DeepSkeleton: Learning Multi-task Scale-associated Deep Side Outputs for Object Skeleton Extraction in Natural Images

Wei Shen, Kai Zhao, Yuan Jiang, Yan Wang, Xiang Bai and Alan Yuille

***Abstract*—Object skeletons are useful for object representation and object detection. They are complementary to the object contour, and provide extra information, such as how object scale (thickness) varies among object parts. But object skeleton extraction from natural images is very challenging, because it requires the extractor to be able to capture both local and non-local image context in order to determine the scale of each skeleton pixel. In this paper, we present a novel fully convolutional network with multiple scale-associated side outputs to address this problem. By observing the relationship between the receptive field sizes of the different layers in the network and the skeleton scales they can capture, we introduce two scale-associated side outputs to each stage of the network. The network is trained by multi-task learning, where one task is skeleton localization to classify whether a pixel is a skeleton pixel or not, and the other is skeleton scale prediction to regress the scale of each skeleton pixel. Supervision is imposed at different stages by guiding the scale-associated side outputs toward the groundtruth skeletons at the appropriate scales. The responses of the multiple scale-associated side outputs are then fused in a scale-specific way to detect skeleton pixels using multiple scales effectively. Our method achieves promising results on two skeleton extraction datasets, and significantly outperforms other competitors. Additionally, the usefulness of the obtained skeletons and scales (thickness) are verified on two object detection applications: Foreground object segmentation and object proposal detection.**

***Index Terms*—Skeleton, fully convolutional network, scale-associated side outputs, multi-task learning, object segmentation, object proposal detection.**

I. Introduction

In this paper, we investigate an important and nontrivial problem in computer vision, namely object skeleton extraction from natural images (Fig. 1). Here, the concept of "object" means a standalone entity with a well-defined boundary and center [1], such as an animal, a human, and a plane, as opposed to amorphous background stuff, such as sky, grass, and mountain. The skeleton, also called the *symmetry axis*, is a useful structure-based object descriptor. Extracting object skeletons directly from natural images can deliver important information about the presence and size of objects. Therefore, it is useful for many real applications including object recognition/detection [2], [3], text recognition [4], road detection and blood vessel detection [5].

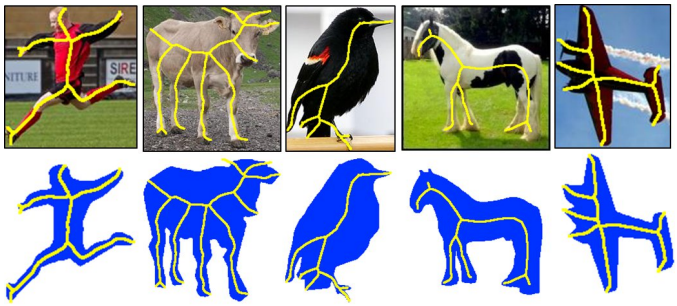

Fig. 1. Object skeleton extraction in natural images. The skeletons are in yellow. Top: Skeleton localization. Bottom: Scale prediction which enables object segmentation (blue regions are the segments reconstructed from skeletons according to the scales).

Skeleton extraction from pre-segmented images [6] has been well studied and successfully applied to shape-based object matching and recognition [7]–[10]. However, such methods have severe limitations when applied to natural images, because segmentation from natural images is still an unsolved problem.

Skeleton extraction from natural images is a very challenging problem, which requires addressing two tasks. One is skeleton localization to classify whether a pixel is a skeleton pixel or not (the top row in Fig. 1) and the other is skeleton scale prediction to estimate the scale of each skeleton pixel (the bottom row in Fig. 1). The latter task has not been studied explicitly in the past, although it is very important, because using the predicted scales, we can obtain object segmentation from a skeleton directly. In this paper, we address skeleton localization and scale prediction in a unified framework which performs them simultaneously. The main difficulties for skeleton extraction stem from four issues: (1) The complexity of natural scenes: Natural scenes are typically very cluttered. Amorphous background elements, such as fences, bricks and even the shadows of objects, exhibit some self-symmetry, and thus can cause distractions. (2) The diversity of object appearance: Objects in natural images exhibit very different colors, textures, shapes and sizes. (3) The variability of skeletons: local skeleton segments have a variety of patterns, such as straight lines, T-junctions and Y-junctions. (4) The *unknown-scale problem*: A local skeleton segment is naturally

W. Shen, K. Zhao and Y. Jiang are with Key Laboratory of Specialty Fiber Optics and Optical Access Networks, Shanghai University, Shanghai 200444 China. W. Shen is also with Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218-2608 USA. E-mail: shenwei1231@gmail.com, {zeakey,jy9387}@outlook.com.

Y. Wang is with Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218-2608 USA. E-mail: wyanny.9@gmail.com

X. Bai is with School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074 China. Email: xiang.bai@gmail.com

A. Yuille is with Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218-2608 USA. E-mail: alan.yuille@jhu.edu .

associated with an unknown scale, determined by the thickness of its corresponding object part. We term this last problem the unknown-scale problem for skeleton extraction.

A number of methods have been proposed to perform skeleton extraction or skeleton localization in the past decade. Broadly speaking, they can be categorized into two groups: (1) Traditional image processing methods [11]–[14], which compute skeletons from a gradient intensity map according to some geometric constraints between edges and skeletons. Due to the lack of supervised learning, these methods have difficulty in handling images with complex scenes; (2) Recent learning based methods [5], [15]–[18], which learn a per-pixel classification or segment-linking model based on hand-designed features for skeleton extraction computed at multi-scales. But the limitations of hand-designed features cause these methods to fail to extract the skeletons of objects with complex structures and cluttered interior textures. In addition, such per-pixel/segment models are usually time consuming. More importantly, most current methods only focus on skeleton localization, but are unable to predict skeleton scales, or are only able to provide a coarse prediction for skeleton scales. This big shortcoming limits the application of the extracted skeletons to object detection. Consequently, there remain big gaps between these skeleton extraction methods and human perception, in both performance and speed. Skeleton extraction has the unique aspect of requiring both local and non-local image context, which requires new techniques for both multi-scale feature learning and classifier learning. This is challenging, since 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 (HED) [19] is a deep fully convolutional network (FCN) [20], which enables holistic image training and prediction for per-pixel tasks. A side output is the output of a hidden layer of a deep network. The side outputs of the hidden layers, from shallow to deep, give multi-scale responses, and can be guided by supervision to improve the directness and transparency of the hidden layer learning process [21]. Here we connect two sibling scale-associated side outputs 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 at a specific scale; then only the filters with sizes larger than the scale will have responses, and others will not. Note that the sequential convolutional layers in a hierarchical network can be consider as filters with increasing sizes (the receptive field sizes of 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. This sequence of increasing receptive field sizes provide a principle to quantize the skeleton scale space. With these observations, we propose to impose supervision at each side output (SO), optimizing them towards a scale-associated groundtruth skeleton map. More specifically, only

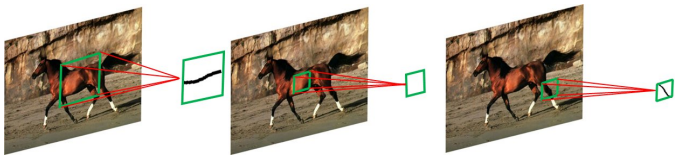

Fig. 2. Using filters (the green squares on images) of multiple sizes for skeleton extraction. Only when the size of the filter is larger than the scale of current skeleton part can the filter capture enough context feature to detect it.

skeleton pixels whose scales are smaller than the receptive field size of the SO are labeled by quantized scale values. The two sibling SOs at each stage are trained with multi-task loss for both skeleton localization and skeleton scale prediction. Thus the SOs at each stage are associated with specific scales and give a number of scale-specific skeleton score maps (the score map for one specified quantized scale value) as well as a skeleton scale map. Since the SOs in our network are scale-associated, we call them scale-associated side outputs (SSOs) and we refer to the SSOs for skeleton localization and skeleton scale prediction as Loc-SSO and ScalePred-SSO respectively.

The final predicted skeleton map is obtained by fusing Loc-SSOs. A straightforward fusion method is to average them. However, a skeleton pixel with large scale typically has a stronger response at the deeper SOs, and a weaker response at the shallower SOs; By contrast, a skeleton pixel with small scale may have strong responses at both of the two SOs. This motivates us to use a scale-specific weight layer to fuse the corresponding scale-specific skeleton score maps provided by each Loc-SSO.

