SPFUSIONNET: SKETCH SEGMENTATION USING MULTI-MODAL DATA FUSION

Anonymous ICME submission

ABSTRACT

The sketch segmentation problem remains largely unsolved because conventional methods are greatly challenged by the highly abstract appearances of freehand sketches and their numerous shape variations. In this work, we tackle such challenges by exploiting different modes of sketch data in a unified framework. Specifically, we propose a deep neural network SPFusionNet to capture the characteristic of sketch by fusing from its image and point set modes. The image-modal component SketchNet learns hierarchically abstract robust features and utilizes multi-level representations to produce pixel-wise feature maps, while the point set-modal component SPointNet captures local and global contexts of the sampled point set to produce point-wise feature maps. Then our framework aggregates these feature maps by a fusion network component to generate the sketch segmentation result. The extensive experimental evaluation and comparison with peer methods on our large SketchSeg dataset verify the effectiveness of the proposed framework.

Index Terms— sketch segmentation, multi-modal fusion, deep neural network

1. INTRODUCTION

Sketches are very intuitive to humans and have long been used as an convenient communicative tool. With the popularity of touchscreen devices, sketching has become convenient and ubiquitous. Research on sketches has consequently flourished in recent years, with a wide range of applications being investigated, including sketch recognition [1], sketch segmentation and labeling [2], sketch-based object retrieval [3], scene modeling [4] and free-hand sketch synthesis [5].

However, prior works on sketch segmentation typically analyze an input sketch with conventional hand-crafted features from separate modes (e.g., image based HOG [6] and geometric based shape context [7]). They are often coupled with complicated models like Mixed Integer Programming [8] and Conditional Random Field [2] to yield the final semantically segmented components with labels. However, the segmentation accuracy is far from satisfying although plenty of effort has been dedicated. Inspired by the recent breakthroughs of deep learning approaches [9, 10, 11], we investigate an elegant unified framework that is capable of joint optimization to incorporate multiple complemental modes of sketch data for segmentation.

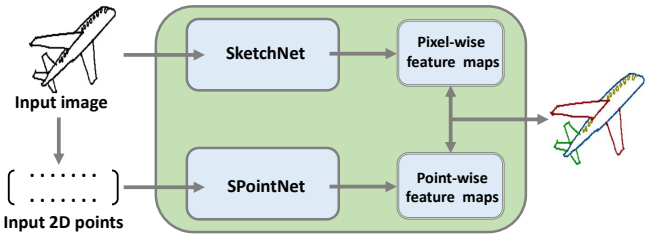

Fig. 1. Illustration of our unified framework tackling sketch segmentation by fusing image and point set modes.

Therefore, in this paper we propose a novel deep neural network, Sketch-Point Fusion Network (SPFusionNet), as illustrated in Fig. 1, for free-hand sketch segmentation. Our SPFusionNet well exploits the characteristics of sketch with both sketch image and point set data by aggregating two network components: the SketchNet takes a sketch image as input and produces pixel-wise feature maps, and the SPointNet takes a 2D point set sampled from the sketch and produces point-wise feature maps. Specifically, the SketchNet is designed in an encoder-decoder scheme, where concatenated Spatial Invariance Enhanced Residual (SIER) blocks are designed to facilitate effective feature learning and handle sketch deformation while multi-level abstract feature representations are combined and decoded into pixel-wise feature maps. On the other hand, the SPointNet borrows the network structure from the PoinNet for 3D point cloud, and it encodes and combines global and local features from the 2D point set to produce point-wise feature maps. In the aggregation stage, these feature maps are fed into a fusion network component in our framework to yield the final segmentation results.

The contributions of this paper are threefold. First, we propose a unified deep neural network framework to fuse multiple complementary modes of sketch data for the sketch segmentation task, and our method can easily be generalized to other related tasks such as the classification and retrieval of sketches or 3D models. Second, we introduce a spatial invariance enhanced residual block for learning more robust image features. Third, we have constructed a large dataset SketchSeg for training and evaluating deep learning-based sketch segmentation methods, and conducted extensive experimental results on it to verify the effectiveness of our method.

