SkeletonNet: Shape Pixel to Skeleton Pixel

Conference Paper · June 2019

DOI: 10.1109/CVPRW.2019.00156

Sabari Nathan
Couger Inc

SkeletonNet: Shape Pixel to Skeleton Pixel

Sabari Nathan
Couger Inc.
sabari@couger.co.jp

Priya Kansal
Couger Inc.
priya@couger.co.jp

Abstract

Deep Learning for Geometric Shape Understating has organized a challenge for extracting different kinds of skeletons from the images of different objects. This competition is organized in association with CVPR 2019. There are three different tracks of this competition. The present manuscript describes the method used to train the model for the dataset provided in the first track. The first track aims to extract skeleton pixels from the shape pixels of 89 different objects. For the purpose of extracting the skeleton, a U-net model which is comprised of an encoder-decoder structure has been used. In our proposed architecture, unlike the plain decoder in the traditional U net, we have designed the decoder in the format of HED architecture, wherein we have introduced 4 side layers and fused them to one dilation convolutional layer to connect the broken links of the skeleton. Our proposed architecture achieved the F1 score of 0.77 on test data.

1. Introduction

The extracted skeletons from the images are widely used in various areas like computer vision and image processing for optical character recognition [17], fingerprint recognition [28], motion detection [14], object tracking [13], etc. Skeletons are also widely used in life sciences for plant morphology [4]. Deep Learning for Geometric Shape Understanding at CVPR 2019 has organized SkelNetOn challenge. In this challenge, a pre-segmented image dataset with the corresponding skeleton representations in three tracks is provided [25]. The first track has posed the challenge of extracting the skeleton pixels from the given pre-segmented images [25][16][19][24]. We have approached this challenge as an edge detection problem and introduced a version of HED architecture in the decoder part of our proposed architecture. The rest of the sections of this manuscript describe the dataset, related work, methodology and results of the model used to secure 3rd place in the challenge.

2. Related Work

Skeleton extraction is a widely researched area in the last 10 years. However, the most recent works are mainly focused on the extracting skeleton from the RGB images [22][11], which involves segmentation or detection of the objects and extract the skeleton at the same time. Also, an extensive research is done either on edge detection [8][3][27][23] or segmentation [27][10] individually. These kinds of works do not suit fully to the present task. Some initial works are done on the extracting skeleton from the pre-segmented images [2][1][5] [9] which is similar to our task. However, most of these works are focused on the skeleton pruning to remove the unwanted branches rather than skeleton extraction. In the work done by [7], the authors introduced the boundary noise to avoid the uninformative branch creations. [15] used skeleton strength maps (SSM) which are calculated by the isotropic diffusion of the Euclidian distance transformation of binary images and their gradient. After calculating the SSM, they connected all the local maxima points of SSM with the shortest possible line to extract the skeletons. [6] approached the task of skeleton extraction as image generation model and used the generative adversarial network to extract the skeletons.

We have approached the present task as an edge detection problem and hence our work is more inspired by Holistically-nested Edge Detection (HED) model [26]. Similar to HED architecture, we have also fused the side layers into the final output layer. But to improve the performance of HED, instead of taking the output of convolution layers as side layers, we have introduced CS-SE layers at the end of each up-sampling layer and have considered the output of CS-SE layers as side layers. The detail of our approach is presented in section 3.1.

3. Dataset

The challenge is organized in two phases. In the development phase, 1219 images with their ground-truth for training and 242 images without ground-truth for validation are provided. In the final phase, a total of 266 test images are given. Participants are asked to submit their prediction for validation images in the development phase