In summary, the core contribution of this paper is the scale-associated side output layers, which enable both multi-task learning and fusion in a scale-depended way, to deal with the unknown scale problem. Therefore our network is able to detect skeleton pixels at multiple scales and estimate the scales.

To evaluate the performances of skeleton extraction methods, datasets with groundtruth skeleton maps as well as groudtruth scale maps are required. We constructed such a dataset in our previous work [22], which we called SK506[1]. There are 506 natural images in this dataset, which were selected from the recent published MS COCO dataset [23]. A skeletonization method [24] was applied to the human-annotated foreground segmentation maps of the selected images to generate the groundtruth skeleton maps and the groundtruth scale maps. But the size of this dataset was small. Therefore, in this paper, we construct a larger dataset, containing $1,491$ natural images, annotated in the same way. We rename the SK506 dataset SK-SMALL and call the newly constructed one SK-LARGE. For consistency, SK-SMALL is a subset of SK-LARGE.

This paper extends our preliminary work [22] by the following contributions: (1) Training the side outputs of each stage with a multi-task loss by introducing a new scale regression term. (2) Constructing a larger dataset for skeleton extraction. (3) More experimental results and discussions about

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

the usefulness of the extracted skeletons in object detection applications.

II. Related Works

Object skeleton extraction has been studied a lot in recent decades. However, most works in the early stages [6], [24] only focus on skeleton extraction from pre-segmented images. As these works make a strict assumption that object silhouettes are provided, i.e., the object has already been segmented, they cannot be applied to our task.

Pioneering researchers tried to extract skeletons from the gradient intensity maps computed on natural images. The gradient intensity map was typically obtained by applying directional derivative operators to a gray-scale image smoothed by a Gaussian kernel. For instance, in [13], Lindeberg provided an automatic mechanism to determine the best size of the Gaussian kernel for gradient computation, and also proposed to detect skeletons as the pixels for which the gradient intensity takes a local maximum (minimum) in the direction of the main principal curvature. In [25], he also gave a theoretic analysis of such scale selection mechanisms and showed that they are useful for other low level feature detection, such as interesting point detection. Majer [26] pointed out that the second derivative of Gaussian filter kernel can detect skeletons under the assumption that skeletons are consider to be step or Gaussian ridge models. Jang and Hong [12] extracted the skeleton from the pseudo-distance map which was obtained by iteratively minimizing an object function defined on the gradient intensity map. Yu and Bajaj [11] proposed to trace the ridges of the skeleton intensity map calculated from the diffused vector field of the gradient intensity map, which can remove undesirablely biased skeletons. [27] was the pioneer for detecting symmetry and perform segmentation simultaneously by modeling and linking local symmetry parts, where skeleton extraction was formulated in terms of minimizing a goodness of fitness function defined on the gradient intensities. But due to the lack of supervised learning, these methods are only able to handle images with simple scenes.

Recent learning based skeleton extraction methods are better at dealing with complex scene. One type of methods formulates skeleton extraction as a per-pixel classification problem. Tsogkas and Kokkinos [15] computed hand-designed features of multi-scale and multi-orientation at each pixel, and employed multiple instance learning to determine whether it is symmetric[2] or not. Shen *et al.* [28] then improved this method by training MIL models on automatically learned scale- and orientation-related subspaces. Sironi *et al.* [5] transformed the per-pixel classification problem to a regression one to achieve skeleton localization and learn the distance to the closest skeleton segment in scale-space. Another type of learning based methods aims to learn the similarity between local skeleton segments (represented by superpixel [16], [17] or spine model [18]), and links them by hierarchical clustering [16], dynamic programming [17] or particle filtering [18]. Due to the limited power of hand-designed features, these methods are not effective at detecting skeleton pixels with large scales, as large context information is needed.

Our method was inspired by [19], which developed a holistically-nested network for edge detection (HED). But detecting edges does not need to deal with scales explicitly. Using a local filter to detect an edge pixel, no matter what the size of the filter is, will give some response. So summing up the multi-scale detection responses, which occurs in the fusion layer in HED, is able to improve the performance of edge detection [29]–[31], while bringing false positives across the scales for skeleton extraction (see the results in Fig. 6). There are three main differences between HED and our method. (1) We supervise the SOs of the network with different scale-associated groundtruths, but the groundtruths in HED are the same at all scales. (2) We use different scale-specific weight layers to fuse the corresponding scale-specific skeleton score maps provided by the SOs, while the SOs are fused by a single weight layer in HED. (3) We perform multi-task learning for the SOs of each stage by introducing a new scale regression loss, but only classification loss is considered in HED. The first two changes use the multi stages in a network to explicitly detect the unknown scale, which HED is unable to deal with. While the last change takes advantage of scale supervision to let our method provide a more informative result, i.e., the predicted scale for each skeleton pixel, which is useful for other potential applications, such as object segmentation and object proposal detection (we will show this in Sec. IV-C and Sec. IV-D). By contrast, the output of HED cannot be applied to these applications.

There are only two other datasets related to our task. One is the SYMMAX300 dataset [15], which is converted from the well-known Berkeley Segmentation Benchmark (BSDS300) [32]. But this dataset is used mostly for local reflection symmetry detection. Local reflection symmetry [33], [34] is a low-level feature of images, and does not depend on the concept of "object". Some examples from this dataset are shown in Fig. 3(a). Note that a large number of symmetries occur outside object. In general, the object skeletons are a subset of the local reflection symmetry. Another dataset is WH-SYMMAX [28], which is converted from the Weizmann Horse dataset [35]. This dataset is suitable to verify object skeleton extraction methods; however, as shown in Fig. 3(b) a limitation is that only one object category, the horse, is contained in it. On the contrary, the objects, in our newly built dataset SK-LARGE, belong to a variety of categories, including humans, animals, such as birds, dogs and giraffes, and man made objects, such as planes and hydrants (Fig. 3(c)). Therefore, SK-LARGE not only contains more images, but also has more variability in object scales. We evaluate several skeleton extraction methods as well as symmetry detection methods on WH-SYMMAX, SK-SMALL and SK-LARGE. The experimental results demonstrate that our method significantly outperforms others.

III. Methodology

In this section, we describe our methods for object skeleton localization and scale prediction. First, we introduce the ar-

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

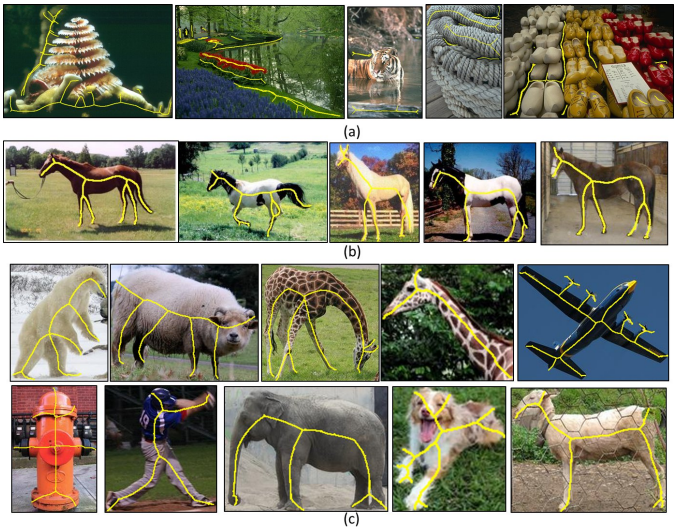

Fig. 3. Samples from three datasets. (a) The SYMMAX300 dataset [15]. (b) The WH-SYMMAX dataset [28]. (c) SK-SMALL and SK-LARGE constructed by us. The groundtruths for skeletons and local reflection symmetries are in yellow.

chitecture of our network. Then, we discuss how to optimize and fuse the multiple scale-associated side outputs (SSOs) to extract the skeleton and predict the scale.