The rest of the paper is organized as follows: Section 2 discusses related work, and we describe the proposed SPFu-

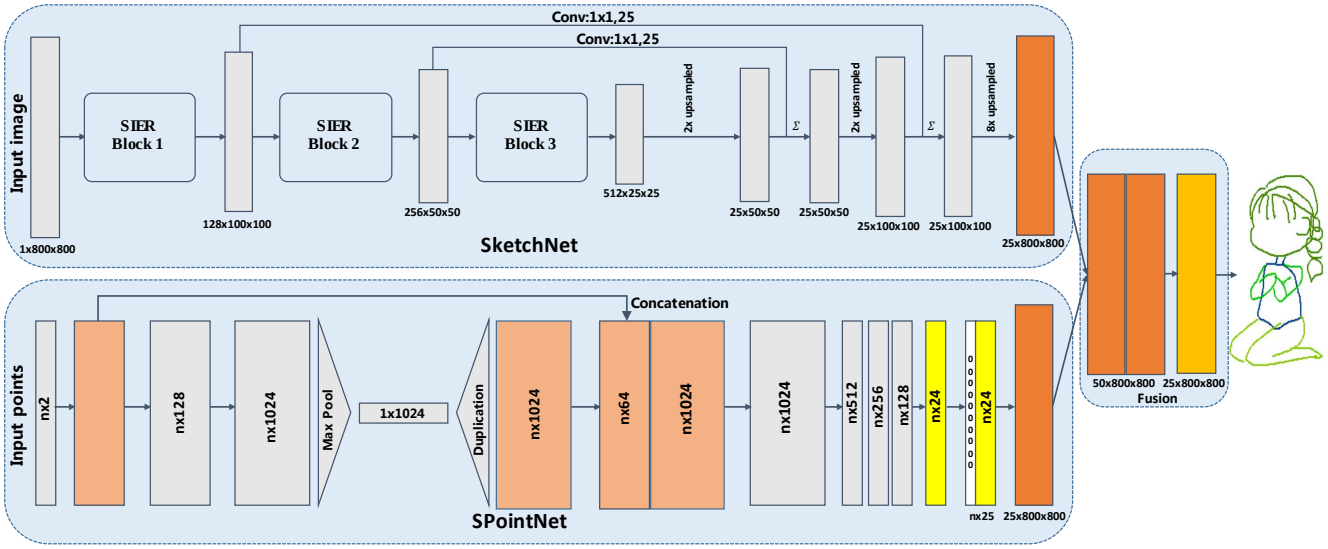

Fig. 2. Illustration of the proposed SPFusionNet framework.

sionNet in Section 3. Experimental results are presented in Section 4 and the last section concludes the paper.

2. RELATED WORK

Most existing works on sketch segmentation [2, 8, 12] build their models on hand-crafted features. In [12], Radial Basis Functions are used for modeling sketches. Schneider et al. [2] apply Conditional Random Field to find the labels of sketch segments. Huang et al. [8] use Mixed Integer Programming for this task by optimizing over both local component fitness and global structure plausibility. However, these methods are greatly challenged by the sparse visual information of sketches and their large appearance variations, rendering the sketch segmentation task remains largely unsolved.

Recently, deep neural networks (DNNs) have been witnessed to achieve great performance in image segmentation [11, 13] that is closely related to sketch segmentation. Farabet et al. [14] apply DNNs to extract dense feature vectors from raw pixels and then employ multiple post-processing methods to fulfil pixel labeling. Long et al. [11] propose a fully convolutional network (FCN) to achieve impressive performance gain on image segmentation. Chen et al. [13] introduce the idea of atrous convolutions to further improve the performance. Ronneberger et al. [15] modify FCN and present the U-Net model for biomedical image segmentation by constructing a strictly symmetric convolution-deconvolution architecture. However, although these DNN-based image segmentation methods can easily be applied to sketch segmentation, there exist no suitable large datasets for training these models, which inspires our construction of the large SketchSeg dataset.

