Object Skeleton Extraction in Natural Images by Fusing Scale-associated Deep Side Outputs

Wei Shen[1], Kai Zhao[1], Yuan Jiang[1], Yan Wang[2], Zhijiang Zhang[1], Xiang Bai[3*]
[1] Key Laboratory of Specialty Fiber Optics and Optical Access Networks, Shanghai University
[2] Rapid-Rich Object Search Lab, Nanyang Technological University
[3] School of Electronic Information and Communications, Huazhong University of Science and Technology
wei.shen@shu.edu.cn,xbai@hust.edu.cn

Abstract

Object skeleton is a useful cue for object detection, complementary to the object contour, as it provides a structural representation to describe the relationship among object parts. While object skeleton extraction in natural images is a very challenging problem, as it requires the extractor to be able to capture both local and global image context to determine the intrinsic scale of each skeleton pixel. Existing methods rely on per-pixel based multi-scale feature computation, which results in difficult modeling and high time consumption. In this paper, we present a fully convolutional network with multiple scale-associated side outputs to address this problem. By observing the relationship between the receptive field sizes of the sequential stages in the network and the skeleton scales they can capture, we introduce a scale-associated side output to each stage. We impose supervision to different stages by guiding the scale-associated side outputs toward groundtruth skeletons of different scales. The responses of the multiple scale-associated side outputs are then fused in a scale-specific way to localize skeleton pixels with multiple scales effectively. Our method achieves promising results on two skeleton extraction datasets, and significantly outperforms other competitors.

1. Introduction

In this paper, we investigate an interesting and nontrivial problem in computer vision, object skeleton extraction from natural images (Fig. 1). Here, the concept of "object" means a standalone thing with a well-defined boundary and center [2], such as an animal, a human, and a plane, as opposed to amorphous background stuff, such as sky, grass, and mountain. Skeleton, also called *symmetry axis*, is a useful structure-based object descriptor. Extracting object skeletons directly from natural images is of broad interests to many real applications including object recognition/detection [4, 28], text recognition [34], road detection and blood vessel detection [27].

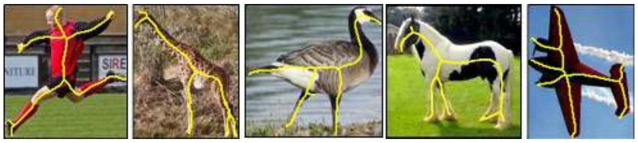

Figure 1. Object skeleton extraction in natural images. The skeletons are in yellow.

Skeleton extraction from pre-segmented images [21] used to be a hot topic, which has been well studied and successfully applied to shape-based object matching and recognition [25, 22, 9]. However, such methods have severe limitations when being applied to natural images, as segmentation from natural images is still an unsolved problem.

Skeleton extraction from natural images is a much more challenging problem. The main difficulties stem from three aspects: (1) Complexity of natural scenes: Natural scenes can be very cluttered. Amorphous background elements, such as fences, bricks and even the shadows of objects, exhibit somewhat self-symmetry, and thus are prone to cause distractions. (2) Diversity of objects: Objects in natural images may exhibit entirely different colors, textures, shapes and sizes. (3) Specificity of skeletons: local skeleton segments have a variety of patterns, such as straight lines, T-junctions and Y-junctions. In addition, a local skeleton segment naturally associates with a certain scale, determined by the thickness of its corresponding object part. However, it is unknown in natural images. We term this problem as unknown-scale problem in skeleton extraction.

A number of works have been proposed to study this problem in the past decade. Broadly speaking, they can be categorized into two groups: (1) Traditional image processing methods [32, 12, 16, 33], which compute skeletons from a gradient intensity map according to some geometric

[*] Corresponding author

constraints between edges and skeletons. Due to the lack of object prior, these methods can not handle the images with complex scenes; (2) Recent learning based methods [29, 15, 14, 27, 30], which learn a per-pixel classification or segment-linking model based on elaborately hand-designed features computed at multi-scales for skeleton extraction. Limited by the ability of traditional learning models and hand-designed features, these methods fail to extract the skeletons of objects with complex structures and cluttered interior textures. In addition, such per-pixel/segment models are usually quite time consuming for prediction. Consequently, there still remains obvious gap between these skeleton extraction methods and human perception, in both performance and speed. Skeleton extraction has its unique aspect by looking into both local and global image context, which requires much more powerful models in both multi-scale feature learning and classifier learning, since the visual complexity increases exponentially with the size of the context field.

To tackle the obstacles mentioned above, we develop a holistically-nested network with multiple scale-associated side outputs for skeleton extraction. The holistically-nested network [31] is a deep fully convolutional network (FCN) [18], which enables holistic image training and prediction for per-pixel tasks. Here, we connect a scale-associated side output to each convolutional layer in the holistically-nested network to address the unknown-scale problem in skeleton extraction.

Referring to Fig. 2, imagine that we are using multiple filters with different sizes (such as the convolutional kernels in convolutional networks) to detect a skeleton pixel with a certain scale; then only the filters with the sizes larger than the scale will have responses on it, and others will not. Note that the sequential convolutional layers in a holistically-nested network can be treated as the filters with increasing sizes (the receptive field sizes on the original image of each convolutional layer are increasing from shallow to deep). So each convolutional layer is only able to capture the features of the skeleton pixels with scales less than its receptive field size. The sequential increasing receptive field sizes provide a principle to quantize the skeleton scale space. With these observations, we propose to impose supervision to each side output, optimizing it towards a scale-associated groundtruth skeleton map. More specifically, each skeleton pixel in it is labeled by a quantized scale value and only the skeleton pixels whose scales are smaller than the receptive filed size of the side output are reserved. Thus, each side output is associated with some certain scales and able to give a certain number of scale-specific skeleton score maps (the score map for one specified quantized scale value) when predicting.