A. Network Architecture

We propose a new architecture for skeleton extraction, which is built on the HED network [19]. HED is used for edge detection. Here, to address the unknown scale problem in skeleton extraction, we make two important modifications in our network: (a) we connect the proposed Loc-SSO and ScalePred-SSO layers 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 sequential stages 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 is too small (only 5 pixels) to capture any skeleton features. There are only a few skeleton pixels with scales less than such a small receptive field. (b) Each Loc-SSO is connected to a slice layer to obtain the skeleton score map for each scale. Then from all these SO 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 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. An illustration for these two modifications are shown in Fig. 4(a) and Fig. 4(b), respectively. To sum up, our holistically-nested network architecture has 4 stages with additional SSO 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 Loc-SSOs.

B. Skeleton Extraction by Learning Multi-task Scale-associated Side Outputs

Skeleton localization can be formulated as a per-pixel classification problem. Given a raw input image $X = \{x_j, j = 1, \ldots, |X|\}$, the 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, $\hat{y}_j = 0$. Here, we also aim to predict the scale map $\hat{S} = \{\hat{s}_j, j = 1, \ldots, |X|\}$, where $\hat{s}_j \in \mathbb{R}$, and $\hat{s}_j > 0$ if $\hat{y}_j = 1$; otherwise $\hat{s}_j = 0$ if $\hat{y}_j = 0$. This is a per-pixel regression problem. To sum up, our purpose is to address two tasks: One is skeleton localization, which takes input X and outputs $\hat{Y}$; the other is scale prediction, whose input is X and outputs $\hat{Y}$ and $\hat{S}$ simultaneously. By addressing the latter task, not only can the performance of the former be improved (Sec. IV-B2), but the object segmentation map can be obtained directly (Sec. IV-C). Next, we describe how to learn and fuse the SSOs in the training phase as well as how to use the learned network in the testing phase, respectively.

1) Training Phase: Following the definition of skeletons [37], 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 groundtruth segmentation map. So we are given a training dataset denoted by $\{(X^{(n)}, Y^{(n)}, S^{(n)}), n = 1, \ldots, N\}$, where $X^{(n)} = \{x_j^{(n)}, j = 1, \ldots, |X^{(n)}|\}$ is a raw input image and $Y^{(n)} = \{y_j^{(n)}, j = 1, \ldots, |X^{(n)}|\}$ ($y_j^{(n)} \in \{0, 1\}$) and $S^{(n)} = \{s_j^{(n)}, j = 1, \ldots, |X^{(n)}|\}$ ($s_j^{(n)} \geq 0$) are its corresponding groundtruth skeleton map and groundtruth scale map. Note that, we have $y_j^{(n)} = \mathbf{1}(s_j^{(n)} > 0)$, where $\mathbf{1}(\cdot)$ is an indicator function. First, we describe how to compute a quantized skeleton scale map for each training image, which will be used for guiding the network training.

a) Skeleton scale quantization.: As now we consider a single image, we drop the image superscript n. We aim to learn a network with multiple stages of convolutional layers linked with two sibling SSO layers. Assume that there are M such stages in our network, in which the receptive field sizes of the convolutional layers increase in sequence. 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 inside 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 to capture the features of the skeleton pixels with the maximum scale). The quantized value z of a scale s is computed by

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

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

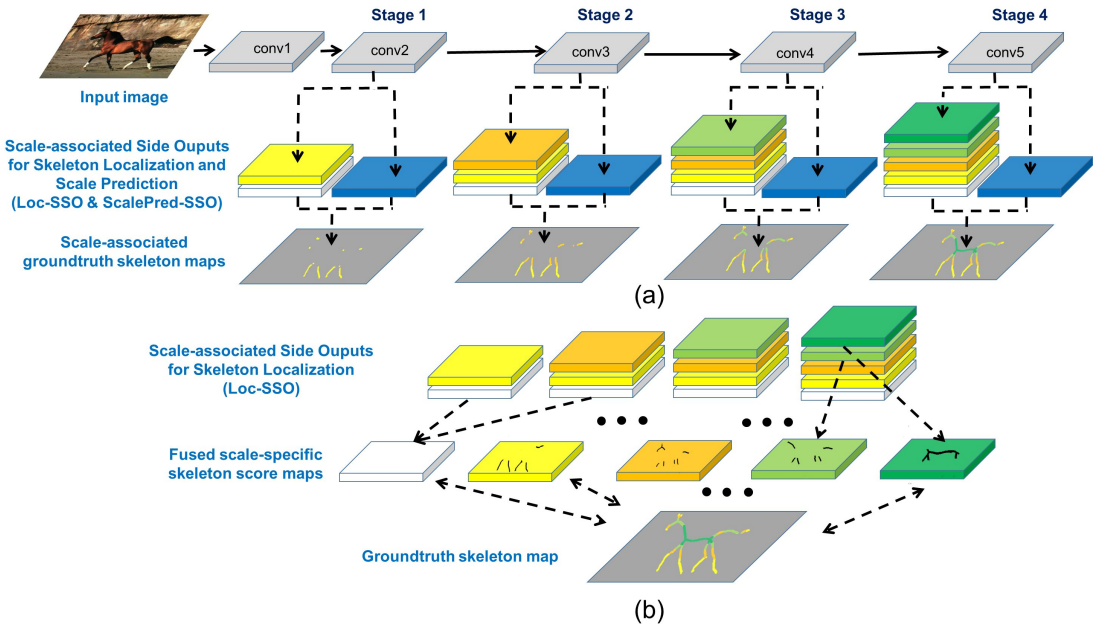

Fig. 4. The proposed network architecture for skeleton extraction, which is converted from VGG 16-layer net [36]. (a) Multi-task Scale-associated side outputs (SSOs) learning. Our network has 4 stages with SSO layers connected to the convolutional layers. Each stage branches into two sibling SSO layers, one for skeleton localization and the other for scale prediction, denoted by Loc-SSO (the left multi-color blocks) and ScalePred-SSO (the right blue block), respectively. The SSOs in each stage are guided by a scale-associated groundtruth skeleton map (The skeleton pixels with different quantized scales are in different colors. Each block in a Loc-SSO is the activation map for one quantized scale, marked by the corresponding color). (b) Scale-specific fusion. Each Loc-SSO 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.

b) Scale-associated side outputs learning for pixel classification.: The groundtruth skeleton map Y can be trivially computed from Z: $Y = \mathbf{1}(Z > 0)$, but not vice versa. So we guide the network training by Z instead of Y, since it gives more supervision. This converts a binary classification problem to a multi-class classification one, where each class corresponds to a quantized scale. Towards this end, each Loc-SSO layer in our network is associated with a softmax classifier. But according to the above discussions, each stage in our network is only able to detect the skeleton pixels at scales less than its corresponding receptive field size. Therefore, the side output is scale-associated. For the i-th Loc-SSO, we supervise 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 $K^{(i)} = i$, then we have $Z^{(i)} = \{z_j^{(i)}, j = 1, \ldots, |X|\}, z_j^{(i)} \in \{0, 1, \ldots, K^{(i)}\}$. To better understand this computation, we show an example of computing these variables in Fig. 5. Let $\ell_{cls}^{(i)}(\mathbf{W}, \mathbf{\Phi}^{(i)})$ denote the loss function for this Loc-SSO, where $\mathbf{W}$ and $\mathbf{\Phi}^{(i)}$ are the layer parameters of the network and the parameters of the classifier of this stage. The loss function of our network is computed over all pixels in the training image X and the scale-associated groundtruth skeleton map $Z^{(i)}$. Generally, the numbers of skeleton pixels at different scales are different and are much less than the number of non-skeleton pixels in an image. Therefore, we define a weighted softmax loss function to balance the loss between these multiple classes:

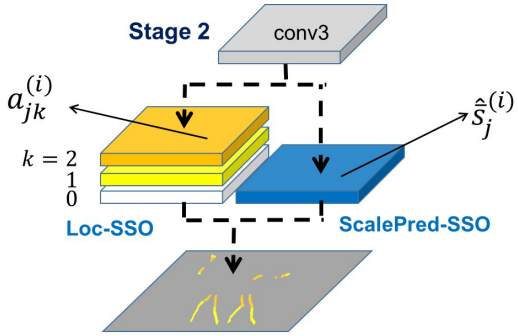

Fig. 5. An example of the computation of the scale-associated side outputs (SSOs) at each stage. The stage index is 2. Thus, $i = 2$, $K^{(i)} = 2$. $a_{jk}^{(i)}$ and $\hat{s}_j^{(i)}$ are the activations of the i-th Loc-SSO associated with the quantized scale k and the i-th ScalePred-SSO for the input x_j, respectively. Please refer to text to see the meanings of the notations.

