Fusing Image and Segmentation Cues for Skeleton Extraction in the Wild

Xiaolong Liu[1]*, Pengyuan Lyu[1]*, Xiang Bai[1]†, Ming-Ming Cheng[2]
[1]Huazhong University of Science and Technology, [2]Nankai University
liuxl@hust.edu.cn, lvpyuan@gmail.com, xbai@hust.edu.cn, cmm@nankai.edu.cn

Abstract

Extracting skeletons from natural images is a challenging problem, due to complex backgrounds in the scene and various scales of objects. To address this problem, we propose a two-stream fully convolutional neural network which uses the original image and its corresponding semantic segmentation probability map as inputs and predicts the skeleton map using merged multi-scale features. We find that the semantic segmentation probability map is complementary to the corresponding color image and can boost the performance of our baseline model which trained only on color images. We conduct experiments on SK-LARGE dataset and the F-measure of our method on validation set is 0.738 which outperforms current state-of-the-art significantly and demonstrates the effectiveness of our proposed approach.

1. Introduction

Skeleton is an important cue to describe the shape of objects and has been studied and applied to many fields in computer vision, such as object detection [8, 2], pose estimation [21], text recognition [27] and action recognition [5] *etc.* since 1970s [15].

The early methods proposed to extract skeleton are generally applied to binary images and are used to recognize and retrieve shapes [22]. However, those methods have difficulty extracting skeletons from natural images unless foreground masks or contour are well-segmented. Compared to binary images, natural images are more challenging to extract skeleton due to following reasons: 1) The background in natural images can be very complex. Except for some elements in the background that are very similar to objects, the non-uniform illumination, shadow, occlusion, and noise in images will also make the method fail. 2) The appearance of objects varies in colors, textures. 3) The sizes of objects are diversified. Objects may have different sizes in different parts, which determines the variance of skeleton scales.

*Equal contribution
†Corresponding author

Some methods have been proposed to extract skeletons in natural images. [10, 12, 25, 23] use low-level features such as gradient intensity map, super-pixels and hand-designed features *etc.* to extract skeletons. Those methods achieve good performance in the simple scene but often fail to handle complex images.

With the development of deep learning, Convolutional Neural Networks (CNNs) have been successfully applied to various vision tasks, such as image classification[24], semantic segmentation[4], edge / contour detection[18, 26, 14], *etc.*. Recently, some CNN-based methods for skeleton-extraction are also proposed to enjoy the benefit of strong discriminatory power[20, 19, 11, 9]. In [20, 19], Shen *et al.* propose a fully convolutional networks based on HED [26] and formulate the skeleton extraction as an image-to-image translation. To cope with the scale variance problem, this paper designs scale-associated ground-truth to train the proposed model. In [11], Ke *et al.* propose a model which learns the error between side outputs and groundtruth to ease the problem of fitting complex outputs.

In this paper, we also build a fully convolutional network to extract skeletons based on several existing models. To handle the scale diversity problem, we follow the network architecture of U-Net [16] and FPN [13] which use a top down network to merge features from top stage to bottom stage, layer by layer. The merged features contain multi-scale information and are robust to scale variation of objects. To simplify the complex groundtruth fitting problem, we train a model which learns to combine image and segmentation cues for skeleton extraction. We use the Deeplab model[4] to obtain the corresponding segmentation probability map of an image, and then pass the original image and the segmentation probability map to our two-stream network in parallel to predict skeleton. The experimental results show that the semantic segmentation probability maps are complementary to the corresponding color images and can boost the performance of our baseline model which trained only on original color images.

The contributions of this paper are as follows:

1) We use multi-scale fusion features which can adaptively the skeleton of objects with sizes from small to large.

2) We build a two-stream network that can take semantic segmentation probability maps to provide complementary information to the original color images for skeleton extraction.

3) We achieve state-of-the-art performance on the evaluation benchmark. The F-measure of our method on SK-LARGE is 0.738 which outperforms the alternatives significantly.

2. Our Approach

In this section, we give a detailed description of our approach, including our base network, the two-stream network, which is very important to capture complementary information for skeleton localization, and learning strategies.