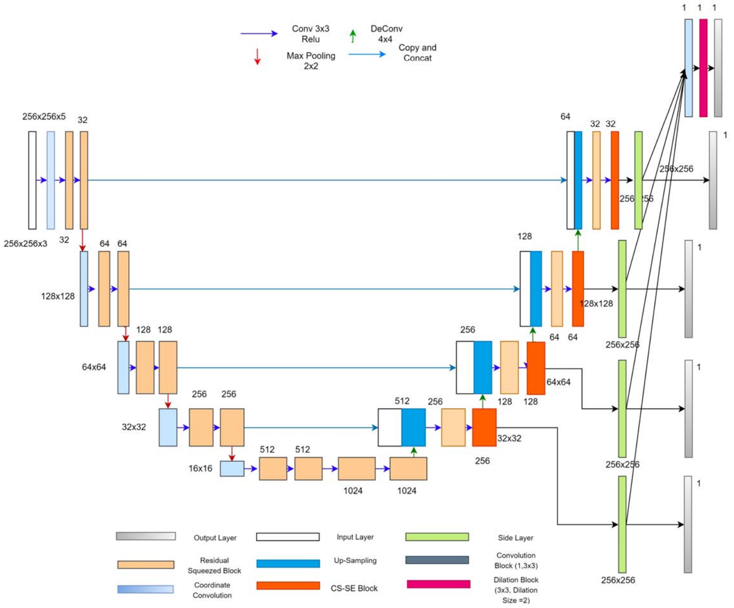

Figure 1: SkeletonNet: A detailed view of Proposed Architecture

and test images in the final phase. For the purpose of training, we have split the training images into 80:20. While splitting, we have ensured to split object-wise, so that both training and validation sets would have all the objects. After splitting the dataset into train and validation, we found that the data is quite imbalanced. It ranges from 1image to 58 images across the 89 objects in the dataset. Hence, we have augmented the train set (975) images into 1296 images. For the purpose of augmentation, image and mask rotation from -45 degree to +45 degree are used.

4. Method:

4.1 Details of the Architecture:

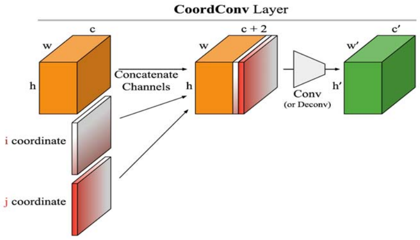

Figure 2: Coordinate convolutional layer as proposed in original paper

Unlike the plethora of classification and segmentation task, here we need to focus on the skeleton of images from the masks which is somewhat related to the problem of edge detection. In the process of extracting the skeleton from the mask of the objects, we have designed an encoder-decoder structure proposed in [20] with side layers inspired by HED architecture [26]. We have used a unique version of the HED architecture in the decoder part of U-net. The detail of the architecture is shown in Figure 1. As shown in Figure 1, before passing the image into the encoder, we have first passed the input image into a coordinate convolution layer as proposed by [18]. Coordinate convolutional layer helps the network to decide on the features related to translation invariance which further improves the generalization capacity of the model.

As suggested in the original paper, with the help of coordinate convolutional layer, spatial coordinates can be mapped with the coordinates in Cartesian space through the use of extra coordinate channels which gives the power to the model to use either complete or varying degree of translation features. Here, we have used the same two extra coordinate channels (i, j) which are suggested in the original paper. Figure 2 shows the detailed coordinate convolutional layer as given in the original paper. The coordinate channel i is a matrix in which row one is filled with all zeros, row 2 is all 1s, row three is all 2s and so on. Channel j is also similar to channel i but in this channel, the columns are filled with the numbers. Also, since we added two more channels, we have used a special residual squeezed block to extract the feature map in the encoder part (Figure 3). In our residual squeezed block, we have included the squeezed and excitation block to pass the output of the convolution layer and then have added this to the identity layer as the normal procedure of the residual block. The purpose of passing the output of residuals in the squeezed and excitation block is to prevent the overfitting caused by the extra feature maps. The squeezed and excitation block have adaptively weighed to all the feature maps [12]. As far as the decoder part is concerned, we have passed the output of up-sampling layer to the residual squeezed block. The output of the residual squeezed block is further passed to the channel squeeze and spatial excitation (CS-SE) block [21]. The CS-SE block slices its input corresponding to the spatial location (x,y) where, x ∈ {1,2, ….H} and y ∈ {1,2, ….W}. This spatial mapping has helped the network in concentrating the meaningful features over the weak features.

