Feature Hourglass Network for Skeleton Detection

Nan Jiang[†], Yifei Zhang[†‡], Dezhao Luo[†‡], Chang Liu[†], Yu Zhou[‡] and Zhenjun Han[†*]

[†]University of Chinese Academy of Sciences, Beijing, China
[‡]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{jiangnan18,liuchang615}@mails.ucas.ac.cn
{zhangyifei0115,luodezhao,zhouyu}@iie.ac.cn, hanzhj@ucas.ac.cn

Abstract

Geometric shape understanding provides an intuitive representation of object shapes. Skeleton is typical geometrical information. Lots of traditional approaches are developed for skeleton extraction and pruning, while it is still a new area to investigate deep learning for geometric shape understanding. In this paper, we build a fully convolutional network named Feature Hourglass Network (FHN) for skeleton detection. FHN uses rich features of a fully convolutional network by hierarchically integrating side-outputs with a deep-to-shallow manner to decrease the residual between the prediction result and the ground-truth. Experiment data shows that FHN achieves better performance compared with baseline on both Pixel SkelNetOn and Point SkelNetOn datasets.

1. Introduction

Geometric shape understanding provides an intuitive representation of object shapes, which can be used for foreground extraction, shape modeling, object proposal, *et al.* Even though deep learning approaches obtain great success for detection and segmentation tasks, it is still a new area to investigate deep learning for geometric shape understanding, especially for extracting topological and geometric information from shapes.

For geometric shape understanding, we focus on skeleton detection, *i.e.,* skeleton pixels for images and skeleton points for point clouds. With traditional methods, it usually takes morphological operation [12] and Multiple Instance Learning [15]. In [15], each pixel is treated as an instance bag considering multi-scale and multi-orientation. MIL is used to train a binary classifier to determine whether a pixel is on the skeleton curve. Recently, various structure of Convolutional Neural Networks (CNN) are proposed, which can naturally learn from low to high or high to low level features in a shallow to deep way. In [8], Lin *et al.* propose FPN, developing a topdown architecture with lateral connections, exploiting the inherent multi-scale, pyramidal hierarchy of deep convolutional networks to construct feature pyramids with marginal extra cost. He *et al.* propose SPPNet in [5], adopting spatial pyramid pooling and some additional feature transformations to generate a pool of feature maps with different sizes.

With the success of CNN, several deep learning approaches are proposed for skeleton detection [6, 9, 10, 11], which also deal with skeleton detection as a binary classification problem. In these approaches, the structure of convolutional neural networks almost consist of backbone network and side-output modules. The backbone is used to generate predictions for multi-scale, while the side-output modules are designed to integrate the predictions.

In this paper, we build a fully convolutional network named Feature Hourglass Network (FHN) for skeleton detection. Specifically, we choose feature hourglass network as the backbone. Experiment data shows that the proposed FHN improves skeleton detection performance compared with baseline both on Pixel SkelNetOn and Point SkelNetOn datasets [1, 3, 7, 13].

2. Methodology

In this section, we review various kinds of backbone and side-output of deep convolutonal networks with different multi-scale fusion strategies and introduce Feature Hourglass Network (FHN) in detail.

2.1. Multi-scale Architecture Review

Multi-scale is an essential property of skeleton, for example, the scale of the skeleton pixels on the leg of a horse are smaller than the pixels on the body. It impels us to employ all stages of one convolutional neural network as the receptive fields increases from shallow stage to deep stage.

[*]Corresponding author

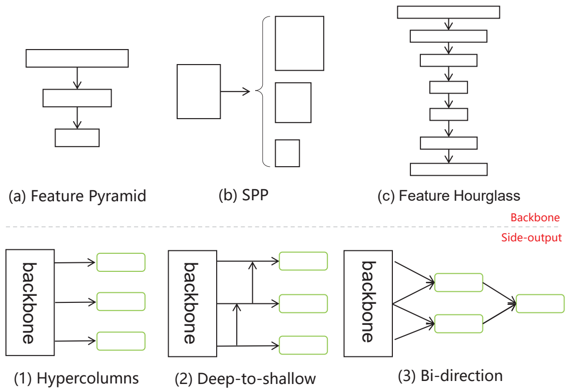

Figure 1. Illustration of multi-scale architecture. Including the structure of the backbone (top) and the fusion of the side outputs (bottom).