2.1. Base Network

Network architectures are of great importance in the design of deep CNNs. Recently, VGG16 network[24], a very deep convolutional neural network (DCNN), has been widely used and proved to be effective in a variety of vision tasks. The original VGG16 network is composed of 13 convolutional layers and 3 fully connected layers. The convolutional part can be divided into 5 stages, and each stage is ended with a 2-stride down sampling layer. Previous literatures also demonstrate that fine-tuning deep neural networks pre-trained on large scale image classification dataset can achieve better performance compared with training from scratch. We therefore build our skeleton extraction network upon VGG-16 and make the following modifications:

- All fully connected layers are removed to fit the fully convolutional design.
- Following Deeplab[4], we discard the last two down-sampling layers so that the feature maps in the last *conv* stage have larger spatial resolution and are therefore better for precise skeleton localization. The *conv5* layers are also replaced with dilated convolution layers to compensate the loss of receptive fields caused by the above modification.
- We extend the fully convolutional network to an encoder-decoder model inspired by U-Net[16] and FPN[13]. The existing 5-stage convolutional network serves as an encoder, where image features with different scales and levels are extracted in different stages. The decoder is used to recover the spatial resolution and more importantly, to merge the multi-scale features of different stages from coarse to fine. To be concrete, in each stage, we fuse the decoder feature from the coarser stage (if available) and the encoder feature from the current stage (i.e. the output of the last convolution layer) by concatenation, and the result is made as the decoder feature in current stage. As the decoder goes deeper, the combined features become richer.

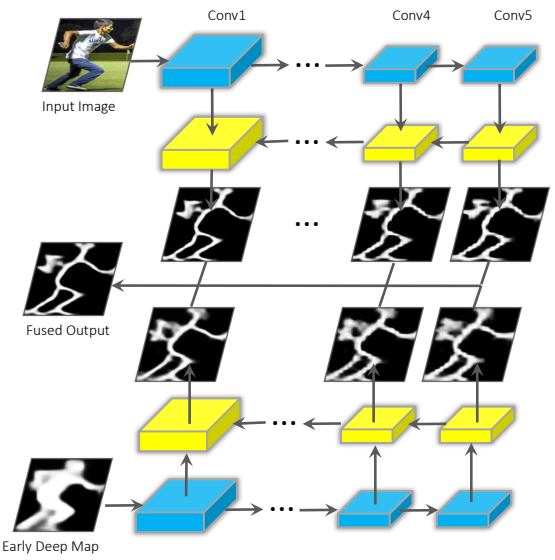

Figure 1. Architecture of the two-stream network. We input RGB images and segmentation probability maps to predict skeletons.

Similar to HED[26], side-output is introduced to predict skeleton maps from each stage of the decoder. These side-outputs are weighed linearly to generate a fusion output. All side-outputs and fusion output are supervised to train the model holistically. The resulting encoder-decoder network is set as the baseline of our work, and further used to build the two-stream network, as shown in Figure 1 and described in the next subsection.

2.2. Two-stream Network

Theoretically, segmentation results can serve as a powerful cue for skeleton extraction. Early works[3, 1] in skeletonization have tried to compute medial axes from binary images, where foreground masks are off-the-shelf. To extract skeletons from color images, these methods rely on a image segmentation step as pre-processing. Due to the sensitivity of skeleton to object boundary and region connectivity, the segmentation mask need to be of high quality. Otherwise, errors caused in the segmentation stage will damage the skeleton detection results. However, we believe that even not good enough segmentation results can still be helpful if used in a proper way. On one hand, as the original image contains complete information for skeleton extraction, errors in segmentation can be compensated in a way, if the two kinds of data are combined. On the other hand, the segmentation result suppresses most of background clutter and texture inside objects, which make it easier to handle complex scene in an image. We therefore develop a two-stream

network that takes both a natural image and the corresponding segmentation prediction as input, and that is capable of learning complementary information in an end-to-end manner.

As shown in Figure 1, we duplicate our base network described in Sec.2.1 to a unified one with two parallel branches. The two branches have independent parameters since they are designated to learn from features with different semantic levels. We refer to the one with image input as *im* branch and the other *seg* branch. The segmentation prediction is a one-channel probability map produced by the Deeplab[4] model fine-tuned for foreground (two class) segmentation. To fit the existing base network and share the same initial weights as the *im* branch, the probability map is rescaled to a [0, 255] range and converted to a 3-channel image. The two types of input data go through respective path and produce side-outputs of either branch. Side-outputs of each branch are fused in the same way as HED[26] (not shown in Figure 1 for cleanness). The fused outputs are called *imfuse* and *segfuse*. In order that the two branches can be complementary and jointly trainable, we propose to further weigh all side-outputs and produce a *finalfuse* output. With deep supervision on the *finalfuse* map, the fusion weight is learned automatically and gradients can be propagated back to each *conv* stage in either branch during training.

2.3. Learning

As the foreground object segmentation results are needed by the parallel network, we should build a binary segmentation model first. We choose the Deeplab model[4] for its good performance. Since we only need to segment some specific kinds of foreground objects and treat them equally, we modify the original Deeplab model to perform 2-class segmentation and fine-tune it on our dataset. Though the skeleton dataset doesn't provide object segmentation annotations, we can recover the foreground masks with the annotated medial points and radii of the corresponding maximal disks according to the MAT invertibility property[3]. The generated ground truth masks are then used to train the segmentation model.