Many DNN-based methods also have been developed for 3D shape analysis that may provide geometric structure based modeling solutions for sketch data. In [16, 17], 3D convolutional filters are defined to capture volumetric representations of 3D shapes, but these methods are constrained and costly for sparse point cloud data. Yi et al. [18] introduce a spectral parametrization of dilated convolutional kernels and a spectral transformer network for 3D shape segmentation by combining the power of DNNs and spectral analysis. Qi et al. [10] propose the PointNet model, a novel type of DNN that directly works on 3D point clouds. Although PointNet is inferior to volumetric based DNNs in accuracy, its efficiency is notable. And it inspires us to handle sketches in the point set mode. Li et al. [19] train a deep neural network to transfer existing segmentations and labelings from 3D models to freehand sketches, but each category requires separate training.

3. PROPOSED METHOD

3.1. Overview of SPFusionNet

Our framework addresses the sketch segmentation problem by well exploring multiple modes of sketch data. Specifically, we design a deep neural network, termed Sketch-Point Fusion Network (SPFusionNet), to take sketch images and point sets as input and fuse them effectively to achieve accurate sketch segmentation.

Figure 2 illustrates the detailed configuration of the proposed SPFusionNet. It incorporates two components SketchNet and SPointNet, which turn the sketch image and the point set into pixel-wise and point-wise feature maps, respectively. The two types of feature maps are converted and concatenated in the pixel-wise form and are further fed into a fusion network component to obtain the sketch segmentation result.

The detail of each component will be described thereinafter.

3.2. SketchNet

The SketchNet aims to explore the characteristic of sketch images and works in an encoder-decoder way. In the encoding stage, a sequence of Spatial Invariance Enhanced Residual (SIER) blocks are utilized to learn and extract hierarchically abstract feature representations. These multi-level representations are then combined in the decoding stage to yield pixel-wise feature maps for segmentation.

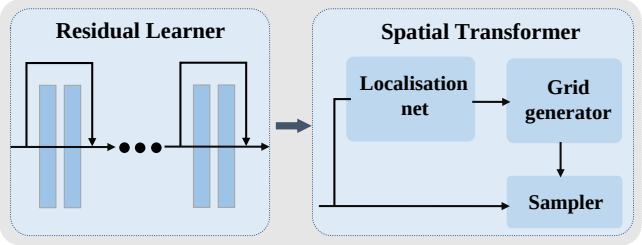

Fig. 3. Illustration of the SIER block.

Our designed SIER block for encoding can facilitate effective feature learning and handle sketch deformation, and it consists of two modules, i.e., a residual learner and a spatial transformer, as shown in Fig. 3. The residual learner takes advantage of the residual structure [9] that is proved to have powerful learning ability, and it consists of stacked convolution layers with residual shortcut connections. Different SIER blocks can use residual learners of different structures. The spatial transformer is used to learn enhanced feature representations robust to sketch deformation, and it is formulated as a spatial transform network (STN) [20] that includes three parts: localization net, grid generator and sampler.

We use a series of deconvolution layers for decoding in SketchNet, and their number is the same as that of SIER blocks. If a SIER block extracts feature representation with the size k times reduced, its corresponding deconvolution layer will upsample the feature maps by a factor of k. As exhibited in Fig. 2, skip-connections are used to aggregate multi-level feature representations. Specifically, before the feature maps are input to a deconvolution layer, the output of its SIER block counterpart will be convolved with 1×1 filters and added to each channel of the input feature maps.

