U-Net based skeletonization and bag of tricks

Nam Hoang Nguyen
Pukyong National University
namnguyen@pukyong.ac.kr

Abstract

Skeletonization is a process focused on providing a compact and simple representation of an object by extracting the skeleton pixels from the given shape in a binary image. This method has been widely applied in various image processing and computer vision applications. In addition to traditional approaches which are not robust and provide low accuracy results, many efforts have been made for creating deep learning based methods to overcome these disadvantages. However, skeletonization is still a new topic in the deep learning world. In this paper, we propose our solution for the Pixel SkelNetOn challenge in the third edition of the "Deep Learning for Geometric Computing" workshop at ICCV 2021, which includes (1) modification of U-Net architecture using the attention mechanism, (2) implementation of auxiliary task learning for a more effective training process and (3) application of several tricks for improving the skeletonization model's performance. Our method achieved 0.8000 on the Pixel SkelNetOn validation set and second place in the leaderboard. We also release our code to facilitate future research at `https://github.com/namdvt/skeletonization`.

1. Introduction

Skeletonization or medial axis transform is a morphological processing function that decreases the foreground regions in an image to obtain the simple skeleton lines, which spread along the medial axis of the object. This representation is widely used in digital image processing and computer vision, especially in pattern recognition and image analysis, such as image retrieval, matching, compression, vectorization and optical character recognition (OCR).

Traditionally, the skeleton of an image can be extracted using the morphological thinning method that erodes the pixels from the object's boundary repeatedly. In addition, other methods compute the skeleton by obtaining the distance to the closest boundary from each point using the distance transform. These classical methods usually provide low accuracy prediction and sensitive to noise. Recently, with the emergence of convolutional neural networks (CNN), some deep learning based skeletonization approaches have been proposed [6, 8, 9], which extract the skeleton of object by solving a segmentation or pixel-wise binary classification problem. Even though CNN approaches provide impressive success in classification and segmentation tasks compared to traditional methods, creating a deep learning model for extracting robust and accurate skeletons from object shapes still remains a challenge.

In this paper, we propose our solution for the Pixel SkelNetOn challenge [1] in the Deep Learning for Geometric Computing - ICCV 2021 Workshop and Challenge, which includes:

(1) The modification on the encoder and decoder components of the original U-Net architecture [10] using the attention mechanism to improve the feature representation and reconstruction capabilities, respectively.

(2) Auxiliary task learning in different image resolutions for boosting the training efficiency and the accuracy of output results.

(3) Bags of tricks for increasing the performance of the deep learning based skeletonization model.

2. Related Works

Due to the various applications of skeletonization, many computational approaches have been proposed for object's skeleton extraction. For example, Zhang *et al.* [14] propose a fast parallel thinning algorithm, which iteratively removing the pixels on object borders until the object shape is thinned down to obtain a unitary thickness skeleton. Lu *et al.* [7] improve this method by considering and preserving the important structure which should not be eliminated during the removing process. Lee *et al.* [4] present the 3-D parallel thinning method using an octree data structure to examine the 3x3x3 local neighborhood of every pixel. However, the computational based skeletonization methods are usually not robust to noise and generate low accuracy results.

Recently, to overcome the limitation of traditional methods, many deep learning based approaches for skeletonization have been proposed. For instance, the DeepSkeleton