Once the fine-tuning procedure is finished, we pass all training and testing images to the trained model to obtain segmentation predictions. The predictions are added to the original skeleton detection data set. We denote the extended training set by $\{(X^{(n)}, Y^{(n)}, S^{(n)}), n = 1, ..., N\}$ where $X_n = \{x_j^{(n)}, j = 1, ..., |X^{(n)}|\}$ is a raw input image, and $Y_n = \{y_j^{(n)}, j = 1, ..., |Y^{(n)}|\}(y_j^{(n)} \in \{0, 1\})$ and $S_n = \{x_j^{(n)}, j = 1, ..., |S^{(n)}|\}$ are its corresponding binary groundtruth skeleton map and foreground object probability map. For notation simplicity, we drop the superscript in later descriptions. Supposing each branch in the parallel network has M side-outputs (fusion outputs not included), fusion of the side-outputs in either branch is computed. To connect the two branches, we additionally stack the $2M$ side-outputs and compute the *finalfuse* output by linear weighing. See HED[26] for the detailed computation of fusion output. For each side-output or fusion output, class-balanced cross-entropy loss is computed, like HED[26],

$$\begin{aligned}\mathcal{L}(\mathbf{W}) = &-\beta \sum_{j \in Y_+} \log P\left(y_j = 1 | X, S; \mathbf{W}\right) \\ &-(1-\beta) \sum_{j \in Y_-} \log P\left(y_j = 0 | X, S; \mathbf{W}\right),\end{aligned} \tag{1}$$

where $\mathbf{W}$ represent all learnable parameters. The multiplier $\beta = |Y_+|/|Y|$ is used to handle the imbalance of numbers of positive / negative samples. Y_+ and Y_- denote the skeleton and background sets of the ground-truth Y, respectively. In the training process, sum of all loss functions is minimized to obtain the optimal parameters, including the fusion weights. In the testing phase, given an input image X and the corresponding segmentation probability map S, the skeleton prediction map is the activation of the final fusion output

$$\hat{Y}^{(final)} = \{\Pr(y_j = 1 | X, S; \mathbf{W}^*), j = 1, ..., |\hat{X}|\}, \tag{2}$$

where $\mathbf{W}^*$ is the set of learned parameters in the network.

3. Experiments

In this section we discuss the implementation details and evaluate our approach on the SK-LARGE dataset.

3.1. Implementation Details

Our implementation can be divided into 3 parts. First, we fine-tune object segmentation model[4] pre-trained on PASCAL VOC 2012[7]. Next, we train our base network on RGB images. Finally, the two-stream network is fine-tuned from an initialization with learned weights of the base network.

a) Data preprocessing. To generate more training data, strong data augmentation is employed in this work. Like [19], we rotate training image to 4 degrees (0° ,90°,180°,270°), and flip each one with different axes (horizontal, vertical, no flip), then resize images to different resolution (0.8, 1.0, 1.2). In order to train the object segmentation model, foreground object mask for each training image is also extracted, by simply filling the maximal disks centered at each skeleton point, as the radius annotations are provided.

b) Object segmentation. We use the best model among Deeplab-v2 [4] variants and finetune it with the training images and generated groundtruth masks. We keep the same

	baseline	imfuse	segfuse	final
val	0.717	0.721	0.699	**0.738**
test*	0.704	0.713	0.692	**0.731**
	MIL[25]	HED[26]	FSDS[20]	LMSDS[19]
test*	0.293	0.497	0.633	0.649

Table 1. The results of our method and the alternatives. "val" means the model is evaluated on the validation set which provided by the ICCV'17 Workshop Symmetry challenges. And "test*" means the model is evaluated on the test set which is split by [17].

Figure 2. Illustration of skeleton extraction results on the SK-LARGE dataset for several selected images. Results of LMSDS[17] are directly collected from its paper. Compared with LMSDS, our method can produce cleaner and smoother results. Zoom out to see better.

settings as [4] except the following changes:(1) Maximal number of iterations is set to 2500. (2) CRF is not applied. (3) The classifier in the network is modified to a binary one.

c) Skeleton extraction. For both the base and the two-stream network, the hyper parameters (and their values) include: mini-batch size (1), base learning rate (1e-6), the learning rate of each side-output layer (1e-7), the learning rate of each weighed fusion layer (1e-9), loss weight for each classifier output (1), momentum (0.9), weight decay (2e-4), initialization of side-output filters(0), initialization of weighted fusion layers ($1/n$, where n is the number of input maps to be fused), maximum number of training epochs (15). For model evaluation, we use the official evaluation code and compute F-measure on the non-maximal suppressed[6] skeleton probability maps.