The final predicted skeleton map can be obtained by fusing these scale-associated side outputs. A straightforward fusion method is to average them. However, a skeleton pixel with larger scale probably has a stronger response on a deeper side output, and a weaker response on a shallower side output; a skeleton pixel with smaller scale may have strong responses on both of the two side outputs. By considering this phenomenon, for each quantized scale value, we propose to use a scale-specific weight layer to fuse the corresponding scale-specific skeleton score map provided by each side output.

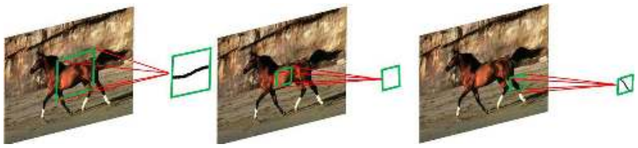

Figure 2. Using filters (the green squares on images) of multiple sizes for skeleton extraction. Only when the size of the filter is larger than the scale of current skeleton part can the filter capture enough context feature to detect it.

In summary, the core contribution of this paper is the proposal of the scale-associated side output layer, which enables both target learning and fusion in a scale-associated way. Therefore, our holistically-nested network is able to localize skeleton pixels with multiple scales.

To the best of our knowledge, there are only two datasets related to our task. One is the SYMMAX300 dataset [29], which is converted from the well-known Berkeley Segmentation Benchmark (BSDS300) [19]. However, this dataset is used for local reflection symmetry detection. Local reflection symmetry [17, 13] is a kind of low-level feature of image, regardless of the concept of "object". Some samples in this dataset are shown in Fig. 3(a). Note that, a large number of symmetries occur in non-object parts. Generally, object skeleton is a subset of local reflection symmetry. The other one is the WH-SYMMAX dataset [23], which is converted from the Weizmann Horse dataset [7]. This dataset is suitable to verify object skeleton extraction methods; however, as shown in Fig. 3(b) the limitation is that only one object category, the horse, is contained in it. To evaluate skeleton extraction methods, we construct a new dataset, named SK506[1]. There are 506 natural images in this dataset, which are selected from the recent published MS COCO dataset [8]. The objects in these 506 images belong to a variety of categories, including humans, animals, such as birds, dogs and giraffes, and artificialities, such as planes and hydrants. We apply a skeletonization method [3] to the provided human-annotated foreground segmentation maps of the selected images to generate the groundtruth skeleton maps. Some samples of the SK506 dataset are shown in Fig. 3(c). We evaluate several skeleton extraction methods as well as symmetry detection methods on both SK506 and WH-SYMMAX. The experimental results

[1] http://wei-shen.weebly.com/uploads/2/3/8/2/23825939/sk506.zip

demonstrate that the proposed method significantly outperforms others.

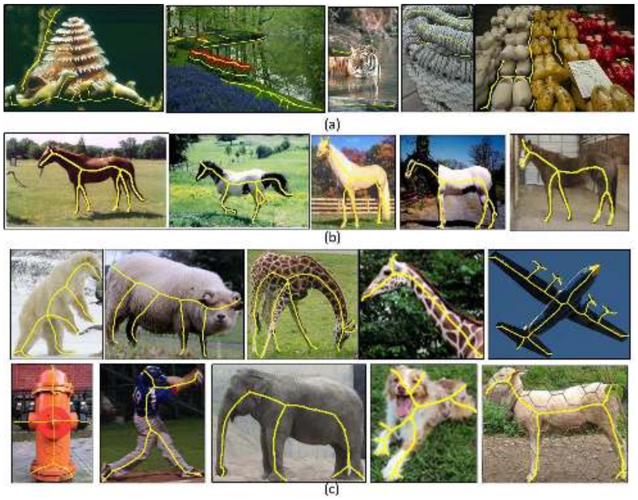

Figure 3. Some samples from three datasets. (a) The SYMMAX300 dataset [29]. (b) The WH-SYMMAX dataset [23]. (c) Our new dataset, the SK506 dataset. The groundtruths for skeleton or local reflection symmetry are in yellow.

2. Related Works

Object skeleton extraction has been paid much attention in previous decades. However, most works in the early stage [21, 3] only focus on skeleton extraction from presegmented images. As these works have a strict assumption that object silhouettes are required to be available, they cannot be applied in our task.

Some pioneers try to extract skeletons from the gradient intensity maps computed on natural images. The gradient intensity map is generally obtained by applying directional derivative operators to a gray-scale image smoothed by a Gaussian kernel. For instance, in [16], the author provides an automatic mechanism to determine the best size of the Gaussian kernel used for gradient computation, and he propose to detect skeletons as the pixels for which the gradient intensity assumes a local maximum (minimum) in the direction of the main principal curvature. Jang and Hong [12] extract the skeleton from the pseudo-distance map which is obtained by iteratively minimizing an object function defined on the gradient intensity map. Yu and Bajaj [32] propose to trace the ridges of the skeleton intensity map calculated from the diffused vector field of the gradient intensity map, which can remove the undesirable biased skeletons. Due to the lack of object prior, these methods are only able to handle the images with simple scenes.