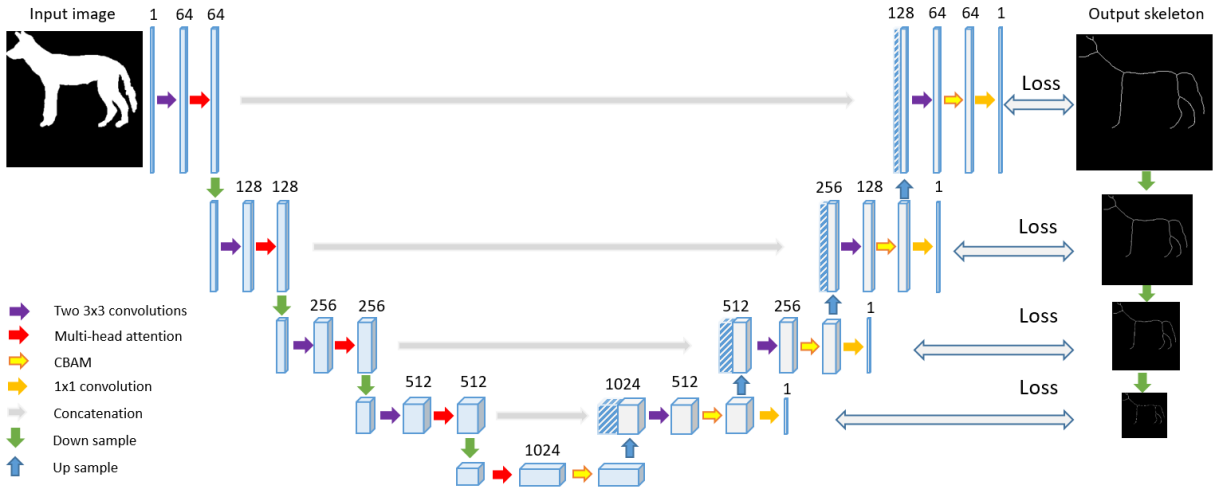


Figure 1: The modified U-Net architecture.

[11] is a fully convolutional network, which is designed to extract the skeleton in different scales from multi stages, then these multi-scale skeletons are combined to obtain the final result. The Multi-Scale Bidirectional Fully Convolutional Network (MSB-FCN) [13] employs a bidirectional structure to capture multi-scale feature representations of deep features of the network to learn the information from multiple sub-regions. Liu *et al.* propose the Rich Side-output Residual Network (RSRN) [5], which fuses rich side-outputs of VGG to utilize the information from each feature layer. Furthermore, the results are refined hierarchically by reducing the residual between side outputs and the ground truth.

Even though convolutional neural networks achieve impressive results in various computer vision problem, it remains problematic to create a high performance skeletonization model. In this paper, we present our solution for the Pixel SkelNetOn 2021 challenge, which is able to provide accurate skeletons from object shapes and a high score on the leaderboard.

3. Proposed Method

In this section, we introduce our modification of U-Net architecture using the attention mechanism in section 3.1. We then provide our loss function and data augmentation methods used to train the model in section 3.2 and section 3.3, respectively. Finally, we present some post-processing tricks for improving the model's performance in section 3.4.

3.1. Network Architecture

Since the skeleton extraction process can be considered as solving a segmentation problem, we start from the U-Net architecture [10] as the baseline, which achieves high performance on various segmentation tasks. Then, we apply

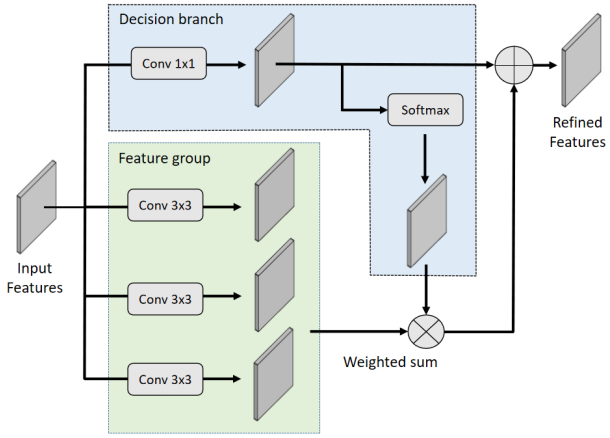


Figure 2: The multi-head attention.

improvements on the U-Net's encoder and decoder components using the attention mechanism to enhance the effectiveness of the original architecture, as illustrated in Figure 1.

Encoder. The encoder component of the original U-Net consists of 3x3 convolutions, followed by ReLU activation functions and max pooling layers for the downsampling. This component is responsible to capture the context of the input image by encoding it into the multi-level feature representations. However, as our experiment, the original U-Net encoder is not suitable for providing high quality feature representations for different object shapes in the skeletonization task, resulting in a moderate predicted skeleton and a low prediction score.