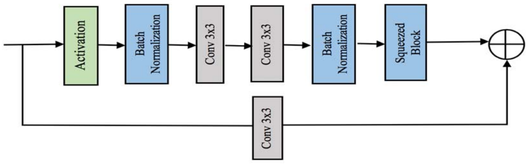

Figure 3: Residual Squeezed (RS) block used in the encoder and decoder part

As discussed before, we have used side layers inspired by the HED network. Total of 4 side layers are fused to the final output layer. The output of the fused layer is passed to the dilation layer to get the strongest features without losing the received resolution of the output of the fused layer. Further, side layers' output and the output of the final layer are then passed through a sigmoid layer individually under the supervision of ground truth (Figure 4). This approach helped us to connect the broken links of the skeleton predicted.

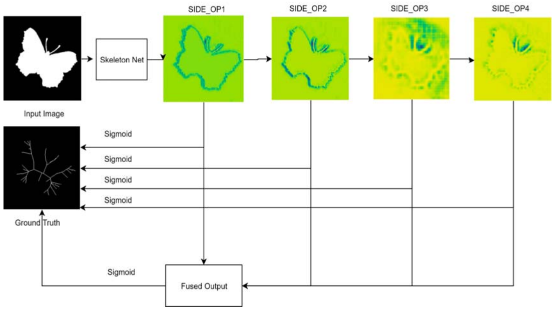

Figure 4: Side – Layers and Fused layer guidance with the Ground-truth images

4.2 Image Preprocessing

Images are divided by 255 to normalize the value of each pixel between 0 and 1.

4.3 Training

We have trained the network for five outputs which include the four side layers and one fused output layer for the skeletons of the input images. Adam optimizer is used to update the weights while training. The learning rate is initialized with 0.001 and reduced after 10 epochs to 10% if validation loss does not improve. The batch size is set to 4. The total epochs are set to 500. However, training is stopped early when the network started overfitting. The dataset is trained using Nvidia 1080 GTX GPU.

4.4 Loss Function:

We have proposed a novel yet simple loss function. Our loss function is the sum of binary cross-entropy and Dice Loss as defined in equation 1. The network is trained to minimize the Binary loss with sigmoid activation function.

$$Loss = L + DiceLoss \qquad \ldots (1)$$

Dice Loss is defined in equation (2) and L is cross-entropy loss defined in equation (3)

$$Dice\ Loss = 1 - \frac{2\sum_{i=0}^{k} y_i p_i + \varepsilon}{\sum_{i=0}^{k} y_i + \sum_{i=0}^{k} p_i + \varepsilon} \qquad \ldots . (2)$$

$$L = -\sum_{i=0}^{k}[y_i \log p_i + (1 - y_i)\log(1 - p_i)] \ldots (3)$$

where, y_i and p_i are the ground truth and the predicted skeleton images respectively. The coefficient ε is used to ensure the loss function stability by avoiding the zero value in the denominator of dice loss.

5. Results:

Table 1: Results of the side layer, fused layer and the ensemble output for the validation (split) data

Output	F1-score
Side Layer 1	0.7708
Side Layer 2	0.7245
Side Layer 3	0.5832
Side Layer 4	0.3759
Fused Output	0.7686
Ensembled	**0.7877**

The official metric for evaluation was F1-score. We have used the same to evaluate the results. Since the network is trained for 5 outputs, we have evaluated the output of each layer to get the best results. Table 1 shows the F1-score of all five layers. From Table 1, it is very clear that the output of first side layer is most important in the fused output. Hence, we tried to ensemble the results of the first side output layer and the fused layer. The weighted average ensemble method is used to ensemble the results.

The resulted images of this ensemble are used for final submission. The results on all the datasets are presented in Table 2.

Table 2: Results of the Proposed network: Skeleton. These results are the results of the final ensembled layers

	No. of Images	F1-score (ours)	F1-score (baseline [6])
Train	1296	0.8406	-
Validation (Split)	244	0.7877	-
Validation (Original)	242	0.7480	0.6244
Test	266	0.7711	-

Figure 5 shows the resulted images of all the outputs layers and ensembled image as well.