Recent learning based skeleton extraction methods are more suitable to deal with the scene complexity problem in natural images. One type of them formulates skeleton extraction to be a per-pixel classification problem. Tsogkas and Kokkinos [29] compute the hand-designed features of multi-scale and multi-orientation at each pixel, and employ the multiple instance learning framework to determine whether it is symmetry[2] or not. Shen *et al.* [23] then improve their method by training MIL models on automatically learned scale- and orientation-related subspaces. Sironi *et al.* [27] transform the per-pixel classification problem to a regression one to achieve accurate skeleton localization, which learns the distance to the closest skeleton segment in scale-space. Alternatively, another type of learning based methods aim to learn the similarity between local skeleton segments (represented by superpixel [15, 14] or spine model [30]), and link them by hierarchical clustering [15], dynamic programming [14] or particle filter [30]. Due to the limited power of the hand-designed features and traditional learning models, these methods are intractable to detect the skeleton pixels with large scales, as much more context information is needed to be handled.

Our method is inspired by [31], which develops a holistically-nested network for edge detection (HED). Edge detection does not face the unknown-scale problem. Using a local filter to detect an edge pixel, no matter what the size of the filter is, will have responses, either stronger or weaker. So summing up the multi-scale detection responses, which is adopted in the fusion layer in HED, is able to improve the performance of edge detection [20, 10, 24], while bringing noises across the scales for skeleton extraction. There are two main differences between HED and our method. 1. We supervise the side outputs of the network with different scale-associated groundtruths, while the groundtruths in HED are the same. 2. We use different scale-specific weight layers to fuse the corresponding scale-specific skeleton score maps provided by the side outputs, while the side outputs are fused by a single weight layer in HED. Such two changes utilize multi stages in a network to explicitly detect the unknown scale, which HED is unable to handle with. With the extra supervision added to each layer, our method ais able to provide a more informative result, i.e., the predicted scale for each skeleton pixel, which is useful for other potential applications, such as object proposal detection (we will show this in Sec. 4.2.5). While the result of HED cannot be applied to such applications.

3. Methodology

In this section, we describe our methods for object skeleton extraction. First, we introduce the architecture of our holistically-nested network. Then, we discuss how to optimize and fuse the multiple scale-associated side outputs in the network for skeleton extraction.

[2] Although symmetry detection is not the same problem as skeleton extraction, we also compare the methods for it with ours, as skeleton can be considered a subset of symmetry.

3.1. Network Architecture

The recent work [1] has demonstrated that fine-tuning well pre-trained deep neural networks is an efficient way to obtain a good performance on a new task. Therefore, we basically adopt the network architecture used in [31], which is converted from VGG 16-layer net [26] by adding additional side output layers and replacing fully-connected layers by fully-convolutional layers with 1×1 kernel size. Each fully-convolutional layer is then connected to an upsampling layer to ensure that the outputs of all the stages are with the same size. Here, we make several modifications for our task skeleton extraction: (a) we connect the proposed scale-associated side output layer to the last convolutional layer in each stage except for the first one, respectively conv2_2, conv3_3, conv4_3, conv5_3. The receptive field sizes of the scale-associated side output layers are 14, 40, 92, 196, respectively. The reason why we omit the first stage is that the receptive field size of the last convolutional layer in it is too small (only 5) to capture any skeleton features. There are few skeleton pixels with scales less than such a small receptive field size. (b) Each scale-associated side output layer provides a certain number of scale-specific skeleton score maps. Each scale-associated side output layer is connected to a slice layer to obtain the skeleton score map for each scale. Then from all the scale-associated side output layers, we use a scale-specific weight layer to fuse the skeleton score maps for this scale. Such a scale-specific weight layer can be achieved by a fully-convolutional layer with 1×1 kernel size. In this way, the skeleton score maps for different scales are fused by different weight layers. The fused skeleton score maps for each scale are concatenated together to form the final predicted skeleton map. To sum up, our holistically-nested network architecture has 4 stages with additional scale-associated side output layers, with strides 2, 4, 8 and 16, respectively, and with different receptive field sizes; it also has 5 additional weight layers to fuse the side outputs. An illustration for the network architecture is shown in Fig. 4.

3.2. Skeleton Extraction by Fusing Scale-associated Side Outputs

Skeleton extraction can be formulated as a per-pixel classification problem. Given a raw input image $X = \{x_j, j = 1, \ldots, |X|\}$, our goal is to predict its skeleton map $\hat{Y} = \{\hat{y}_j, j = 1, \ldots, |X|\}$, where $\hat{y}_j \in \{0, 1\}$ denotes the predicted label for each pixel x_j, i.e., if x_j is predicted as a skeleton pixel, $\hat{y}_j = 1$; otherwise, $\hat{y}_j = 0$. Next, we describe how to learn and fuse the scale-associated side outputs in the training phase as well as how to utilize the learned network in the testing phase, respectively.

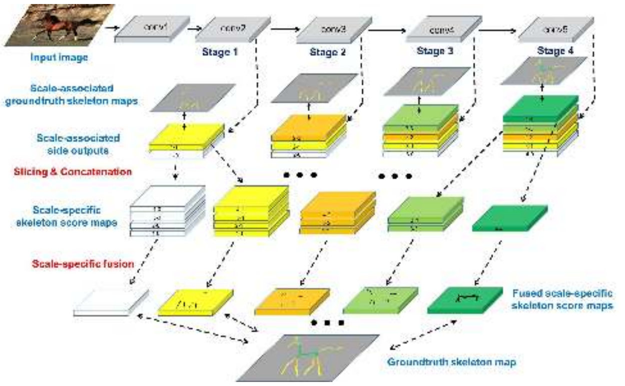
conv1
conv2
conv3
conv4
conv5
Input image
Stage 1
Stage 2
Stage 3
Stage 4
Scale-associated groundtruth skeleton maps
Scale-associated side outputs
Slicing & Concatenation
Scale-specific skeleton score maps
Scale-specific fusion
Fused scale-specific skeleton score maps
Groundtruth skeleton map

Figure 4. The proposed network architecture for skeleton extraction, which is converted from VGG 16-layer net [26]. It has 4 stages with additional scale-associated side output layers connected to the convolutional layers. Each scale-associated side output is guided by a scale-associated groundtruth skeleton map (The skeleton pixels with different quantized scales are in different colors.). Each scale-associated side output layer provides a certain number of scale-specific skeleton score maps (identified by stage number-quantized scale value pairs). The score maps of the same scales from different stages will be sliced and concatenated. Five scale-specific weighted-fusion layers are added to automatically fuse outputs from multiple stages.