Roughly, the backbone with multi-scale is divided into three types: feature pyramid, spatial pyramid pooling, and feature hourglass, as shown in Fig. 1(a) to Fig. 1(c). Feature pyramid is a straightforward way for multi-scale feature extraction. The receptive field increases in the orientation from shallow to deep, which corresponding to scale from small to large. Spatial pyramid pooling takes downsampling with different kernel sizes, it will lose information for elaborate context. The feature hourglass not only takes the advantage of feature pyramid, but also increasing the range of scale with the same model size.

With different backbones, the side-output can be fused as shown in Fig. 1(1) to Fig. 1(3). Hypercolumns predict side-outputs for each stage and integrate them with weighted sum operation. Deep-to-shallow integrates side-outputs one by one so that the residual between the side-output and ground-truth decreases in order. The bi-direction integration manner adopts a different strategy to adapt scale varience.

2.2. Architecture of FHN

In this section, the backbone, a strategy of side-output integrating and learning of the proposed Feature Hourglass Network (FHN) is introduced in detail.

Backbone Construction: The backbone is built on VGGNet [14], Fig. 2. The first four stages of VGG are shown on the top of Fig. 2. It consists of 2, 2, 3 and 3 convolutional layers followed by a pooling layer for different stages, respectively.

The original architecture of VGG keeps the same structure for the fifth stage with 3 convolutional layers, as shown at the bottom-left of Fig. 2. For the proposed backbone, we delete the pooling layer following the fourth stage of VGG and insert deconvolutional layers among the convolutional layers of the fifth stage, which further improve the size of receptive field and output features, as shown at the bottom-right of Fig. 2. Dilation [2] is utilized to increase the receptive field which has the same size of model compared with the original VGG. Updating VGG backbone to FHN backbone, we add a prefix ’H-’ to name the FHN version of VGG based state-of-the-art networks, such as H-VGG, H-RCF, H-HIFI, H-RSRN and so on.

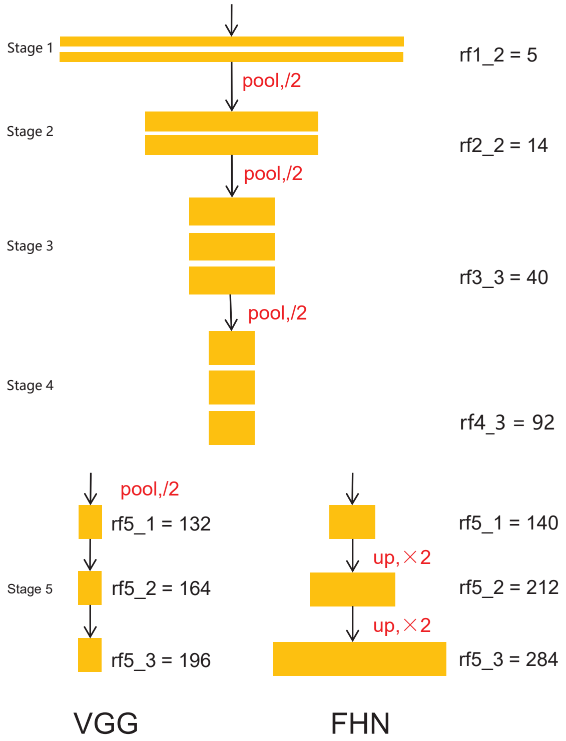

Figure 2. Comparison of the network backbone architecture between VGG and FHN. The first 4 stages are the same.

For geometric shape understanding condensing both local and global information about the shape, the receptive field size of the deep convolutional network has important influence on the effect of multi-scale feature fusion. FHN with larger receptive field size has more potential to adapt severe scale varied skeleton detection task.