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

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. Let $\mathcal{N}(\cdot)$ denote 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 Loc-SSO associated with the quantized scale k for the input x_j (Fig. 5), then we use the softmax function [38] $\sigma(\cdot)$ to compute

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

The partial derivation of $\ell_{cls}^{(i)}(\mathbf{W}, \mathbf{\Phi}^{(i)})$ w.r.t. $a_{jl}^{(i)}$ ($l \in \{0, 1, \ldots, K^{(i)}\}$) is obtained by

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

c) Scale-associated side outputs learning for scale prediction.: As we described, scale prediction is a per-pixel regression problem. In a regression problem, regression target normalization is a crucial pre-process. The receptive field size of each stage can serve as a good reference for scale normalization. For the i-th ScalePred-SSO, we guide it to a normalized scale-associated groundtruth skeleton map $\bar{S}^{(i)} = 2\frac{Z^{(i)} \circ S}{r_i} - 1$. This normalization maps each element s_j in S into the range $[-1, 1)$. Let $\hat{\bar{s}}_j^{(i)}$ be the predicted scale by the i-th ScalePred-SSO, i.e., the activation of the i-th ScalePred-SSO for the input x_j (Fig. 5), the regression loss is defined by

$$\ell_{reg}^{(i)}(\mathbf{W}, \mathbf{\Psi}^{(i)}) = \frac{\sum_{j=1}^{|X|} \mathbf{1}(z_j^{(i)} > 0)\|\hat{\bar{s}}_j^{(i)} - \bar{s}_j^{(i)}\|_2^2}{\mathcal{N}(\mathbf{1}(Z^{(i)} > 0))}, \quad (6)$$

where $\mathbf{\Psi}^{(i)}$ is the parameter of the regressor for i-th stage. Note that, for non skeleton pixels and those which have too large scale to be captured by this stage, do not contribute to the regression loss $\ell_{reg}^{(i)}$.

d) Multi-task loss.: Each stage in our network has two sibling side output layers, i.e., Loc-SSO and ScalePred-SSO. We use a multi-task loss to jointly train them:

$$\ell_s^{(i)}(\mathbf{W}, \mathbf{\Phi}^{(i)}, \mathbf{\Psi}^{(i)}) = \ell_{cls}^{(i)}(\mathbf{W}, \mathbf{\Phi}^{(i)}) + \lambda \ell_{reg}^{(i)}(\mathbf{W}, \mathbf{\Psi}^{(i)}), \quad (7)$$

where the hyper-parameter λ controls the balance between the two task losses. Then the loss function for all the side outputs is simply obtained by

$$\mathcal{L}_s(\mathbf{W}, \mathbf{\Phi}, \mathbf{\Psi}) = \sum_{i=1}^{M} \ell_s^{(i)}(\mathbf{W}, \mathbf{\Phi}^{(i)}, \mathbf{\Psi}^{(i)}). \quad (8)$$

where $\mathbf{\Phi} = (\mathbf{\Phi}^{(i)}; i = 1, \ldots, M)$ and $\mathbf{\Psi} = (\psi^{(i)}; i = 1, \ldots, M)$ denote the parameters of the classifiers and the regressors in all the stages, respectively.

e) 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 $\mathbf{h}_k = (h_k^{(i)}; i = \max(k, 1), \ldots, M)$:

$$f_{jk} = \sum_{i=\max(k,1)}^{M} h_k^{(i)}\Pr(z_j^{(i)} = k|X; \mathbf{W}, \mathbf{\Phi}^{(i)}), \quad \text{s.t.} \sum_{i=\max(k,1)}^{M} h_k^{(i)} = 1. \quad (9)$$

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

$$\mathcal{L}_f(\mathbf{W}, \mathbf{\Phi}, \mathbf{H}) = -\frac{1}{|X|}\sum_{j=1}^{|X|}\sum_{k=0}^{M} \beta_k \mathbf{1}(z_j = k)\log\Pr(z_j = k|X; \mathbf{W}, \mathbf{\Phi}, \mathbf{h}_k), \quad (10)$$

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

$$(\mathbf{W}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{H})* = \arg\min(\mathcal{L}_s(\mathbf{W}, \mathbf{\Phi}, \mathbf{\Psi}) + \mathcal{L}_f(\mathbf{W}, \mathbf{\Phi}, \mathbf{H})). \quad (11)$$

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

$$\hat{y}_j = 1 - \Pr(z_j = 0|X; \mathbf{W}*, \mathbf{\Phi}*, \mathbf{h}_0*). \quad (12)$$

Recall that $z_j = 0$ and $z_j > 0$ mean that x_j is a non-skeleton/skeleton pixel, respectively. To predict the scale for each x_j, we first find its most likely quantized scale by

$$i* = \arg\max_{i=(1,\ldots,M)} \Pr(z_j = i|X; \mathbf{W}*, \mathbf{\Phi}*, \mathbf{h}_i*). \quad (13)$$

Then the predicted scale $\hat{s}_j$ is computed by

$$\hat{s}_j = \frac{\hat{\bar{s}}_j^{(i*)} + 1}{2} r_{i*}, \quad (14)$$

where $\hat{\bar{s}}_j^{(i*)}$ is the activation of the $i*$-th ScalePred-SSO. We refer to our method as LMSDS, for learning multi-task scale-associated deep side outputs.

C. Understanding of the Proposed Method

To understand our method more deeply, we illustrate the intermediate results and compare them with those of HED in Fig. 6. The response of each Loc-SSO can be obtained by the similar way of Eqn. 12. We compare the response of each Loc-SSO to the corresponding side output 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 side output fires on the structures 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 become 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 corresponds to one scale, e.g., the first three response maps in Fig. 6 corresponding to legs, neck and torso respectively. By contrast, 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 causes false positives to the final fusion one.

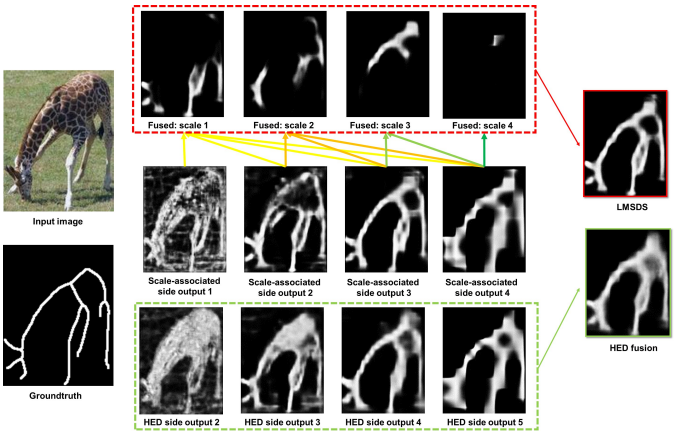

Fig. 6. The comparison between the intermediate results of LMSDS and HED. We observe that the middle row are able to differentiate skeleton pixels with different scales, while the latter cannot.

IV. Experimental Results

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

A. Implementation Details

Our implementation is based on "Caffe" [39] and our architecture is built on the public available implementation of FCN [20] and HED [19]. The whole network is fine-tuned from an initialization with the pre-trained VGG 16-layer net [36].This net is pre-trained on the subset of ImageNet used in an image classification challenge, called ILSVRC-2014 [40], which has 1000 categories and 1.2 million images.

a) Groundtruth generation: The groundtruth skeleton map for each image is computed from its corresponding human-annotated foreground segmentation mask (1 for foreground objects and 0 for background). We apply a binary image skeletonization method based on the distance transform [24] to these segmentation masks to generate the skeleton maps (1 for skeleton pixels and 0 for non-skeleton pixels) and use them as the groundtruths. The groundtruth scale of each skeleton pixel is two times of the minimal distance between this skeleton pixel and the boundary of the corresponding foreground segmentation mask.

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

c) Data augmentation: Data augmentation is a standard 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 them 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.

