* = \arg \min \mathcal{L}(\mathbf{W}, \theta, w). \quad (8)$$

In the testing phase, giving an image  $X$ , a symmetry prediction map is output by the last stacked RU,

$$\hat{Y} = \Pr(y_j = 1 | X; \mathbf{W}^*, \theta^*, w^*). \quad (9)$$

#### 4.4. Difference to Other Networks

The proposed SRN has significant difference with other end-to-end deep learning implementations, *i.e.*, HED [28], FSDS [22], and Laplacian Reconstruction [6]. In HED, the deep supervision is applied on side-outputs directly, while in SRN the deep supervision is applied on the outputs of RUs. According to (2), each RU contains the information of two side-outputs at least, endowing SRN with the capability to smoothly model the multi-scale symmetry across deep layers. FSDS is an improvement of HED that specifies scales for side-outputs, which requires additional annotation for each scale. In contrast, SRN models the scale information with RUs, without any multi-scale annotations. SRN takes the idea of Laplacian reconstruction that uses a mask to indicate the reconstruction residual for segmentation. The difference lies in that SRN pursues scale adaptability while the Laplacian reconstruction focuses on multi-scale error minimization.

### 5. Experimental results

The proposed SRN is first evaluated and compared on the proposed Sym-PASCAL benchmark. It is then evaluated and compared with the state-of-the-art deep learning approaches on other popular datasets including SYMMAX [26], WH-SYMMAX [21], and SK506 [22].

#### 5.1. Experimental Setup

**Implementation details.** The SRN is implemented following the parameter setting of HED [28], by fine-tuning the pre-trained 16-layer VGG net [23]. The hyper-parameters of SRN include: mini-batch size (1), learning rate (1e-8 for in-the-wild image datasets and 1e-6 for simple image datasets), loss-weight for each RU output (1), momentum (0.9), and initialization of the nested filters (0), weight decay (0.002), and maximum number of training iterations (18,000). In the testing phase, a non-maximal suppression (NMS) algorithm [3] is applied on the output map to obtain object symmetry.

**Evaluation Metrics.** The precision-recall metric with F-measure is used to evaluate the performance of symmetry detection, as introduced in [26]. To obtain the precision-recall curves, the detected symmetry response is first thresholded into a binary map, and then matched with the ground-truth symmetry masks. By changing the threshold value, the precision-recall curve is obtained and the best F-measure is computed.

#### 5.2. Results on Sym-PASCAL

##### 5.2.1 SRN setting

SRN is first evaluated on the new benchmark with different settings, Tab. 2. **Architectures:** Tab. 2 shows that SRN with the deep-to-shallow architecture (F-measure

Architecture	Augmentation	Conv1	F-measure
shallow-deep	1×	with	0.381
		w/o	0.397
	0.8×, 1×, 1.2×	with	0.371
		w/o	0.396
deep-shallow	1×	with	<b>0.443</b>
		w/o	<b>0.443</b>
	0.8×, 1×, 1.2×	with	0.384
		w/o	0.397

Table 2: Performance of SRN under different settings on the Sym-PASCAL benchmark.

0.443) performs significantly better than the shallow-to-deep architecture (F-measure 0.397). It confirms that the deep-to-shallow architecture is easier to reduce the residual than the shallow-to-deep one as the initialization is better.

**Data Augmentation:** Data augmentation can aggregate the training datasets. In this work, image rotation, flipping, up-sampling, and down-sampling (multi-scale) are used for data argumentation. For each scale, we rotate the training images every 90 degree and flip each one with different axis. The performance with/without multi-scale data argumentation is compared. Experiments show that the F-measure decreases with multi-scale augmentation, even though it produces more training data. The reason is analyzed as follows. The symmetry ground-truth is made up of curves with one-pixel thickness. The up-sampling operation produces curves that have thickness larger than one pixel, and the down-sampling operation produces discontinuous symmetry curves. **Conv1:** FSDS [22] doesn't use the conv1 stage of VGG as the size of receptive field is so small (only 5) that introduces local noise of symmetry (too small to capture any symmetry response). The negative impact of small receptive field with SRN is also observed. By pairwise comparison in Tab. 2, the F-measure without conv1 is slightly better than that with conv1.

##### 5.2.2 Performance Comparison

Using the deep-to-shallow SRN with data augmentation but without conv1, we compare the performance of SRN with the state-of-the-art, as shown in Fig. 8 and Tab. 3. All the compared results are generated by running the open source code with default parameter settings.