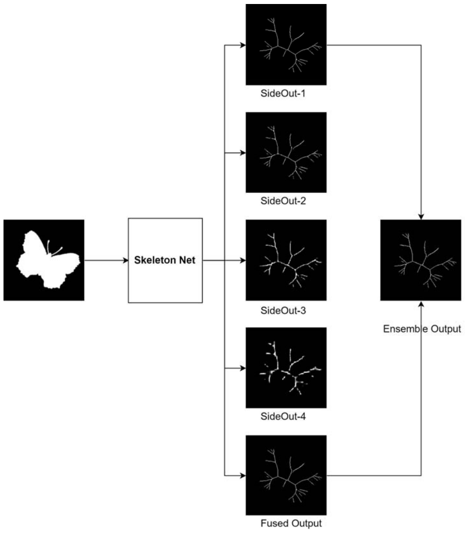

Figure 5: Side – Layers, Fused layer output and Ensembled output

6. Discussion:

The proposed architecture is a combination of many proven state-of-art algorithms. As discussed in section 4.1, we have used the coordinate convolutional layer to choose upon the translation features, this has helped our model to focus on more important features during the training. When compared to the plain encoder, the use of coordinate convolutional layer helped to improve the F1-score by more than 3%. However, this impact may be considered as insignificant in alone but when combined with our custom loss, it has shown the significant improvement in the learning process of the model as the F1-score have increased to 0.7686 from 0.6546 (in case of Binary Cross Entropy) on the validation (split) data.

Table 3: Impact of Using Coordination Convolutional Layer on the F1-score

	No Coordinate Conv Layer		With Coordinate Conv Layer	
Loss Function	Binary Cross Entropy	Our Loss Function	Binary Cross Entropy	Our Loss Function
F1-score	0.6546	0.7212	0.7043	**0.7686**

Further, we have introduced the HED architecture i.e. the side layers in the decoder part along with the dilation layer after fusing. This has boosted up the model performance by more than 10%. Table 4 shows the F1 scores with and without side-layers in the decoder part (Table 4).

Table 4: F1 score on the validation (split) data with and without side-layer

	Vanilla Decoder	Decoder with Side-Layers
F1-score	0.6973	0.7686

Some images from the training data along with the predicted output and ground truth are presented in Figure 6.

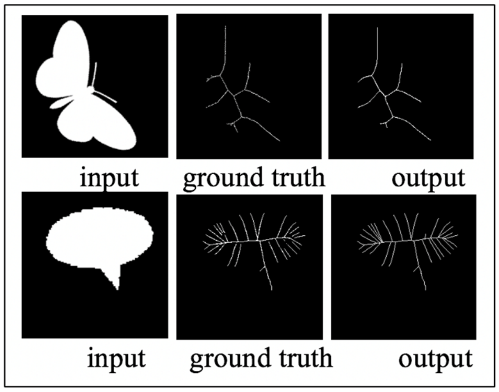

Figure 6: Illustrations of the predicted results. Results are directly compared with the ground truth images

7. Conclusion and Future Work:

In the present task, we have experimented a unique version of HED architecture along with the U-net structure to extract the skeleton from the pre-segmented images. Also, we have proposed a new loss function for converging the network for the best results. The present work also proves the role of side- layers in achieving the best output. As future work, we would like to explore the role of side layers in segmenting and extracting the skeleton from RGB images.

Acknowledgement
This work is supported by Couger Inc., Shibuya, Japan

References

[1] Attali D, Montanvert A. Computing and simplifying 2D and 3D continuous skeletons. Computer vision and image understanding, 67(3):261-73, 1997.

[2] Bai X, Latecki LJ, Liu WY. Skeleton pruning by contour partitioning with discrete curve evolution. IEEE transactions on pattern analysis and machine intelligence, 29(3):449-62, 1997.

[3] Bertasius G, Shi J, Torresani L. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4380-4389, 2015.