B. Skeleton Localization

1) Evaluation Protocol: To evaluate skeleton localization performances, we follow the protocol used in [15], under which the detected skeletons are measured by their maximum *F-measure* ($\frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$) as well as precision-recall curves with respect to the groundtruth skeleton map. To obtain the precision-recall curves, the detected skeleton response is first thresholded into a binary map, which is then matched with the groundtruth skeleton map. The matching allows small localization errors between detected positives and groundtruths. If a detected positive is matched with at least one groundtruth skeleton pixel, it is classified as a true positive. By contrast, pixels that do not correspond to any groundtruth skeleton pixel are false positives. By assigning different thresholds to the detected skeleton response, we obtain a sequence of precision and recall pairs, which is used to plot the precision-recall curve.

2) Design Evaluation: The main difference between LMSDS and our preliminary work [22], FSDS, is that we apply multi-task learning in LMSDS. Since the two tasks influence each other through their shared representation (convolutional features), we can ask how multi-task learning influences the result of skeleton localization?

To answer this question, we compare the skeleton localization performances of these two methods on three datasets: SK-LARGE, SK-SMALL and WH-SYMMAX. Note that, by setting $\lambda = 0$ in Eqn. 7, LMSDS reduces to FSDS. The comparison is summarized in Table I, from which we observe that training with multi-task loss leads to a slight decrease in skeleton localization performance on SK-SMALL, but yeilds considerable improvements on SK-LARGE and WH-SYMMAX. The reason why the results are opposite on SK-SMALL and SK-LARGE may be because scale prediction is more difficult than skeleton localization, i.e., training a good model by using multi-task loss requires more training data. Although the training set of WH-SYMMAX is small, the variance of the data is also small, because only one object category is contained in it. To sum up, we argue that multi-task training with sufficient training data can improve

pure skeleton localization compared to training for skeleton localization alone. In Sec. IV-C, we will show that multi-task learning is important to obtain accurate predicted scales, which is useful for skeleton based object segmentation.

TABLE I
THE VALIDATION OF THE INFLUENCE OF MULTI-TASK TRAINING ON SKELETON LOCALIZATION. THE LOCALIZATION RESULTS ARE MEASURED BY THEIR F-MEANSURES.

	SK-SMALL	SK-LARGE	WH-SYMMAX
FSDS	**0.623**	0.633	0.769
LMSDS	0.621	**0.649**	**0.779**

Since our network is finetuned from the pre-trained VGG 16-layer net, another question is does the pre-trained VGG 16-layer net already have the ability to detect skeletons? To verify this, we consider two network parameter settings. One is we fix the weights of the VGG part in our network and train the rest part (denoted by LMSDS-VGGFixed w Finetune), the other is we fix the weights of the VGG part in our network and leave the rest in random initialization (denoted by LMSDS-VGGFixed w/o Finetune). As shown in Fig. 7, the performance of "LMSDS-VGGFixed w Finetune" drops significantly and "LMSDS-VGGFixed w/o Finetune" even does not work (The skeleton detection results are nearly random noises. So for all the points on its precision-recall curve, the precision is very low and the recall is near 0.5.). This result demonstrates that the pre-trained VGG 16-layer net is purely for the initialization of a part of our network, e.g., it does not initialize the weights for the SSOs layers, and final weights of our network differ enormously from the initial weights. Consequently, the pre-trained VGG 16-layer net does not have the ability to detect skeletons.

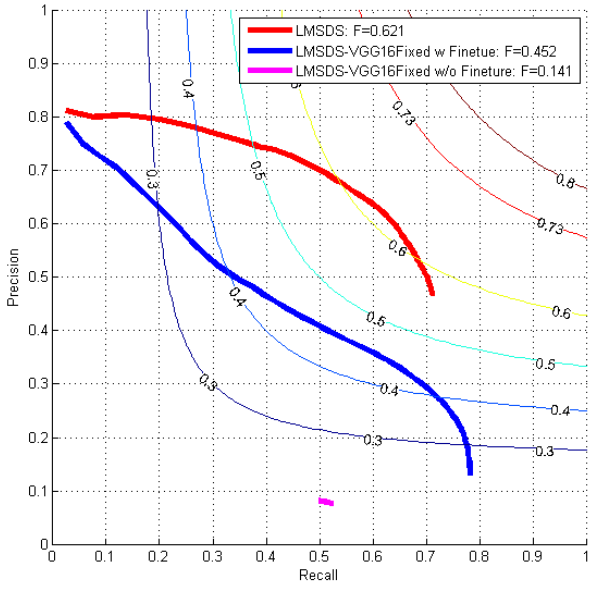

Fig. 7. The comparison between different network parameter settings in LMSDS.

3) Performance Comparison: We conduct our experiments by comparing our method LMSDS with others, including a traditional image processing method (Lindeberg's method [13]), three learning based segment linking methods (Levinshtein's method [16], Lee's method [17] and Particle Filter [18]), three per-pixel classification/regression methods (Distance Regression [5], MIL [15] and MISL [28]) and two deep learning based method (HED [19] and FSDS [22]). For all theses methods, we use the source code provided by the authors with the default setting. For HED, FSDS and LMSDS, we perform sufficient iterations to ensure convergence. We apply a standard non-maximal suppression algorithm [30] to the response maps of HED and ours to obtain the thinned skeletons for performance evaluation.

a) SK-LARGE: We first conduct our experiments on our newly built SK-LARGE dataset. Object skeletons in this dataset have large variabilities in both structures and scales. We split this dataset into 746 training and 745 testing images. We report the F-measure as well as the average runtime per image of each method on this dataset in Table. II. Observed that, both traditional image processing and per-pixel/segment learning methods do not perform well, indicating the difficulty of this task. Moreover, the segment linking methods are extremely time consuming. Our method LMSDS outperforms others significantly, even compared with the deep learning based method HED. In addition, 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. 8 show again that LMSDS is better than the alternatives, as ours gives both improved recall and precision in most of the precision-recall regimes. We illustrate the skeleton extraction results obtained by several methods in Fig. 9 for qualitative comparison. These qualitative examples show that our method detects more groundtruth skeleton points and also suppresses false positives. The false positives in the results of HED are probably introduced because it does not use learning to combine different scales. Benefiting from scale-associated learning and scale-specific fusion, our method is able to suppress these false positives.

TABLE II
SKELETON LOCALIZATION PERFORMANCE COMPARISON BETWEEN DIFFERENT METHODS ON SK-LARGE. †GPU TIME.

Method	F-measure	Avg Runtime (sec)
Lindeberg [13]	0.270	4.05
Levinshtein [16]	0.243	146.21
Lee [17]	0.255	609.10
MIL [15]	0.293	42.40
HED [19]	0.497	0.05†
FSDS (ours)	0.633	0.05†
LMSDS (ours)	0.649	0.05†

b) SK-SMALL: We then perform comparisons on SK-SMALL. The training and testing sets of SK-SMALL contain 300 and 206 images, respectively. From the precision/recall curves shown in Fig. 10 and summary statistics reported in Table. III, we observe that LMSDS outperforms the others, except for our preliminary method, FSDS. LMSDS performs slightly worse on skeleton localization on SK-SMALL, for reasons we discussed in Sec. IV-B2.

Fig. 9. Illustration of skeleton extraction results on SK-LARGE 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.

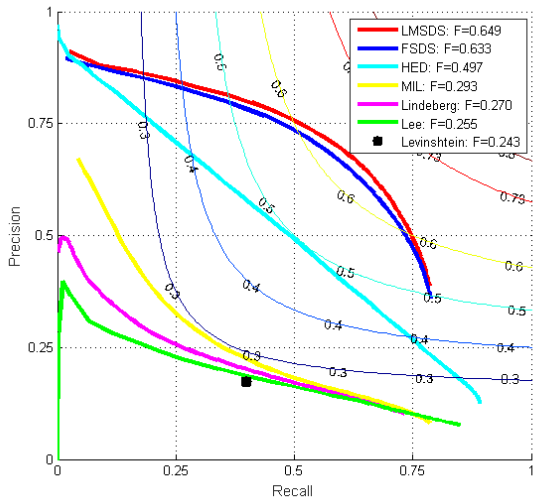

Fig. 8. Skeleton localization evaluation of skeletons extracted on SK-LARGE, which consists of 746 training and 745 testing images. Leading skeleton extraction methods are ranked according to their best F-measure with respect to groundtruth skeletons. LMSDS and FSDS achieve the top and the second best results, respectively. See Table II for more details about the other quantity (Avg Runtime) and citations to competitors.