It's observed that the traditional methods perform poorly and are time consuming. The best F-measure of traditional methods is 0.174, indicating the challenge of the proposed benchmark. Lindeberg [12] runs fastest with 5.79s per frame. Levinshstein [11], MIL [26], Lee [9] and Particle Filter [27] need much more running time for the complex features they used.

The end-to-end deep learning methods perform well. HED gets the F-measure 0.369 and uses only ten milliseconds to process an image. FSDS is degenerated to



Figure 7: Object symmetry detection results on the Sym-PASCAL dataset: the first and second columns for one-object images with/without complex background, the third and fourth columns for multi-object images with/without complex background, and the last two columns for images with occluded objects. (Best viewed in color)

HED when the scale information is not used. Its F-measure reaches 0.418 when slicing and concatenating of each side-output is used. Our proposed SRN gets the best performance with F-measure 0.443 which outperforms the baseline HED approach by 7.4%. It also outperforms the state-of-the-art method, FSDS, by 2.5%.

To show the effectiveness of the end-to-end pipeline in complex backgrounds, we compare the proposed SRN with a two-stage approach composing of semantic segmentation/object detection and skeleton extraction. We choose the best segmentation network FCN-8s [15] to localize objects, and the skeleton method [20] to extract symmetry, getting F-measure 0.386, Fig. 8. We also compare the FSDS [22] on the detection results from the state-of-the-art object detection methods, FasterRCNN [17] and YOLO [16]. As shown in Fig. 8, the F-measures are 0.343 and 0.354, respectively. Experiments results indicate that the proposed end-to-end learning approach is a more effective and efficient way to detect object symmetry than the two-stage approaches.

The object symmetry detection results by the state-of-the-art deep learning approaches are illustrated in Fig. 7. From the first and second columns, it's observed that the object symmetry obtained by our SRN approach in one-object images is more consistent with the ground-truth with/without complex background. The third and fourth columns show examples that contain multiple objects, in which the proposed SRN approach achieves more accurate object symmetry detection results than other approaches. The last two columns show the results of images with occluded objects.

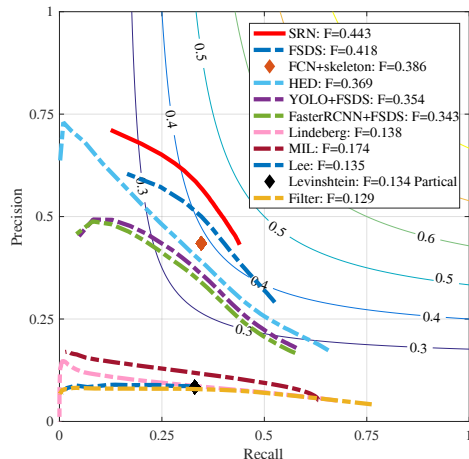


Figure 8: Precision-recall comparison of different approaches on the Sym-PASCAL dataset.

Methods	F-measure	Runtime(s)
Partial Filter [27]	0.129	25.30
Levinshtein [11]	0.134	183.87
Lee [9]	0.135	658.94
Lindeberg [12]	0.138	5.79
MIL [26]	0.174	80.35
HED (baseline) [28]	0.369	<b>0.10</b> †
FSDS [22]	0.418	0.12†
FasterRCNN [17]+FSDS [22]	0.343	0.33†
YOLO [16]+FSDS [22]	0.354	0.12†
FCN [15]+[20]	0.386	0.76†
SRN (ours)	<b>0.443</b>	0.12†

Table 3: Performance comparison of the state-of-the-art approaches on the Sym-PASCAL dataset. †GPU time with NVIDIA Tesla K80