To this end, motivated by [3], we improve the feature representation of the original U-Net encoder by adopting

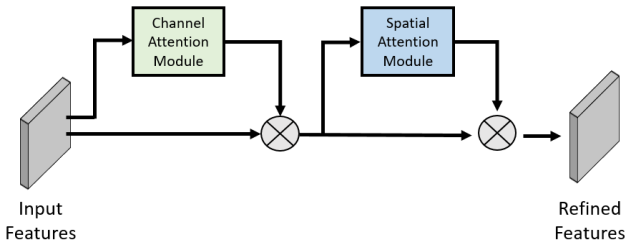


Figure 3: The CBAM architecture

the multi-head attention to learn diverse features from object shapes. Specifically, the input feature map is fed into multiple 3x3 convolution layers, which is expected to learn distinct encoded features of the object. Then, the decision branch which consists of a 1x1 convolution layer and a softmax layer is responsible to decide the contribution of each head's feature to the final feature representation, as illustrated in Figure 2.

Decoder. In the original U-Net architecture, the decoder consists of upsampling and concatenation layers followed by multiple 3x3 convolution layers, which is responsible for reconstructing the skeleton image using the features learnt by the encoder. For improving the decoder capabilities, the attention mechanism is also adopted for refining the feature map. Specifically, the Convolutional Block Attention Module (CBAM) [12] was added before 3x3 convolution layers in the decoder component. This modification increases the representation power by focus on important and suppressing unnecessary information from encoded features, as shown in Figure 3.

3.2. Loss Function

Weighted focal loss. To address the problem of unbalanced class in the Pixel SkelNetOn dataset, we adopt the weighted focal loss function as

$$L_{\text{focal}} = \alpha p^{\gamma} \log(p) + (1-\alpha)(1-p)^{\gamma} \log(1-p), \quad (1)$$

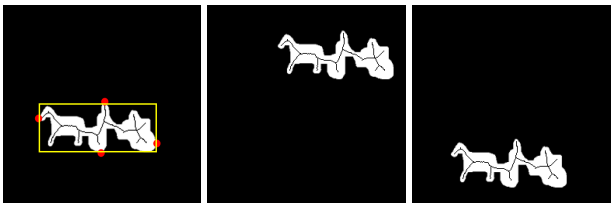
where α is the weight for positive class, p is the probability that the sample belongs to positive class and γ is the focusing parameter.

Dice loss. We also use the dice loss to minimize the overlap between predicted and target skeleton image, as

$$L_{\text{dice}} = 1 - 2\frac{\sum_i y_i p_i + \epsilon}{\sum_i y_i + \sum_i p_i + \epsilon}, \quad (2)$$

where y_i is the target label and ϵ is a small constant to avoid division by zero. Then, the final loss function is defined as the combination of focal loss and dice loss, as

$$L = L_{\text{focal}} + L_{\text{dice}}. \quad (3)$$



(a) (b) (c)

Figure 4: The illustration of the shift augmentation. (a) Original sample with extreme points (red) and bounding box (yellow) and (b,c) two examples of the augmentation results.

Auxiliary task learning. Furthermore, by learning multiple outputs from a single target skeleton, as shown in Figure 1, the auxiliary task learning appears to improve training efficiency, reduce over-fitting problem and boost the performance of the primary task. Specifically, feature maps with different resolutions from each stage of the decoder components are fed into 1x1 convolutions layers to get the low resolution predicted skeletons. Then, the auxiliary tasks are formulated to minimize the loss between these outputs and the corresponding down-sampled target skeletons. This modification provides better convergence and able to improve the performance of the desired main task. Then, the final loss function is defined as

$$L_{\text{final}} = 0.5L_{256} + 0.3L_{128} + 0.2L_{64} + 0.1L_{32}, \quad (4)$$

where L_{256}, L_{128}, L_{64} and L_{32} are the loss of primary and auxiliary tasks of multiple image resolution of 256x256, 128x128, 64x64 and 32x32, respectively.

3.3. Data Augmentation

Following the previous works in the Pixel SkelNetOn challenge [9, 2], we augment the data using rotation (90°, 180°, 270°) and flipping (horizontal, vertical) operations. However, the number of samples is still limited, results in over-fitting problem. To tackle this issue, we propose a shifting augmentation to effectively increase the amount of training data by moving the object into different locations in the image. Specifically, we first find the four extreme points (north, south, east, west) of the object contour in the input binary image. Then the object is cropped using the bounding box obtained from the estimated extreme points. Finally, the object is randomly allocated in the image to create new training data, as presented in Figure 4.