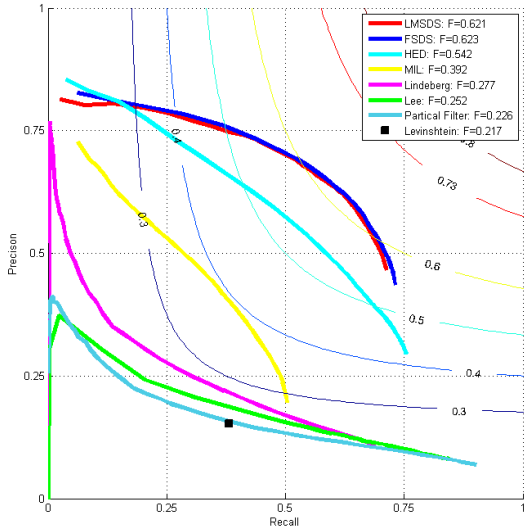

Fig. 10. Skeleton localization evaluation of skeleton extractors on SK-SMALL, which consists of 300 training and 206 testing images. Skeleton extraction methods are measured by their best F-measure with respect to groundtruth skeletons. FSDS and LMSDS achieve the top and the second best results, respectively. See Table III for more details about the other quantity (Avg Runtime) and citations to competitors.

TABLE III
SKELETON LOCALIZATION PERFORMANCE COMPARISON BETWEEN DIFFERENT METHODS ON SK-SMALL. †GPU TIME.

Method	F-measure	Avg Runtime (sec)
Lindeberg [13]	0.277	4.03
Levinshtein [16]	0.218	144.77
Lee [17]	0.252	606.30
Particle Filter [18]	0.226	322.25†
MIL [15]	0.392	42.38
HED [19]	0.542	0.05†
FSDS (ours)	0.623	0.05†
LMSDS (ours)	0.621	0.05†

c) WH-SYMMAX: The WH-SYMMAX dataset [28] contains 328 images, of 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. 12 and summary statistics are in Table IV. Qualitative comparisons are illustrated in Fig. 13. Both quantitative and qualitative results demonstrate that our method is clearly better than others.

d) Skeleton Extraction for Multiple Objects: Our method does not have the constraint that one image can only contain a single object. Here, we directly apply our model trained on SK-SMALL to images from SYMMAX300 [15], which contain multiple objects and complex background, e.g., the merged zebras. As the comparison shows in Fig. 11, our method can obtain good skeletons for each object in these images, which have significantly less false positives corre-

TABLE IV
SKELETON LOCALIZATION PERFORMANCE COMPARISON BETWEEN DIFFERENT METHODS ON WH-SYMMAX [28]. †GPU TIME.

Method	F-measure	Avg Runtime (sec)
Lindeberg [13]	0.277	5.75
Levinshtein [16]	0.174	105.51
Lee [17]	0.223	716.18
Particle Filter [18]	0.334	13.9†
Distance Regression [5]	0.103	5.78
MIL [15]	0.365	51.19
MISL [28]	0.402	78.41
HED [19]	0.732	0.06†
FSDS (ours)	0.769	0.07†
LMSDS (ours)	0.779	0.07†

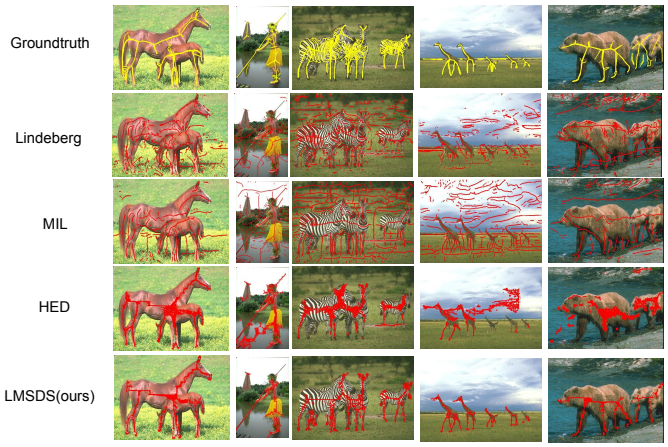

Fig. 11. Illustration of skeleton extraction results on the SYMMAX300 dataset [15] for several selected images. The groundtruth skeletons are in yellow and the thresholded extraction results are in red. Thresholds were optimized over the whole dataset.

sponding to background and interior textures.

e) Cross Dataset Generalization: A concern is that the scale-associated side outputs learned from one dataset might lead to higher generalization error when applied 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 [15], HED [19] and our method in Table V. Our method achieves better cross dataset generalization results than both the "non-deep" method (MIL) and the "deep" method (HED).

TABLE V
CROSS-DATASET GENERALIZATION RESULTS. TRAIN/TEST INDICATES THE TRAINING/TESTING DATASET USED.

Method	Train/Test	F-measure
MIL [15]	SK-LARGE/WH-SYMMAX	0.350
HED [19]	SK-LARGE/WH-SYMMAX	0.583
LMSDS (ours)	SK-SMALL/WH-SYMMAX	0.701
MIL [15]	WH-SYMMAX/SK-LARGE	0.357
HED [19]	WH-SYMMAX/SK-LARGE	0.420
LMSDS (ours)	WH-SYMMAX/SK-LARGE	0.474

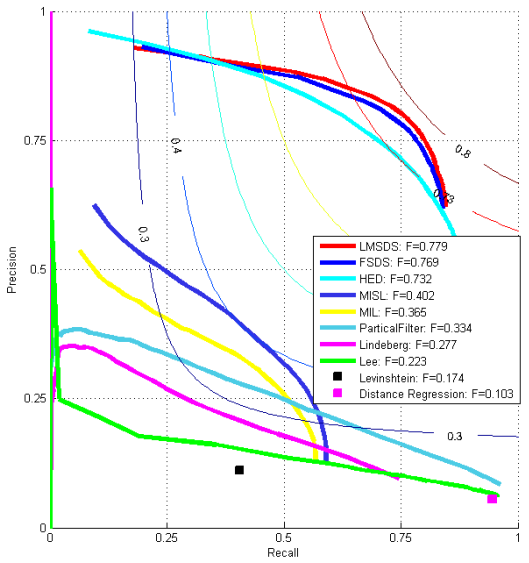

Fig. 12. Evaluation of skeleton extractors on WH-SYMMAX [28], which consists of 228 training and 100 testing images. Leading skeleton extraction methods are ranked according to their best F-measure with respect to groundtruth skeletons. Our method, FSDS achieves the top result and shows both improved recall and precision at most of the precision-recall regime. See Table IV for more details about the other quantity (Avg Runtime) and citations to competitors.

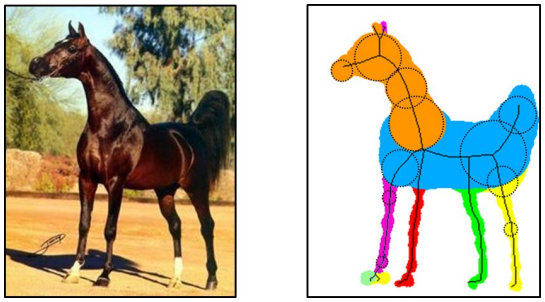

Fig. 14. Skeleton based object segmentation. Left: The original image. Right: The object segments reconstructed from the skeleton with scales. Different object segments are marked in different colors. The dashed circles are sampled maximal disks.

C. Object Segmentation

We can use the predicted scale for each skeleton pixel to segment the foreground objects in images. For each skeleton pixel x_j, let $\hat{s}_j$ be its predicted scale, then for a skeleton segment $\{x_j, j = 1, \ldots, N\}$, where N is the number of the skeleton pixels in this segment, we obtain a object segment mask by $\mathcal{M} = \bigcup_{j=1}^{N} D_j$, where D_j is the disk of center x_j and diameter $\hat{s}_j$. Fig. 14 illustrates an example of object segments obtained by the above process. The more accurate the predicted scales are, the more better segmentation results. Therefore, evaluating the object segmentation results, not only can we validate the performance for skeleton extraction, but the potential usefulness of the obtained skeletons for high level vision tasks can be demostrated.