3.2.1 Training Phase

We are given a training dataset denoted by $\mathcal{S} = \{(X^{(n)}, Y^{(n)}), n = 1, \ldots, N\}$, where $X^{(n)} = \{x_j^{(n)}, j = 1, \ldots, |X^{(n)}|\}$ is a raw input image and $Y^{(n)} = \{y_j^{(n)}, j = 1, \ldots, |X^{(n)}|\}$ ($y_j^{(n)} \in \{0, 1\}$) is its corresponding groundtruth skeleton map. First, we describe how to compute a quantized skeleton scale map for each training image, which will be used for guiding the network training.

Skeleton scale quantization. According to the definition of skeleton [6], we define the scale of each skeleton pixel as the diameter of the maximal disk centered at it, which can be obtained when computing the groundtruth skeleton map from the pre-segmented image. By defining the scale of each non-skeleton pixel to be zero, we build a scale map $S^{(n)} = \{s_j^{(n)}, j = 1, \ldots, |X^{(n)}|\}$ for each $Y^{(n)}$ and we have $y_j^{(n)} = \mathbf{1}(s_j^{(n)} > 0)$, where $\mathbf{1}(\cdot)$ is an indicator function. As we consider each image holistically, we drop the superscript n in our notation. We aim to learn a holistically-nested network with multiple stages of a convolutional layer linked with a scale-associated side output layer. Assume that there are M such stages in our network, in which the receptive field sizes of the convolutional layers increase sequentially. Let $(r_i; i = 1, \ldots, M)$ be the sequence of the receptive field sizes. Recall that only when the receptive field size is larger than the scale of a skeleton pixel can the convolutional layer capture the features of it. Thus, the scale of a skeleton pixel can be quantized into a discrete value, to indicate which stages in the network are able to detect this skeleton pixel. (Here, we assume that r_M is sufficiently large for capturing the features of the skeleton pixels with the maximum scale). The quantized value z of a scale s is computed by

$$z = \begin{cases} \arg\min_{i=1,\ldots,M} i, \text{ s.t. } r_i > \lambda s & \text{if } s > 0 \\ 0 & \text{if } s = 0 \end{cases}, \quad (1)$$

where $\lambda > 1$ is a factor to ensure that the receptive field sizes are sufficiently large for feature computation. (We set $\lambda = 1.2$ in our experiments.) Now, for an image X, we can build a quantized scale value map $Z = \{z_j, j = 1, \ldots, |X|\}\}(z_j \in \{0, 1, \ldots, M\})$.

Scale-associated side output learning. The groundtruth skeleton map Y can be trivially converted from Z: $Y = \mathbf{1}(Z > 0)$, but not vice versa. So we would like to guild the network training by Z instead of Y, as more supervision can be included. This actually convert a binary classification problem to a multi-class classification, where each class corresponds a quantized scale. Towards this end, each side output layer in our network is associated with a softmax regression classifier. While according to the above discussions, one stage in our network is only able to detect the skeleton pixels with scales less than its corresponding receptive field size. Therefore, the side output is scale-associated. For the i-th side output, we guild it to a scale-associated groundtruth skeleton map: $Z^{(i)} = Z \circ \mathbf{1}(Z \leq i)$, where $\circ$ is an element-wise product operator. Let $K^{(i)}$ denote the maximum value in $Z^{(i)}$, i.e., $K^{(i)} = i$, then we have $Z^{(i)} = \{z_j^{(i)}, j = 1, \ldots, |X|\}, z_j^{(i)} \in \{0, 1, \ldots, K^{(i)}\}$. Let $\ell_s^{(i)}(\mathbf{W}, \mathbf{\Phi}^{(i)})$ denote the loss function for this scale-associated side output, where $\mathbf{W}$ and $\mathbf{\Phi}^{(i)}$ are the layer parameters of the network and the parameters of the classifier of this stage. As our network enables holistic image training, the loss function is computed over all pixels in the training image X and the scale-associated groundtruth skeleton map $Z^{(i)}$. Generally, the distribution of skeleton pixels with different scales and non-skeleton pixels is biased in an image. Therefore, we define a weighted softmax loss function to balance the loss between these multiple classes:

$$\ell_s^{(i)}(\mathbf{W}, \mathbf{\Phi}^{(i)}) = \\ -\frac{1}{|X|} \sum_{j=1}^{|X|} \sum_{k=0}^{K^{(i)}} \beta_k^{(i)} \mathbf{1}(z_j^{(i)} = k) \log \Pr(z_j^{(i)} = k | X; \mathbf{W}, \mathbf{\Phi}^{(i)}), \quad (2)$$

where $\beta_k^{(i)}$ is the loss weight for the k-th class and $\Pr(z_j^{(i)} = k | X; \mathbf{W}, \mathbf{\Phi}^{(i)}) \in [0, 1]$ is the predicted score given by the classifier for how likely the quantized scale of x_j is k. $\mathcal{N}(\cdot)$ denotes the number of non-zero elements in a set, then β_k can be computed by

$$\beta_k^{(i)} = \frac{\frac{1}{\mathcal{N}(\mathbf{1}(Z_i == k))}}{\sum_{k=0}^{K^{(i)}} \frac{1}{\mathcal{N}(\mathbf{1}(Z_i == k))}}. \quad (3)$$