3.4. Post-Processing

We apply some post-processing techniques to further enhance the performance of our skeletonization model.

Method	Split-Test	SkelNetOn
Baseline	0.6200	0.6800
Baseline + Multi-head Attention	0.7731	0.7552
Baseline + Multi-head Attention + CBAM	0.7824	0.7725
Baseline + Multi-head Attention + CBAM + Auxiliary tasks	0.8032	0.7891

Table 1: Performance of different methods on Split-Test and official Pixel SkelNetOn validation set.

Method	Performance		
Threshold searching	✓	✓	✓
Test time augmentation		✓	✓
Ensemble			✓
Prediction score	0.7926	0.7971	0.8000

Table 2: Performance of different post-processing techniques on official Pixel SkelNetOn validation set.

Threshold searching. After getting output from the last sigmoid layer in the model, the predicted skeleton is obtained by thresholding every pixel with the same threshold value, normally set to 0.5. If the pixel intensity is greater than the threshold value, it is set to 255, otherwise it is set to 0. However, using a fixed threshold value 0.5 is not efficient, since the dataset is highly unbalance. To address this issue, we attempt to search for an optimal value of threshold on the hold-out validation set, then the best threshold is used to create output skeletons from the official SkelNetOn validation set.

Test time augmentation. We apply test time augmentation to make the model more robust and improve the prediction results. Specifically, we create multiple of augmented sample from the input image using flip and rotate operations. After obtaining the prediction for each, we simply average these predictions to make the final skeleton.

Ensemble. Finally, to obtain higher model generalization performance and reduce the over-fitting problem, we ensemble five fold models by taking the average prediction from each model.

4. Experimental Results

The model was trained on the Pixel SkelNetOn dataset provided by the SkelNetOn 2021 Challenge [1], which contains 1,725 binary images with size 256x256 pixels. We split the dataset to the Split-Train and Split-Test in the ratio of 80:20, which considering the object type information. Specifically, Split-Train contains 1,380 images and Split-Test contains 345 images. We train the model using SGD optimizer with the cosine annealing scheduler with learning rate is 0.02 and validate the model performance using F1-score. For the weighted focal loss function, we set $\alpha = 0.01$ and $\gamma = 2$ for all experiments.

4.1. Ablation Study

In this section, we summarize the results of our method on two validation sets, the Split-Test and the official SkelNetOn validation set, refer as SkelNetOn in Table 1 and Table 2.

Baseline. We used the U-Net model as the baseline and trained using the focal loss and dice loss defined in equation 3. This method achieved 0.62 and 0.68 F1-score on the Split-Test and SkelNetOn validation set, respectively, which are moderate scores since the original U-Net architecture is not suitable for the skeletonization task.

U-Net with attention mechanism. The encoder and decoder components of the original U-Net are modified by adding the multi-head attention and CBAM architecture as described in section 3.1. These improvements significantly increase the model performance to 0.7731 and 0.7824 on Split-Test, and 0.7552 and 0.7725 on SkelNetOn validation set. The results showed that the attention mechanism is capable to enhance the feature representation power of the original U-Net.

Auxiliary tasks learning. By adding the auxiliary tasks in different stages of the decoder network, the performance of the main task is improved by 0.0208 and 0.0166 to reach 0.8032 and 0.7891 F1-score on Split-Test and SkelNetOn validation set, respectively.

Post-processing. Table 2 illustrates the results of post-processing methods on the SkelNetOn validation set. Firstly, by using the threshold searching instead of using threshold value 0.5, the prediction score increases from 0.7891 to 0.7926. Secondly, the TTA method helps our model gain more 0.0045 scores. Finally, by ensembling five-fold models, we achieve 0.8000 prediction score on the Pixel SkelNetOn validation set.