1) Evaluation Protocol: Following [41]–[43], we evaluate object segmentation results by assessing their consistency with the groundtruth object segmentation. Two evaluation metrics

Fig. 13. Illustration of skeleton extraction results on WH-SYMMAX [28] 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.

are adopted here. One is the *F-measure* [43], which calculates the average best F-score between the groundtruth object segments and the generated segments (for each groundtruth object segment, find the generated one with highest F-score, then these F-scores are averaged over the whole dataset). The other is the *Covering* metric [41], [42], which calculates the average best overlapping score between groundtruth object segments and generated segments, weighted by the object size. Note that, these segmentation method generally produce multiple segments. Indeed the graph cut based methods generates hundreds of segments. Hence we prefer methods with higher *F-measure*/*Covering* but using fewer segments. We also report the average number of segments (*Avg num segments*) per image for each method.

2) Performance Comparison: We compare the object segmentation results of LMSDS with those of other skeleton based methods (Levinshtein's method [16], Lee's method [17], MIL [15] and FSDS [22]), those of graph cut based methods (Shape Sharing [41] and CPMC [42]) and that of a deep learning based segmentation method (FCN [20]). To obtain object segments reconstructed from skeletons, we threshold the thinned skeleton map (after non-maximal suppression) into a binary one. Thresholds were optimized over the whole dataset according to the F-measures for localization. FSDS does not explicitly predict skeleton scale, but we can estimate a coarse scale for each skeleton pixel according to the receptive field sizes of the different stages. For each skeleton pixel x_j, the scale predicted by FSDS is $\hat{s}_j = \sum_{i=1}^{M} r_i \Pr(z_j = i|X; \mathbf{W}*, \mathbf{\Phi}*, \mathbf{h}_0*)$. FCN was originally used for semantic segmentation (multi-class classification) in [20]. Here, we use it for foreground object segmentation (binary classification): Foreground objects have label "1" and background have label "0". We finetune the FCN-8s model released in [20] on our datasets to obtain foreground object segmentation.

We conduct the object segmentation experiments on SK-LARGE and WH-SYMMAX and evaluate the results according to the segmentation groundtruths provided by MS COCO [23] and Weizmann Horse [35], respectively. The quantitative results on these two datasets are summarized in Table VI and Table VII, respectively. LMSDS achieves significant higher *F-measure*/*Covering* than others, except for the result of CPMC on SK-LARGE. However, CPMC has a clear disadvantage compared with LMSDS: LMSDS only generates about 2 segments per image while CPMC produces 100 times more segments per image, moreover most CPMC segments fires on the background. Then, as can be seen from the qualitative results illustrated in Fig. 15 and Fig. 16[3], we find that CPMC misses some significant parts; FCN-8s is usually unable to ensure smoothness between similar pixels, and spatial and appearance consistency of the segmentation output; FSDS often generates much "fatter" bodies due to inaccurate scale predication; LMSDS produces better segmentation outputs, thanks to the learned scale regressors. Note that even the narrow gap between the tail and the leg of the last horse in Fig. 16 can be obtained by LMSDS.

D. Object Proposal Detection

To illustrate the potential of the extracted skeletons for object detection, we performed an experiment on object proposal detection. Let h_B^E be the objectness score of a bounding box

[3]Since the graph cut based method (CPMC) generates a large number of segments, we only show the one with the maximum overlap between the groundtruth segment. For others, we show the whole detected foreground segments.

TABLE VI
OBJECT SEGMENTATION PERFORMANCE COMPARISON BETWEEN DIFFERENT METHODS ON SK-LARGE.

Method	F-measure	Covering (%)	Avg num segments
Lee [17]	0.496	33.8	210.5
MIL [15]	0.268	27.5	8.4
Shape Sharing [41]	0.854	75.4	716.2
CPMC [42]	0.896	81.8	287.0
FCN-8s [20]	0.840	74.2	3.8
FSDS (ours)	0.814	69.1	2.0
LMSDS (ours)	0.873	78.1	2.1

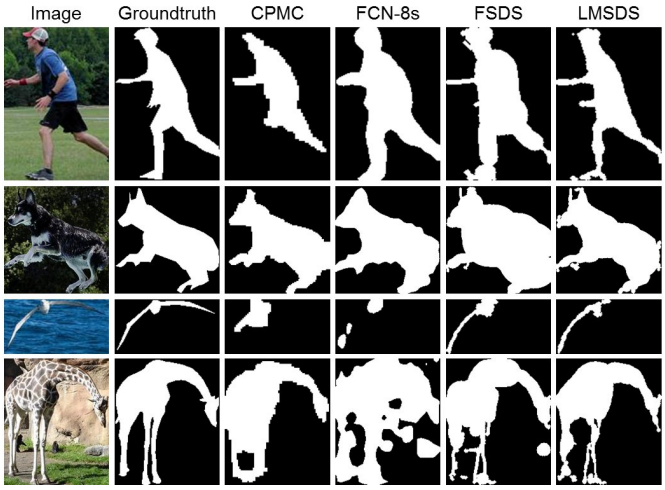

Fig. 15. Illustration of object segmentation on SK-LARGE for several selected images.

TABLE VII
OBJECT SEGMENTATION PERFORMANCE COMPARISON BETWEEN DIFFERENT METHODS ON WH-SYMMAX.

Method	F-measure	Covering (%)	Avg num segments
Lee [17]	0.597	43.4	253.0
MIL [15]	0.278	30.7	8.2
Shape Sharing [41]	0.857	75.4	879.8
CPMC [42]	0.887	80.1	511.2
FCN-8s [20]	0.823	72.1	2.3
FSDS (ours)	0.838	72.5	1.7
LMSDS (ours)	0.902	82.4	1.3

B obtained by EdgeBoxes [44], we define our objectness score by $h_B = \frac{\bigcup_{\forall \mathcal{M} \cap B \neq \emptyset}(B_{\mathcal{M}} \cap B)}{(\bigcup_{\forall \mathcal{M} \cap B \neq \emptyset} B_{\mathcal{M}}) \bigcup B} \cdot h_B^E$, where $\mathcal{M}$ is a part mask reconstructed by a detected skeleton segment and $B_{\mathcal{M}}$ is the minimal bounding box of $\mathcal{M}$. Let LMSDS+EdgeBoxes and FSDS+EdgeBoxes denote the scoring methods based on the skeletons obtained by LMSDS and FSDS, respectively. As shown in Fig. 17, LMSDS+EdgeBoxes achieves a better object proposal detection result than EdgeBoxes and FSDS+EdgeBoxes.

V. CONCLUSION

We proposed a new network architecture, which is a fully convolutional network with multiple multi-task scale-associated side outputs, to address the unknown scale problem in skeleton extraction. By studying the relationship between the receptive field sizes of the sequential scale-associated side outputs in the network and the skeleton scales they capture, we showed the importance of our proposed scale-associated side outputs for (1) guiding multi-scale feature learning, (2) fusing scale-specific responses from different stages and (3) training with multi-task loss to perform both skeleton localization and scale prediction. The experimental results demonstrate the effectiveness of the proposed method for skeleton extraction from natural images. It achieves significant improvements over the alternatives. We performed additional experiments on applications, such like object segmentation and object proposal detection, which verified the usefulness of the extracted skeletons in object detection.

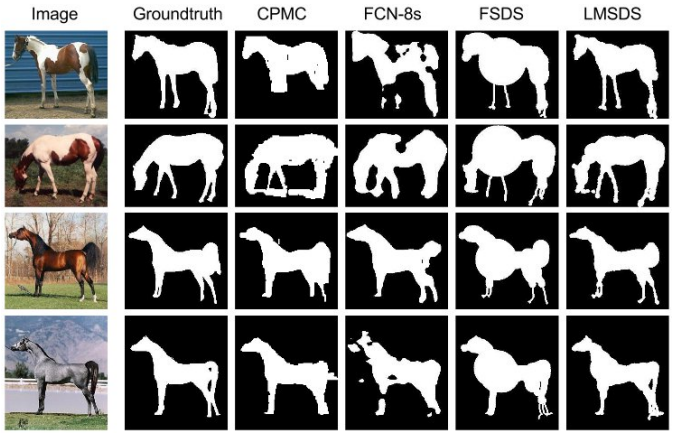

Fig. 16. Illustration of object segmentation on WH-SYMMAX [28] for several selected images.