Let $a_{jk}^{(i)}$ be the activation of the i-th side output associated with the quantized scale k for the input x_j, then we use the softmax function [5] $\sigma(\cdot)$ to compute

$$\Pr(z_j^{(i)} = k | X; \mathbf{W}, \mathbf{\Phi}^{(i)}) = \sigma(a_{jk}^{(i)}) = \frac{\exp(a_{jk}^{(i)})}{\sum_{k=0}^{K^{(i)}} \exp(a_{jk}^{(i)})}. \quad (4)$$

One can show that the partial derivation of $\ell_s^{(i)}(\mathbf{W}, \mathbf{\Phi}^{(i)})$ w.r.t. $a_{jl}^{(i)}$ $(l \in \{0, 1, \ldots, K^{(i)}\})$ can be obtained by

$$\frac{\partial \ell_s^{(i)}(\mathbf{W}, \mathbf{\Phi}^{(i)})}{\partial a_{jl}^{(i)}} = -\frac{1}{m}\Bigg(\beta_l^{(i)} \mathbf{1}(z_j^{(i)} = l) - \\ \sum_{k=0}^{K^{(i)}} \beta_k^{(i)} \mathbf{1}(z_j^{(i)} = k) \Pr(z_j^{(i)} = l | X; \mathbf{W}, \mathbf{\Phi}^{(i)})\Bigg). \quad (5)$$

$\mathbf{\Phi} = (\mathbf{\Phi}^{(i)}; i = 1, \ldots, M)$ denotes the parameters of the classifiers in all the stages, then the loss function for all the side outputs is simply obtained by

$$\mathcal{L}_s(\mathbf{W}, \mathbf{\Phi}) = \sum_{i=1}^{M} \ell_s^{(i)}(\mathbf{W}, \mathbf{\Phi}^{(i)}). \quad (6)$$

Multiple scale-associated side outputs fusion. For an input pixel x_j, each scale-associated side output provides a predicted score $\Pr(z_j^{(i)} = k | X; \mathbf{W}, \mathbf{\Phi}^{(i)})$ (if $k \leq K^{(i)}$) for representing how likely its quantized scale is k. We can obtain a fused score f_{jk} by simply summing them with weights $\mathbf{a}_k = (a_k^{(i)}; i = \max(k, 1), \ldots, M)$:

$$f_{jk} = \sum_{i=\max(k,1)}^{M} a_k^{(i)} \Pr(z_j^{(i)} = k | X; \mathbf{W}, \mathbf{\Phi}^{(i)}), \\ \text{s.t.} \sum_{i=\max(k,1)}^{M} a_k^{(i)} = 1. \quad (7)$$

We can understand the above fusion by this way: each scale-associated side output provides a certain number of scale-specific predicted skeleton score maps, and we utilize $M+1$ scale-specific weight layers: $\mathbf{A} = (\mathbf{a}_k; k = 0, \ldots, M)$ to fuse them. Similarly, we can define a fusion loss function by

$$\mathcal{L}_f(\mathbf{W}, \mathbf{\Phi}, \mathbf{A}) = -\frac{1}{|X|}\sum_{j=1}^{|X|}\sum_{k=0}^{M}\beta_k \mathbf{1}(z_j = k)\log \Pr(z_j = k|X; \mathbf{W}, \mathbf{\Phi}, \mathbf{a}_k), \tag{8}$$

where β_k is defined by the same way in Eqn. 3 and $\Pr(z_j = k|X; \mathbf{W}, \mathbf{\Phi}, \mathbf{w}_k) = \sigma(f_{jk})$.

Finally, we can obtain the optimal parameters by

$$(\mathbf{W}, \mathbf{\Phi}, \mathbf{A})* = \arg\min(\mathcal{L}_s(\mathbf{W}, \mathbf{\Phi}) + \mathcal{L}_f(\mathbf{W}, \mathbf{\Phi}, \mathbf{A})). \tag{9}$$

3.2.2 Testing Phase

Given a testing image $X = \{x_j, j = 1, \ldots, |X|\}$, with the learned network $(\mathbf{W}, \mathbf{\Phi}, \mathbf{A})*$, its predicted skeleton map $\hat{Y} = \{\hat{y}_j, j = 1, \ldots, |X|\}$ is obtained by

$$\hat{y}_j = 1 - \Pr(z_j = 0|X; \mathbf{W}*, \mathbf{\Phi}*, \mathbf{a}_0*). \tag{10}$$

Recall that $z_j = 0$ and $z_j > 0$ mean that x_j is a non-skeleton/skeleton pixel, respectively. We refer to our method as FSDS, for fusing scale-associated deep side outputs.

3.3. Understanding of the Proposed Method

To understand our method more deeply, we illustrate the intermediate results of our method and compare with those of HED in Fig. 5. The response of each scale-associated side output can be obtained by the similar way of Eqn. 10. We compare the response of each scale-associated side output to the corresponding one in HED (The side output 1 in HED is connected to conv1_2, while ours start from conv2_2.). With the extra scale-associated supervision, the responses of our side outputs are indeed related to scale. For example, the first one fires on the structure with small scales, such as the legs, the interior textures and the object boundaries; while in the second one, the skeleton parts of the head and neck are clear and meanwhile the noises on small scale structure are suppressed. In addition, we perform scale-specific fusion, by which each fused scale-specific skeleton score map indeed corresponds to one scale (See the first three response maps corresponding to legs, neck and torso respectively). The side outputs in HED are not able to differentiate skeleton pixels with different scales. Consequently, the first two respond on the whole body, which bring noises to the final fusion one.