Side-output Integrating: We follow RSRN [9] to integrate our side-output. 12 Residual Units (RU) are stacked to take full use of features from all convolutional layers. All side-outputs are then fused together to get the final output. The overall loss function of the proposed FHN is formulated as:

$$\mathcal{L} = \sum_{i=1}^{M} \mathcal{L}_{side}(h(X|\mathbf{W}, \theta_i), G) + \mathcal{L}_{fuse}(h(X|\mathbf{W}, \theta), G),$$

where $h(X|\mathbf{W}, \theta)$ is the result of side-output prediction. M, equal to 13, is the number of layers in the architecture of backbone. X, $\mathbf{W}$, θ_i, and G represent the input image, the parameter of backbone model, the parameter of the added simple 1×1 convolutional layer for the i-th layer of the backbone, and the ground-truth of final fused output, respectively.

Balanced entropy loss is used to compute the side loss and fuse loss, which is defined as:

$$\mathcal{L}(P, G) = -\beta \sum_{j \in Y_+} \log P_j 1(G_j = 1) \\ - (1 - \beta) \sum_{j \in Y_-} \log P_j 1(G_j = 0),$$

where P is the prediction $h(X|\Theta)$, the balancing weight $\beta = [\sum_{j=1}^{|x|} 1(G_j \neq 0)]/|X|$, and $1(.)$ is an indicator function.

We can obtain the optimal parameters $(\mathbf{W}^*, \Theta^*)$by a standard stochastic gradient descent (SGD):

$$(\mathbf{W}^*, \Theta^*) = \text{argmin}\ \mathcal{L}.$$

3. Experimental results

In this section, we discuss the implementation of FHN in detail and compare FHN with other approaches for skeleton detection. All the experiments are performed with NVIDIA GTX 1080Ti.

3.1. Implementation

Dataset: Pixel SkelNetOn and Point SkelNetOn are used to evaluate the proposed FHN.

Pixel SkelNetOn contains 1,727 binary images with size 256 × 256 pixels, splitting into 1,219 images for training, 242 images for validating, and 266 images for testing. The ground truth of Pixel SkelNetOn are the skeleton images which represent the skeletons corresponding to the shape images. Pixel SkelNetOn is focused on extracting the skeleton pixels from the shape of training images.

Point SkelNetOn consists of 1,727 shape point clouds with format *.pts*, splitting into 1,219 training point clouds, 242 validation point clouds, and 266 test point clouds. Ground truth for Point SkelNetOn are given as point clouds which represent the skeletons. Point SkelNetOn represents the full shape by point clouds, and focus on extracting the skeleton points from given point clouds. In order to use the same input form, every Point SkelNetOn data is drawn in a white background according to the corresponding coordinate before training. And then the results are converted to point clouds form before testing.

Protocol: The precision-recall (PR) metric with maximum F-measure is used to evaluate the performance of skeleton detection[15], as $F = \frac{2PR}{P+R}$.

The point skeleton extraction results are evaluated by using the symmetric Chamfer distance function, defined by:

$$Ch(A, B) = \frac{1}{|A|} \sum_{a \in A} \min_{b \in B} ||a - b||_2 + \frac{1}{|B|} \sum_{b \in B} \min_{a \in A} ||a - b||_2,$$

where A and B represent the skeleton point sets to be compared.

Backbone&sideout	Score(%)	Runtime (s)
VGG	12.71	**0.026**
H-VGG	**50.39**	0.059
H-RCF	59.12*	**0.244**
H-HIFI	64.56*	0.632
H-RSRN	**64.77***	0.396

Table 1. Comparison of backbones and our backbone with different side outputs on Pixel SkelNetOn validation dataset. * indicates score on test dataset.

Iteration	1-1	4-1	9-1	12-1	14-1
Aug	51.02	55.93	58.67	59.44	**59.61**
W/O aug	54.72	57.18	59.17*	58.72*	58.86*

Table 2. Performance of different iterations with augmentation and without augmentation. i-j means that the learning rate is set to 1e-6 for the first i*10k iterations and reduced to 1e-7 for the remaining j*10k iterations.

Data augmentation: In [6], Ke et al. discuss the data augmentation for medial axis training in deep learning approaches. We follow the data augmentation manner in [6], by rotating each image to 4 angles (0°, 90°, 180°, 270°) and flipping it with three different axes. Finally, we resize training images to 3 different scales (0.5, 1, 1.5).