Rank	Team name	Prediction Score
1	BOE_AIoT_CTO	0.8129
2	**namdvt(Ours)**	**0.8000**
3	priyakansal	0.7961
4	sabarinathan	0.7950
5	natsubk95	0.7948

Table 3: Leaderboard of Pixel SkelNetOn challenge.

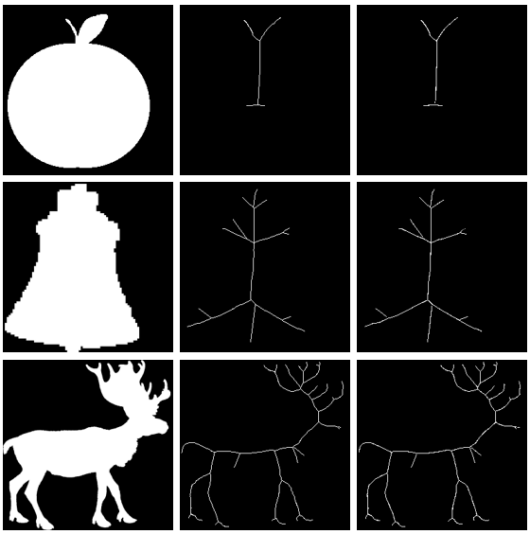


Figure 5: Visualize results obtained by proposed method. From left to right: Input, ground truth and predicted skeleton.

4.2. Competition Results

In the Pixel SkelNetOn challenge, our team achieves 0.8000 in the F1 score, which yields second place on the final leaderboard, without using any external data. Performance of top 5 teams are shown in Table 3. We also visualize some of the results for predicted skeleton. As shown in Figure 5, our proposed method is able to obtain high quality skeletons which close to the ground truth.

5. Conclusions

In this paper, we proposed our solution for the skeletonization problem, by making improvements on the original U-Net architecture using the attention mechanism and exploiting the auxiliary tasks. Furthermore, we also present some tricks for improving the model performance to create more robust and accurate results. Our method is simple and efficient for extracting the skeleton from binary images, and achieves a prediction score of 0.8000 on the Pixel SkelNetOn validation set, which is in the second place on the leaderboard.

References

[1] 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. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[2] Nan Jiang, Yifei Zhang, Dezhao Luo, Chang Liu, Yu Zhou, and Zhenjun Han. Feature hourglass network for skeleton detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[3] Yonghyun Kim, Wonpyo Park, Myung-Cheol Roh, and Jongju Shin. Groupface: Learning latent groups and constructing group-based representations for face recognition. *CoRR*, abs/2005.10497, 2020.

[4] T.C. Lee, R.L. Kashyap, and C.N. Chu. Building skeleton models via 3-d medial surface axis thinning algorithms. *CVGIP: Graphical Models and Image Processing*, 56(6):462–478, 1994.

[5] Chang Liu, Wei Ke, Jianbin Jiao, and Qixiang Ye. Rsrn: Rich side-output residual network for medial axis detection. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 1739–1743, 2017.

[6] Chang Liu, Wei Ke, Fei Qin, and Qixiang Ye. Linear span network for object skeleton detection. *CoRR*, abs/1807.09601, 2018.

[7] H. E. Lü and P. S. P. Wang. A comment on "a fast parallel algorithm for thinning digital patterns". *Commun. ACM*, 29(3):239–242, Mar. 1986.

[8] Sabari Nathan and Priya Kansal. Skeletonnet: Shape pixel to skeleton pixel. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[9] Oleg Panichev and Alona Voloshyna. U-net based convolutional neural network for skeleton extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[11] Wei Shen, Kai Zhao, Yuan Jiang, Yan Wang, Xiang Bai, and Alan Yuille. Deepskeleton: Learning multi-task scale-associated deep side outputs for object skeleton extraction in natural images. *IEEE Transactions on Image Processing*, 26(11):5298–5311, Nov 2017.

[12] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: convolutional block attention module. *CoRR*, abs/1807.06521, 2018.

[13] Fan Yang, Xin Li, Hong Cheng, Yuxiao Guo, Leiting Chen, and Jianping Li. Multi-scale bidirectional fcn for object skeleton extraction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[14] T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Commun. ACM*, 27(3):236–239, Mar. 1984.