4. Experimental Results

In this section we discuss the implementation details and compare the performance of our skeleton extraction methods with competitors.

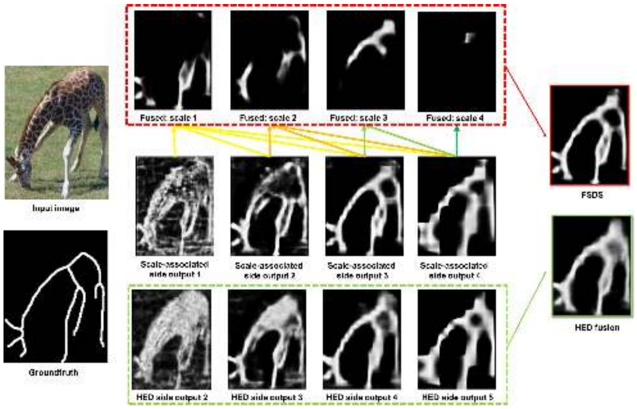

Figure 5. The comparison between the intermediate results of FSDS and HED. We can observe that the former are able to differentiate skeleton pixels with different scales, while the latter cannot.

4.1. Implementation Details

Our architecture is built on the public available implementation of FCN [18] and HED [31]. The whole network is fine-tuned from an initialization with the pre-trained VGG 16-layer net [26].

Model parameters The hyper parameters of our network include: mini-batch size(10), base learning rate (1×10^{-6}), loss weight for each side-output (1), momentum (0.9), initialization of the nested filters(0), initialization of of the scale-specific weighted fusion layer ($1/n$, where n is the number of sliced scale-specific map), the learning rate of the scale-specific weighted fusion layer (5×10^{-6}), weight decay (2×10^{-4}), maximum number of training iterations ($20,000$).

Data augmentation Data augmentation is a principal way to generate sufficient training data for learning a "good" deep network. We rotate the images to 4 different angles (0°, 90°, 180°, 270°) and flip with different axis(up-down,left-right,no flip), then resize images to 3 different scales (0.8, 1.0, 1.2), totally leading to an augmentation factor of 36. Note that when resizing a groundtruth skeleton map, the scales of the skeleton pixels in it should be multiplied by a resize factor accordingly.

4.2. Performance Comparison

We conduct our experiments by comparing our method FSDS with many others, including a traditional image processing method (Lindeberg's method [16]), three learning based segment linking methods (Levinshtein's method [15], Lee's method [14] and Particle Filter [30]), three per-pixel classification/regression methods (Distance Regression [27], MIL [29] and MISL [23]) and a deep learning based method (HED [31]). For all theses methods, we use the source code provided by authors under default setting. For HED and FSDS, we perform iterations of sufficient numbers to obtain optimal models, $15,000$ and $18,000$ iterations for FSDS and HED, respectively. We apply a standard non-maximal suppression algorithm [10] to the response maps of HED and ours to obtain the thinned skeletons for performance evaluation.

4.2.1 Evaluation Protocol

We follow the evaluation protocol used in [29], under which the performances of skeleton extraction methods are measured by their maximum F-meansure ($\frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision}+\text{Recall}}$) as well as precision-recall curves with respect to the groundtruth skeleton map. To obtain the precision-recall curves, the detected symmetry response is first thresholded into a binary map, which is then matched with the groundtruth skeleton map.

4.2.2 SK506

We first conduct our experiments on our newly built SK506 Dataset. Object skeletons in this dataset have large variances in both structures and scales. We split this dataset into 300 training and 206 testing images. We report the F-measure as well as the average runtime per image of each method on this dataset in Table. 1. Observed that, both traditional image processing and per-pixel/segment learning methods perform not well, indicating the difficulty of this task. In addition, the segment linking methods are extremely time consuming. Our method FSDS outperforms others significantly, even compared with the deep learning based method HED. Besides, thanks to the powerful convolution computation ability of GPU, our method can process images in real time, about 20 images per second. The precision/recall curves shown in Fig. 6 evidence again that FSDS is better than the competitors, as ours shows both improved recall and precision at most of the precision-recall regimes. We illustrate the skeleton extraction results obtained by several methods in Fig. 7 for qualitative comparison. These qualitative examples show that our method hits on more groundtruth skeleton points and meanwhile suppresses the false positives. The false positives in the results of HED are probably introduced across response of different scales. Benefited from scale-associated learning and scale-specific fusion, our method is able to suppress such false positives.

4.2.3 WH-SYMMAX

The WH-SYMMAX dataset [23] totally contains 328 images, among which the first 228 are used for training and the rest are used for testing. The precision/recall curves of skeleton extraction methods are shown in Fig. 8 and summary statistics are in Table 2. Qualitative comparisons are given in the Supp.. Both quantitative and qualitative results demonstrate that our method is clearly better than others.

Table 1. Performance comparison between different methods on the SK506 dataset. †GPU time.