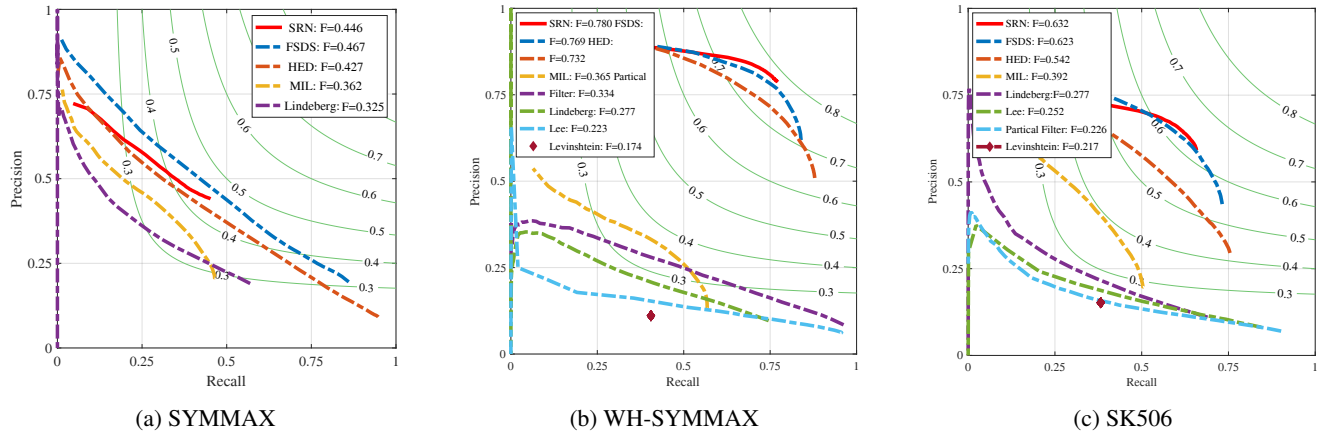


Figure 9: The precision-recall curves of SYMMAX, WH-SYMMAX, and SK506 datasets.

datasets	Levinstein [11]	Lee [9]	Lindeberg [12]	Particle Filter [27]	MIL [26]	HED [28]	FSDS [22]	SRN(ours)
SYMMAX	–	–	0.360	–	0.362	0.427	<b>0.467</b>	0.446
WH-SYMMAX	0.174	0.223	0.277	0.334	0.365	0.732	0.769	<b>0.780</b>
SK506	0.217	0.252	0.227	0.226	0.392	0.542	0.623	<b>0.632</b>

Table 4: Performance comparison of the state-of-the-art approaches on the SYMMAX, WH-SYMMAX, and SK506 dataset.

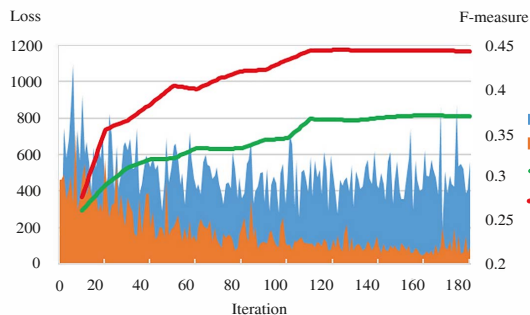


Figure 10: The loss and F-measure comparison of HED and SRN. (Best viewed in color)

### 5.3. Results on Other Datasets

The performances on other three symmetry datasets are shown in Fig. 9 and Tab. 4. Similar with Sym-PASCAL, the deep learning based methods get significantly better performance on all the datasets, especially for the simple image datasets, WH-SYMMAX and SK506. Compared with the baseline HED, the proposed SRN improves the F-measure from 0.427 to 0.446, 0.732 to 0.780, 0.542 to 0.632 on SYMMAX, WH-SYMMAX and SK506, respectively.

### 5.4. Learning Convergence

The learning convergence of the baseline HED and the proposed SRN is shown in Fig. 10. It can be clearly seen that HED has a problem of slow convergence during learning, despite the fact that it achieves good performance on the edge and symmetry detection tasks. The reason could be that the complex backgrounds of input images seriously interrupt the end-to-end (image-to-mask) learning procedure. Benefits from the output residual fitting, the loss

curve of the proposed SRN tends to converge, Fig. 10. In addition, HED needs 12K learning iterations to get the best performance while SRN needs only 3K iterations to get the same performance.

## 6. Conclusion

Symmetry detection has great applicability in computer vision yet remains not being well solved, as indicated by the low performance (often lower than 50%) of the state-of-the-art methods. In this work, we release a new object symmetry benchmark, as well as propose the Side-output Residual Network, establishing a strong baseline for object symmetry detection in the wild. The new benchmark, with challenges related to real-world images, is validated to be a good touchstone of various state-of-the-art approaches. The proposed Side-output Residual Network, with well-defined and stacked Residual Units, is validated to be more effective to perform symmetry detection in complex backgrounds. With the adaptability to object scales, the robustness to complex backgrounds, and the end-to-end learning architecture, the Side-output Residual Network has great potential to process a class of end-to-end (image-to-mask) computer vision tasks.

## Acknowledgement

This work is partially supported by NSFC under Grant 61671427, Beijing Municipal Science and Technology Commission under Grant Z161100001616005, and Science and Technology Innovation Foundation of Chinese Academy of Sciences under Grant CXJJ-16Q218. Tekes, Academy of Finland and Infotech Oulu are also gratefully acknowledged.