Model Parameters: Following the setting of RSRN [9], we train FHN by fine-tuning the pre-trained 16-layer VGG net [14]. The hyper-parameters of FHN include: minibatch size (1), initial learning rate (1e-6 for Pixel SkelNetOn,1e-7 for Point SkelNetOn), loss-weight for each RU output (1.0), momentum (0.9), weight decay (0.0002), and maximum number of training iterations (150k for Pixel SkelNetOn, 80k for Point SkelNetOn). In the testing phase, a standard non-maximal suppression algorithm [4] is applied on the output map to obtain thinner skeleton.

3.2. Results

We measure performance in terms of *F1* score in Pixel SkelNetOn validation dataset, considering the factors of various backbone, different strategies of side-output integration, data augmentation, training strategy, Non-maximum Suppression (NMS), and threshold for binarization.

The performance comparison with different backbone and strategies of side-output integration is shown in Table 1. Our hourglass backbone achieves significant better F-measure compared with the original VGG from 12.71% to 50.39%. With the same hourglass backbone, our method gets the best F-measure of 64.77%.

Table 2 illustrates the performance comparison with dif-

Threshold	150	200	250
Score	3.23	2.90	**2.66**

Table 3. Performance of different thresholds with NMS on Point SKelNetOn validation dataset. Lower score indicates better performance.

Iteration	60k	80k	100k	120k
Score	2.71	**2.68**	2.70	2.75

Table 4. Performance of different iterations on Point SKelNetOn test dataset.

	Skeleton pixel(%)		Skeleton point	
Team	x2	Prisdl	RG	Prisdl
Val	75.82	63.25	2.26	2.40
Test	78.46	64.77	1.87	2.68

Table 5. Performance comparison of skeleton pixel detection and skeleton point datasets from leaderboard. Higher score indicates better performance on Pixel SkelNetOn dataset, while lower score indicates better performance on Point SkelNetOn dataset. Prisdl is ours.

ferent data augmentation and training strategies. With data augmentation, the best F-measure is achieved by training the network for 140k iterations with the initial learning rate being 1e-6 and another 10k iterations with 1e-7.

We compare the influence of NMS and the threshold for binarization in Tabel 3. It achieves the best score of 2.66 when NMS is taken and threshold is set as 250.

As shown in Table 4, we compare the performance with different training iteration numbers following the best setting in Tabel 3. Finally, we find FHN achieves the best performance in 80k iterations with the learning rate being 1e-7.

3.3. Evaluation of Skeleton Detection

The final performance on skeleton pixel and skeleton point is shown in Table 5.Through multiscale strategy, our proposed method scored 63.25% on Pixel SkelNetOn validation dataset, 64.77% on test dataset. And it scored 2.40 on Point SkelNetOn validation dataset, 2.68 on test dataset. In Pixel SkelNetOn challenge, four teams ranked above us. In Point SkeletOn challenge, we took the second place. Some qualitative results for skeleton pixel and skeleton point detection are shown in Fig. 3 and 4. Fig. 3 shows that H-RSRN has less noise compared with others. It can be seen that skeleton information can be obtained very well by point clouds from Fig. 4.

4. Conclusion

In this paper we propose FHN for geometric shape understanding. One can see that FHN significantly outperforms the baseline on both Pixel SkelNetOn and Point SkelNetOn datasets. Experimental results show that the proposed approach achieves the prediction score of 2.68 on test dataset, which is in the second place on the performance leader-board.