Method	F-measure	Avg Runtime (sec)
Lindeberg [16]	0.227	4.03
Levinshtein [15]	0.218	144.77
Lee [14]	0.252	606.30
Particle Filter [30]	0.226	322.25†
MIL [29]	0.392	42.38
HED [31]	0.542	0.05†
FSDS (ours)	**0.623**	**0.05†**

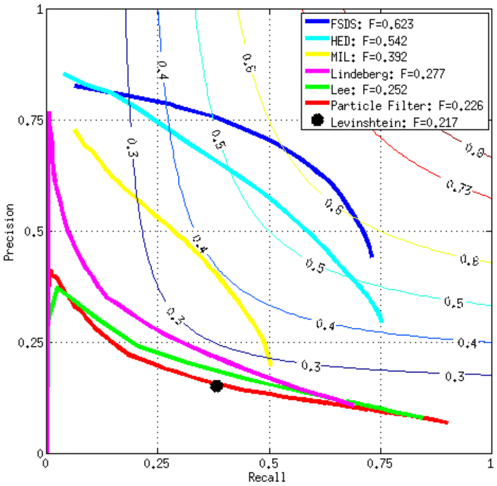

Figure 6. Evaluation of skeleton extractors on the SK506 dataset. Leading skeleton extraction methods are ranked according to their best F-measure with respect to groundtruth skeletons. Our method, FSDS achieves the top result and shows both improved recall and precision at most of the precision-recall regime. See Table 1 for more details about the other two quantities and method citations.

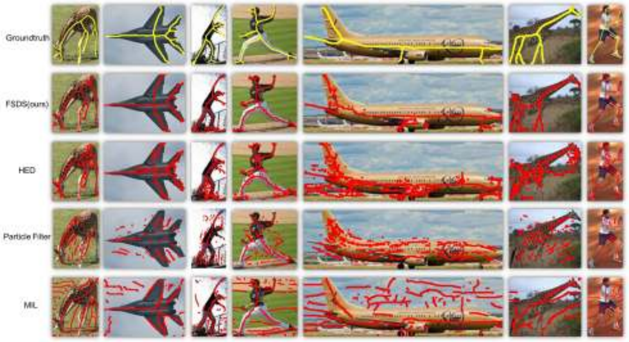

Figure 7. Illustration of skeleton extraction results on the SK506 dataset for several selected images. The groundtruth skeletons are in yellow and the thresholded extraction results are in red. Thresholds were optimized over the whole dataset.

Table 2. Performance comparison between different methods on the WH-SYMMAX Dataset [23]. †GPU time.

Method	F-measure	Avg Runtime (sec)
Lindeberg [16]	0.277	5.75
Levinshtein [15]	0.174	105.51
Lee [14]	0.223	716.18
Particle Filter [30]	0.334	13.9†
Distance Regression [27]	0.103	5.78
MIL [29]	0.365	51.19
MISL [23]	0.402	78.41
HED [31]	0.732	0.06†
FSDS (ours)	**0.769**	0.07†

4.2.4 Cross Dataset Generalization

One may concern the scale-associated side outputs learned from one dataset might lead to higher generalization error when applying them to another dataset. To explore whether this is the case, we test the model learned from one dataset on another one. For comparison, we list the cross dataset generalization results of MIL [29], HED [31] and our method in Table 3. Our method achieves better cross dataset generalization results than both the "non-deep" method (MIL) and the "deep" method (HED).

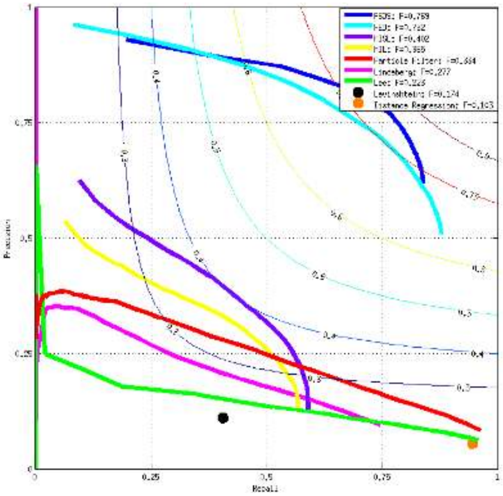

Figure 8. Evaluation of skeleton extractors on the WH-SYMMAX Dataset [23]. Leading skeleton extraction methods are ranked according to their best F-measure with respect to groundtruth skeletons. Our method, FSDS achieves the top result and shows both improved recall and precision at most of the precision-recall regime. See Table 2 for more details about the other two quantities and method citations.

4.2.5 Usefulness Verification

To verify the usefulness of the extracted skeletons, we do an experiment on object proposal detection. Let h_B^E be the objectness score of a bounding box B obtained by Edge Boxes [35], we define our objectness score by $h_B = \frac{\sum_{\forall \mathcal{M} \cap B \neq \emptyset} \mathcal{M} \cap B}{\sum_{\forall \mathcal{M} \cap B \neq \emptyset} \mathcal{M} + \epsilon} \cdot h_B^E$, where ϵ is a very small number to ensure the denominator to be non-zero, and $\mathcal{M}$ is a part mask reconstructed by a detected skeleton segment. As shown in Fig. 9, the new scoring method achieves a better object proposal detection result than Edge Boxes.

Table 3. Cross-dataset generalization results. TRAIN/TEST indicates the training/testing dataset used.

Method	Train/Test	F-measure
MIL [29]	SK506/WH-SYMMAX	0.363
HED [31]	SK506/WH-SYMMAX	0.637
FSDS (ours)	SK506/WH-SYMMAX	**0.692**
MIL [29]	WH-SYMMAX/SK506	0.387
HED [31]	WH-SYMMAX/SK506	0.492
FSDS (ours)	WH-SYMMAX/SK506	**0.529**

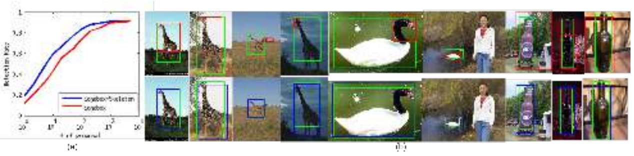

Figure 9. Object proposal results on ETHZ Shape Classes [11]. (a) The curve (IoU = 0.7). (b) Examples. Groundtruth (green), the closest proposal to groundtruth of Edgebox (red) and ours (blue).