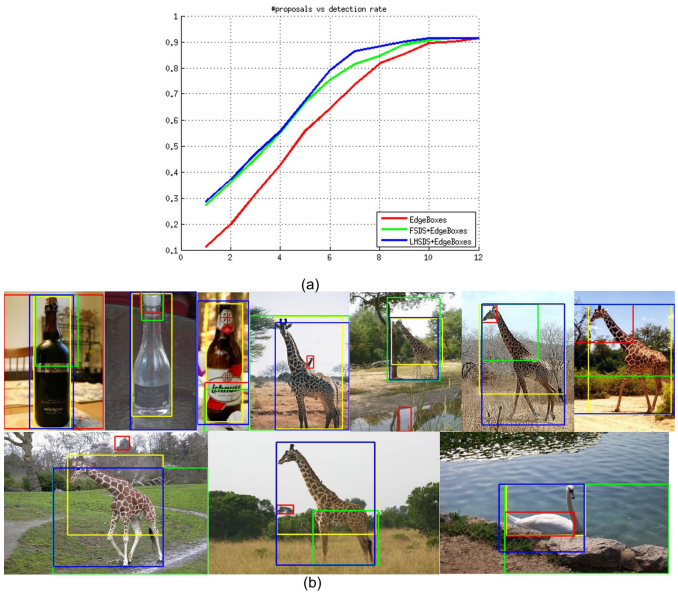

Fig. 17. Object proposal results on ETHZ Shape Classes [45]. (a) The detection rate curve (IoU = 0.7). (b) Examples. Groundtruth (yellow), the closest proposal to groundtruth of Edgebox (red), FSDS+EdgeBoxes (green) and LMSDS+EdgeBoxes (blue).

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 61672336 and 61573160, in part by "Chen Guang" project supported by Shanghai Municipal Education Commission and Shanghai Education Development Foundation under Grant 15CG43, in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DoI/IBC) contract number D16PC00007, and in part by Office of Naval Research N00014-15-1-2356. We thank NVIDIA Corporation for providing their GPU device for our academic research.

REFERENCES

[1] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?" in *Proc. CVPR*, 2010, pp. 73–80.

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

[3] 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*, vol. 94, no. 2, pp. 215–240, 2011.

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

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

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

[7] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker, "Shock graphs and shape matching," *International Journal of Computer Vision*, vol. 35, no. 1, pp. 13–32, 1999.

[8] T. B. Sebastian, P. N. Klein, and B. B. Kimia, "Recognition of shapes by editing their shock graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 550–571, 2004.

[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*, vol. 69, no. 2, pp. 203–222, 2006.

[10] X. Bai and L. J. Latecki, "Path similarity skeleton graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 7, pp. 1282–1292, 2008.

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

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

[13] T. Lindeberg, "Edge detection and ridge detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 117–156, 1998.

[14] Q. Zhang and I. Couloigner, "Accurate centerline detection and line width estimation of thick lines using the radon transform," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 310–316, 2007.

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

[16] A. Levinshtein, S. J. Dickinson, and C. Sminchisescu, "Multiscale symmetric part detection and grouping," in *Proc. ICCV*, 2009, pp. 2162–2169.

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

[18] N. Widynski, A. Moevus, and M. Mignotte, "Local symmetry detection in natural images using a particle filtering approach," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5309–5322, 2014.

[19] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. ICCV*, 2015, pp. 1395–1403.

[20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. CVPR*, 2015, pp. 3431–3440.

[21] C. Lee, S. Xie, P. W. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Proc. AISTATS*, 2015.

[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 *Proc. CVPR*, 2016.

[23] 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.

[24] X. Bai, L. J. Latecki, and W. Liu, "Skeleton pruning by contour partitioning with discrete curve evolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 449–462, 2007.

[25] T. Lindeberg, "Scale selection properties of generalized scale-space interest point detectors," *Journal of Mathematical Imaging and Vision*, vol. 46, no. 2, pp. 177–210, 2013.

[26] P. Majer, "On the influence of scale selection on feature detection for the case of linelike structures," *International Journal of Computer Vision*, vol. 60, no. 3, pp. 191–202, 2004.

[27] T. Liu, D. Geiger, and A. L. Yuille, "Segmenting by seeking the symmetry axis," in *Proc. ICPR*, 1998, pp. 994–998.

[28] 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*, vol. 52, pp. 266–278, 2016.

[29] X. Ren, "Multi-scale improves boundary detection in natural images," in *Proc. ECCV*, 2008, pp. 533–545.

[30] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 8, pp. 1558–1570, 2015.

[31] 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*, 2015, pp. 3982–3991.

[32] 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*, 2001, pp. 416–425.

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

[34] S. Lee and Y. Liu, "Curved glide-reflection symmetry detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 266–278, 2012.

[35] E. Borenstein and S. Ullman, "Class-specific, top-down segmentation," in *Proc. ECCV*, 2002, pp. 109–124.

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

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

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

[39] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *CoRR*, vol. abs/1409.0575, 2014.

[41] J. Kim and K. Grauman, "Shape sharing for object segmentation," in *Proc. CVPR*, 2012, pp. 444–458.

[42] J. Carreira and C. Sminchisescu, "CPMC: automatic object segmentation using constrained parametric min-cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1312–1328, 2012.

[43] S. Alpert, M. Galun, A. Brandt, and R. Basri, "Image segmentation by probabilistic bottom-up aggregation and cue integration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 315–327, 2012.

[44] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. ECCV*, 2014, pp. 391–405.

[45] V. Ferrari, T. Tuytelaars, and L. J. V. Gool, "Object detection by contour segment networks," in *Proc. ECCV*, 2006, pp. 14–28.

Wei Shen received his B.S. and Ph.D. degree both in Electronics and Information Engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2007 and in 2012. From April 2011 to November 2011, he worked in Microsoft Research Asia as an intern. In 2012, he joined School of Communication and Information Engineering, Shanghai University as a faculty. He is currently visiting Department of Computer Science, Johns Hopkins University. His current research interests include computer vision, deep learning and shape based object recognition.

Zhao Kai received his B.S. in Communication and Information Engineering from Shanghai University, Shanghai, China, in 2014. He is currently a master student of the School of Communication and Information Engineering, Shanghai University. His research interests include computer vision and deep learning.

Yuan Jiang received his B.S. in Communication and Information Engineering from Shanghai University, Shanghai, China, in 2015. Now he is a master student of the School of Communication and Information Engineering, Shanghai University. His research interests include computer vision, pattern recognition and deep learning.

Yan Wang received her B.S. degree from the Huazhong University of Science and Technology (HUST), China, in 2011, and completed her Ph.D. degree in Nanyang Technological University, Singapore, in 2016. She was a research assistant of The Ohio State University, U.S.A, in 2016. She is now a postdoc in John Hopkins University. Her current research interests include computer vision, object recognition, machine learning, cognitive science.

Xiang Bai received the B.S., M.S., and Ph.D. degrees from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2003, 2005, and 2009, respectively, all in electronics and information engineering. He is currently a Professor with the Department of Electronics and Information Engineering, HUST, where he is also the Vice Director of the National Center of Anti-Counterfeiting Technology. His research interests include object recognition, shape analysis, scene text recognition, and intelligent systems.

Alan Yuille received his B.A. in mathematics from the University of Cambridge in 1976, and completed his Ph.D. in theoretical physics at Cambridge in 1980. He then held a postdoctoral position with the Physics Department, University of Texas at Austin, and the Institute for Theoretical Physics, Santa Barbara. He then became a research scientists at the Artificial Intelligence Laboratory at MIT (1982-1986) and followed this with a faculty position in the Division of Applied Sciences at Harvard (1986-1995), rising to the position of associate professor. From 1995-2002 he worked as a senior scientist at the Smith-Kettlewell Eye Research Institute in San Francisco. From 2002-2016 he was a full professor in the Department of Statistics at UCLA with joint appointments in Psychology, Computer Science, and Psychiatry. In 2016 he became a Bloomberg Distinguished Professor in Cognitive Science and Computer Science at Johns Hopkins University. He has won a Marr prize, a Helmholtz prize, and is a Fellow of IEEE.