## References

- [1] P. Arbelaez, M. Maire, C. C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011. 3
- [2] X. Bai, L. J. Latecki, and W. Liu. Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 29(3):449–462, 2007. 2
- [3] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 37(8):1558–1570, 2015. 6
- [4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. <http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html>. 1, 3
- [5] H. Fu, X. Cao, Z. Tu, and D. Lin. Symmetry constraint for foreground extraction. *IEEE Transaction on Cybernetics*, 44(5):644–654, 2014. 1
- [6] G. Ghiasi and C. C. Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *European Conference on Computer Vision*, pages 519–534, 2016. 6
- [7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *International Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 4
- [8] L. Lam, S. Lee, and C. Y. Suen. Thinning methodologies - A comprehensive survey. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14(9):869–885, 1992. 1, 2
- [9] T. S. H. Lee, S. Fidler, and S. J. Dickinson. Detecting curved symmetric parts using a deformable disc model. In *International Conference on Computer Vision*, pages 1753–1760, 2013. 2, 6, 7, 8
- [10] T. S. H. Lee, S. Fidler, and S. J. Dickinson. Learning to combine mid-level cues for object proposal generation. In *International Conference on Computer Vision*, pages 1680–1688, 2015. 1
- [11] A. Levinshtein, S. J. Dickinson, and C. Sminchisescu. Multiscale symmetric part detection and grouping. In *International Conference on Computer Vision*, pages 2162–2169, 2009. 2, 6, 7, 8
- [12] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):117–156, 1998. 6, 7, 8
- [13] J. Liu, G. Slota, G. Zheng, Z. Wu, M. Park, S. Lee, I. Rauschert, and Y. Liu. Symmetry detection from realworld images competition 2013: Summary and results. In *International Conference on Computer Vision and Pattern Recognition Workshops*, pages 200–205, 2013. 1, 2
- [14] Y. Liu, H. Hel-Or, C. S. Kaplan, and L. J. V. Gool. *Computational Symmetry in Computer Vision and Computer Graphics*, volume 5. 2010. 1
- [15] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *International Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 7
- [16] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *International Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016. 7
- [17] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015. 7
- [18] P. K. Saha, G. Borgefors, and G. S. di Baja. A survey on skeletonization algorithms and their applications. *Pattern Recognition Letters*, 76:3–12, 2016. 1, 2
- [19] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(5):550–571, 2004. 1
- [20] W. Shen, X. Bai, R. Hu, H. Wang, and L. J. Latecki. Skeleton growing and pruning with bending potential ratio. *Pattern Recognition*, 44(2):196–209, 2011. 2, 3, 7
- [21] W. Shen, X. Bai, Z. Hu, and Z. Zhang. Multiple instance subspace learning via partial random projection tree for local reflection symmetry in natural images. *Pattern Recognition*, 52:306–316, 2016. 1, 2, 3, 6
- [22] W. Shen, K. Zhao, Y. Jiang, Y. Wang, Z. Zhang, and X. Bai. Object skeleton extraction in natural images by fusing scale-associated deep side outputs. In *International Conference on Computer Vision and Pattern Recognition*, pages 222–230, 2016. 1, 2, 3, 6, 7, 8
- [23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 6
- [24] C. L. Teo, C. Fermüller, and Y. Aloimonos. Detection and segmentation of 2d curved reflection symmetric structures. In *International Conference on Computer Vision*, pages 1644–1652, 2015. 1, 2
- [25] N. H. Trinh and B. B. Kimia. *Skeleton Search: Category-specific object recognition and segmentation using a skeletal shape model*. *International Journal of Computer Vision*, 94(2):215–240, 2011. 1
- [26] S. Tsogkas and I. Kokkinos. Learning-based symmetry detection in natural images. In *European Conference on Computer Vision*, 2012. 1, 2, 3, 6, 7, 8
- [27] N. Widynski, A. Moevus, and M. Mignotte. Local symmetry detection in natural images using a particle filtering approach. *IEEE Transaction on Image Processing*, 23(12):5309–5322, 2014. 2, 6, 7, 8
- [28] S. Xie and Z. Tu. Holistically-nested edge detection. In *International Conference on Computer Vision*, pages 1395–1403, 2015. 2, 6, 7, 8
- [29] Z. Zhang, W. Shen, C. Yao, and X. Bai. Symmetry-based text line detection in natural scenes. In *International Conference on Computer Vision and Pattern Recognition*, pages 2558–2567, 2015. 1