3.2. Results

We conduct experiments on SK-LARGE and evaluate our model on two splits. The first one is provided by ICCV'17 workshop symmetry detection challenges and named "val". In order to facilitate the comparison with other methods, we also evaluate our model on a subset of SK-LARGE which split by [17]. We refer to this split as "test*" . The quantitative results and visualized skeleton detection results are shown in Table 1 and Figure 2 respectively.

3.2.1 The results of our baseline

The baseline is trained only on natural images. Thanks to the high resolution of features in conv5 to ensure precise localization and the fusion of multi-scale features in decoder to provide richer features, our baseline network achieves very good performance (F-measure 0.717 on "val" split and 0.704 on "test*" split) and has a great improvement compared with LMSDS[19] (F-measure 0.649 on "test*" split) [17] which shows that our base network is fairly powerful.

3.2.2 The effect of segmentation map

We train our model on both natural images and the corresponding segmentation maps using our parallel network. From the results in Table 1, we can observe that:

1) The features from image branch and segmentation branch are complementary. By a learnable weighted fusion of side-outputs on both branches, the performance of the image branch is improved significantly on both splits.

2) The segmentation branch is positive to the learning of image branch. With the aid of segmentation map, performance of the image branch is better that the baseline.

3) Our model can transform a segmentation probability map to a precise skeleton map, though the performance of the segmentation branch is not as good as the image branch. It is not surprising since the input data is only rough segmentation prediction, of which the quality greatly affect the skeleton detection result.

4. Conclusion

In this paper, we propose a two-stream fully convolutional network that takes advantage of semantic segmentation probability map and multi-scale deep features to address the challenging problems: complex background and various size of objects in natural images skeleton extraction. We train and evaluate our model on SK-LARGE and achieve impressive result (F-measure: 0.738), and exceed other state-of-the-arts greatly which proves the effectiveness of our approach.

Acknowledgement

This work was supported by NSFC 61573160.

References

[1] X. Bai, L. J. Latecki, and W.-Y. Liu. Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE*

transactions on pattern analysis and machine intelligence, 29(3), 2007.

[2] X. Bai, X. Wang, L. J. Latecki, W. Liu, and Z. Tu. Active skeleton for non-rigid object detection. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 575–582. IEEE, 2009.

[3] H. Blum. Biological shape and visual science. 1. *Journal of Theoretical Biology*, 38(2):205–287, 1973.

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

[5] X. Chen and M. Koskela. Skeleton-based action recognition with extreme learning machines. *Neurocomputing*, 149:387–396, 2015.

[6] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1558–1570, 2015.

[7] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International journal of computer vision*, 61(1):55–79, 2005.

[9] C. Funk and Y. Liu. Symmetry recaptcha. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5165–5174, 2016.

[10] J.-H. Jang and K.-S. Hong. A pseudo-distance map for the segmentation-free skeletonization of gray-scale images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 18–23. IEEE, 2001.

[11] W. Ke, J. Chen, J. Jiao, G. Zhao, and Q. Ye. Srn: Side-output residual network for object symmetry detection in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[12] A. Levinshtein, C. Sminchisescu, and S. Dickinson. Multiscale symmetric part detection and grouping. *International journal of computer vision*, 104(2):117–134, 2013.

[13] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[14] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai. Richer convolutional features for edge detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[15] D. Marr and H. K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London B: Biological Sciences*, 200(1140):269–294, 1978.

[16] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[17] W. Shen, X. Bai, Z. Hu, and Z. Zhang. Multiple instance subspace learning via partial random projection tree for local reflection symmetry in natural images. *Pattern Recognition*, 52:306–316, 2016.

[18] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3982–3991, 2015.

[19] W. Shen, K. Zhao, Y. Jiang, Y. Wang, X. Bai, and A. Yuille. Deepskeleton: Learning multi-task scale-associated deep side outputs for object skeleton extraction in natural images. *IEEE Transactions on Image Processing*, 2017.

[20] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 222–230, 2016.

[21] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[22] 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.

[23] T. Sie Ho Lee, S. Fidler, and S. Dickinson. Detecting curved symmetric parts using a deformable disc model. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1753–1760, 2013.

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[25] S. Tsogkas and I. Kokkinos. Learning-based symmetry detection in natural images. In *European Conference on Computer Vision*, pages 41–54. Springer, 2012.

[26] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[27] Z. Zhang, W. Shen, C. Yao, and X. Bai. Symmetry-based text line detection in natural scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2558–2567, 2015.