[4] Bucksch A. A practical introduction to skeletons for the plant sciences. Applications in plant sciences, 2(8):1400005, 2014.

[5] Chazal F, Lieutier A. The "λ-medial axis". Graphical Models, 67(4):304-31, 2005.

[6] Demir, I., Hahn, C., Leonard, K., Morin, G., Rahbani, D., Panotopoulou, A., . . . Kortylewski, A. (2019, March 21). SkelNetOn 2019 Dataset and Challenge on Deep Learning for Geometric Shape Understanding. Retrieved from https://arxiv.org/abs/1903.09233

[7] Durix B, Chambon S, Leonard K, Mari JL, Morin G. The Propagated Skeleton: A Robust Detail-Preserving Approach. In International Conference on Discrete Geometry for Computer Imagery, pp. 343-354. Springer, Cham, 2005.

[8] Dollár P, Zitnick CL. Fast edge detection using structured forests. IEEE transactions on pattern analysis and machine intelligence, 37(8):1558-70, 2015.

[9] Giesen J, Miklos B, Pauly M, Wormser C. The scale axis transform. In Proceedings of the twenty-fifth annual symposium on Computational geometry, pp. 106-115, ACM, 2009.

[10] Hariharan B, Arbeláez P, Girshick R, Malik J. Hypercolumns for object segmentation and fine-grained localization. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 447-456, 2015.

[11] Hou Q, Liu J, Cheng MM, Borji A, Torr PH. Three birds one stone: a unified framework for salient object segmentation, edge detection and skeleton extraction. arXiv preprint arXiv:1803.09860. 2018.

[12] Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132-7141, 2018.

[13] Jalal A, Kamal S, Kim D. Depth map-based human activity tracking and recognition using body joints features and self-organized map. In Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), pp. 1-6, IEEE, 2014.

[14] Kundu M, Sengupta D, Dastidar JG. Tracking Direction of Human Movement-An Efficient Implementation using Skeleton. arXiv preprint arXiv:1506.08815. 2015.

[15] Latecki LJ, Li QN, Bai X, Liu WY. Skeletonization using SSM of the distance transform. In2007 IEEE International Conference on Image Processing, 5(V):349, IEEE, 2007.

[16] Leonard K, Morin G, Hahmann S, Carlier A. A 2D shape structure for decomposition and part similarity. In2016 23rd International Conference on Pattern Recognition (ICPR), pp. 3216-3221, IEEE, 2016

[17] Li N. An implementation of ocr system based on skeleton matching, 1993.

[18] Liu R, Lehman J, Molino P, Such FP, Frank E, Sergeev A, Yosinski J. An intriguing failing of convolutional neural networks and the coordconv solution. In Advances in Neural Information Processing Systems, pp. 9605-9616, 2018.

[19] Bronstein AM, Bronstein MM, Bruckstein AM, Kimmel R. Analysis of two-dimensional non-rigid shapes. International Journal of Computer Vision, 78(1):67-88, 2008.

[20] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pp. 234-241, Springer, Cham, 2015.

[21] Roy AG, Navab N, Wachinger C. Concurrent spatial and channel 'squeeze & excitation' in fully convolutional networks. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 421-429, Springer, Cham, 2018.

[22] Shen W, Zhao K, Jiang Y, Wang Y, Zhang Z, Bai X. 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, pp. 222-230, 2016.

[23] Shen W, Wang X, Wang Y, Bai X, Zhang Z. 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, pp. 3982-3991, 2015.

[24] Sebastian TB, Klein PN, Kimia BB. Recognition of shapes by editing their shock graphs. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1(5):550-71, 2004.

[25] SkelNetOn @ CVPR19. (2019, February). Retrieved from http://ubee.enseeiht.fr/skelneton/challenge.html.

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

[27] Yang J, Price B, Cohen S, Lee H, Yang MH. Object contour detection with a fully convolutional encoder-decoder network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 193-202, 2016.

[28] Zhao F, Tang X. Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction. Pattern Recognition, 40(4):1270-81, 2007.