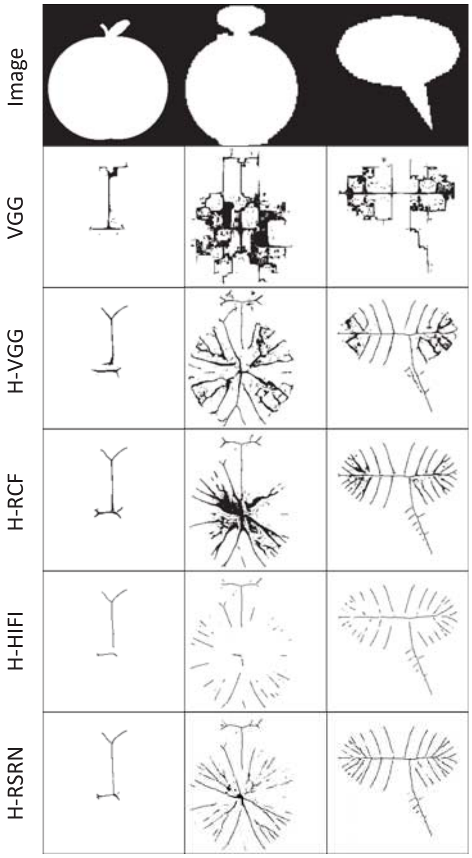

Figure 3. Skeleton detection on Point SkelNetOn.

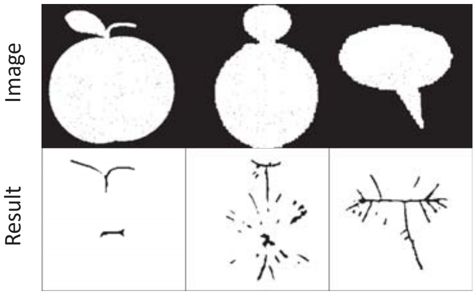

Figure 4. Skeleton detection on Point SkelNetOn.

Acknowledgement

The authors are very grateful to the support by the National Nature Science Foundation of China under Grants 61836012, 61671427 and 61771447, Huawei Innovation Research Program (HIRP) and the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDC02000000.

References

[1] Alexander M. Bronstein, Michael M. Bronstein, Alfred M. Bruckstein, and Ron Kimmel. Analysis of two-dimensional non-rigid shapes. *IJCV*, 2008.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *T-PAMI*, 40(4):834–848, 2018.

[3] Ilke Demir, Camilla Hahn, Kathryn Leonard, Geraldine Morin, Dana Rahbani, Athina Panotopoulou, Amelie Fondevilla, Elena Balashova, Bastien Durix, and Adam Kortylewski. SkelNetOn 2019 Dataset and Challenge on Deep Learning for Geometric Shape Understanding. *arXiv e-prints*, 2019.

[4] Piotr Dollár and C. Lawrence Zitnick. Fast edge detection using structured forests. *T-PAMI*, 37(8):1558–1570, 2015.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, pages 346–361, 2014.

[6] Wei Ke, Jie Chen, Jianbin Jiao, Guoying Zhao, and Qixiang Ye. SRN: side-output residual network for object symmetry detection in the wild. In *CVPR*, pages 1068–1076, 2017.

[7] Kathryn Leonard, Geraldine Morin, Stefanie Hahmann, and Axel Carlier. A 2d shape structure for decomposition and part similarity. In *ICPR*, pages 3216–3221, 2016.

[8] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944, 2017.

[9] Chang Liu, Wei Ke, Jianbin Jiao, and Qixiang Ye. Rsrn: Rich side-output residual network for medial axis detection. In *IEEE ICCV Workshop*, pages 1739–1743, 2017.

[10] Chang Liu, Wei Ke, Fei Qin, and Qixiang Ye. Linear span network for object skeleton detection. In *ECCV*, pages 136–151, 2018.

[11] Chang Liu, Fang Wan, Wei Ke, Zhuowei Xiao, Yuan Yao, Xiaosong Zhang, and Qixiang Ye. Orthogonal decomposition network for pixel-wise binary classification. In *CVPR*, 2019.

[12] Punam K. Saha, Gunilla Borgefors, and Gabriella Sanniti di Baja. A survey on skeletonization algorithms and their applications. *PRL*, 76:3–12, 2016.

[13] Thomas B Sebastian, Philip N Klein, and Benjamin B Kimia. Recognition of shapes by editing their shock graphs. *T-PAMI*, 26(5):550–571, 2004.

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[15] Stavros Tsogkas and Iasonas Kokkinos. Learning-based symmetry detection in natural images. In *ECCV*, pages 41–54, 2012.