We also show that the extracted skeleton can be used for symmetric part segmentation in the Supp..

5. Conclusion

We have presented a fully convolutional network with multiple scale-associated side outputs to extract skeletons from natural images. By pointing out the relationship between the receptive field sizes of the sequential stages in the network and the skeleton scales they can capture, we emphasized the important roles of the proposed scale-associated side outputs in (1) guiding multi-scale feature learning and (2) fusing scale-specific responses from different stages. The encouraging experimental results demonstrate the effectiveness of the proposed method for skeleton extraction from natural images. It achieves significant improvements to all other competitors.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China under Grant 61303095 and 61573160, in part by Research Fund for the Doctoral Program of Higher Education of China under Grant 20133108120017, and in part by Innovation Program of Shanghai Municipal Education Commission under Grant 14YZ018. We thank NVIDIA Corporation for providing their GPU device for our academic research.

References

[1] P. Agrawal, R. B. Girshick, and J. Malik. Analyzing the performance of multilayer neural networks for object recognition. In *Proc. ECCV*, pages 329–344, 2014.

[2] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *Proc. CVPR*, pages 73–80, 2010.

[3] X. Bai, L. J. Latecki, and W. Liu. Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):449–462, 2007.

[4] X. Bai, X. Wang, L. J. Latecki, W. Liu, and Z. Tu. Active skeleton for non-rigid object detection. In *Proc. ICCV*, pages 575–582, 2009.

[5] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, Newyork, NY, USA, 2006.

[6] H. Blum. *Models for the perception of speech and visual form*, chapter A Transformation for extracting new descriptors of shape, pages 363–380. MIT Press, Boston, MA, USA, 1967.

[7] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *Proc. ECCV*, pages 109–124, 2002.

[8] X. Chen, H. Fang, T. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv:1405.0312*, 2015.

[9] M. F. Demirci, A. Shokoufandeh, Y. Keselman, L. Bretzner, and S. J. Dickinson. Object recognition as many-to-many feature matching. *International Journal of Computer Vision*, 69(2):203–222, 2006.

[10] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(8):1558–1570, 2015.

[11] V. Ferrari, T. Tuytelaars, and L. J. V. Gool. Object detection by contour segment networks. In *Proc. ECCV*, pages 14–28, 2006.

[12] J.-H. Jang and K.-S. Hong. A pseudo-distance map for the segmentation-free skeletonization of gray-scale images. In *Proc. ICCV*, pages 18–25, 2001.

[13] S. Lee and Y. Liu. Curved glide-reflection symmetry detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(2):266–278, 2012.

[14] T. S. H. Lee, S. Fidler, and S. J. Dickinson. Detecting curved symmetric parts using a deformable disc model. In *Proc. ICCV*, pages 1753–1760, 2013.

[15] A. Levinshtein, S. J. Dickinson, and C. Sminchisescu. Multiscale symmetric part detection and grouping. In *Proc. ICCV*, pages 2162–2169, 2009.

[16] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):117–156, 1998.

[17] Y. Liu. *Computational symmetry in computer vision and computer graphics*. Now publishers Inc, Hanover, MA, USA, 2009.

[18] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*, pages 3431–3440, 2015.

[19] D. R. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. ICCV*, pages 416–425, 2001.

[20] X. Ren. Multi-scale improves boundary detection in natural images. In *Proc. ECCV*, pages 533–545, 2008.

[21] P. K. Saha, G. Borgefors, and G. S. di Baja. A survey on skeletonization algorithms and their applications. *Pattern Recognition Letters*, 2015.

[22] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):550–571, 2004.

[23] W. Shen, X. Bai, Z. Hu, and Z. Zhang. Multiple instance subspace learning via partial random projection tree for local reflection symmetry in nature images. *Pattern Recognition*, 52:266–278, 2016.

[24] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *Proc. CVPR*, pages 3982–3991, 2015.

[25] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):13–32, 1999.

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[27] A. Sironi, V. Lepetit, and P. Fua. Multiscale centerline detection by learning a scale-space distance transform. In *Proc. CVPR*, pages 2697–2704, 2014.

[28] 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.

[29] S. Tsogkas and I. Kokkinos. Learning-based symmetry detection in natural images. In *Proc. ECCV*, pages 41–54, 2012.

[30] N. Widynski, A. Moevus, and M. Mignotte. Local symmetry detection in natural images using a particle filtering approach. *IEEE Transactions on Image Processing*, 23(12):5309–5322, 2014.

[31] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proc. ICCV*, pages 1395–1403, 2015.

[32] Z. Yu and C. L. Bajaj. A segmentation-free approach for skeletonization of gray-scale images via anisotropic vector diffusion. In *Proc. CVPR*, pages 415–420, 2004.

[33] Q. Zhang and I. Couloigner. Accurate centerline detection and line width estimation of thick lines using the radon transform. *IEEE Transactions on Image Processing*, 16(2):310–316, 2007.

[34] Z. Zhang, W. Shen, C. Yao, and X. Bai. Symmetry-based text line detection in natural scenes. In *Proc. CVPR*, pages 2558–2567, 2015.

[35] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Proc. ECCV*, pages 391–405, 2014.