Given a sketch image I with size of $W \times H$, SketchNet will output its feature maps $\hat{\mathbf{f}}^S \in \mathbb{R}^{W \times H \times (C+1)}$, where W, H and $C+1$ are the width, height and channel number, respectively. Moreover, C equals to the total number of sketch component labels in the dataset, and we add a label 0 for the blank background to result in $C+1$ labels. In this work, the SketchNet is implemented with 3 SIER blocks whose learners are specified by a 34-layer ResNet model [9] so as to take advantage of the model parameters pre-trained on the ImageNet. Specifically, the first 16, middle 12, and last 6 layers of the ResNet-34 model are used as the structures of the 3 learners separately. In this case, the 3 SIER blocks will extract feature representation with sizes reduced by 8, 2 and 2 times, respectively. Therefore, 3 deconvolution layers with upsampled factors of 2, 2 and 8 are used correspondingly.

3.3. SPointNet

The SPointNet component is employed to capture the characteristic of sketch in the point set form where the coordinates of points describe geometry information implicitly. SPointNet is inspired by the work [10] that uses PointNet for 3D point cloud processing, but it instead takes 2D point set as input for sketch segmentation. A point set $P = \{p_i \in R^2\}_{i=1}^N$ from the sketch contour is sampled by scanning the sketch image I from left to right and top to bottom. The SPointNet will capture global context of the sampled point set and combine with point-wise features to produce point-wise feature maps.

In this work, we directly use the PointNet architecture [10] for configuring our SPointNet. Specifically, the point set is first passed through three 1×1 convolution layers with 64, 128 and 1024 channels respectively, and the output feature map of size $N \times 1024$ are max pooled to obtain a 1024-dimensional vector representing global context of the point set. Then this vector is duplicated N times and concatenated with the point-wise feature map of size $N \times 64$ from the first convolution layer. Afterwards, the concatenated feature maps are processed by five 1×1 convolution layers to result in a $N \times C$ feature map. We add a N-dimensional column vector of zeros representing the background to obtain a $N \times (C+1)$ feature map $\mathbf{f}^P$ for later processing.

3.4. Multi-modal fusion

In the final stage of SPFusionNet, we fuse the output feature maps from SketchNet and SPointNet to generate the sketch segmentation result. First of all, these feature maps should be in the same form, and thus the feature map $\mathbf{f}^P$ from SPointNet will be converted to the pixel-wise form. Specifically, we initialize feature maps of size $W \times H \times (C+1)$ with all zeros and the i-th row vector of $\mathbf{f}^P$ will be assigned to the pixel location that corresponds to the i-th point's coordinates, so that we obtain the pixel-wise feature maps $\hat{\mathbf{f}}^P$ in accordance with $\mathbf{f}^P$. Then feature maps $\hat{\mathbf{f}}^S$ and $\hat{\mathbf{f}}^P$ are aggregated by a fusion function, formulated as

$$\mathbf{s} = g(\hat{\mathbf{f}}^S, \hat{\mathbf{f}}^P), \tag{1}$$

where $\mathbf{s} \in \mathbb{R}^{W \times H \times (C+1)}$ denotes the score matrix, with element $\mathbf{s}_{w,h,c}$ representing the probability of pixel (w, h) belonging to label c ($c = 0, ..., C$). In this work, the function $g(\cdot)$ is implemented by a 1×1 convolution layer and a softmax layer.

3.5. Optimization

Given a dataset with M training samples $\{I^m, Y^m\}_{m=1}^M$, where I^m is the m-th sketch image with size $W \times H$, and $Y^m = (y_{w,h})_{W \times H}$ is its corresponding segmentation ground truth with $y_{w,h} \in \{0, ..., C\}$ denoting the component label of pixel (w, h). Here C is the total number of sketch components in the dataset and label 0 is used for background. A point set $P^m = \{p_i^m\}_{i=1}^N$ is sampled for the m-th sketch image as input of SPointNet. For pixel (w, h) in sketch I^m, we further define a one-hot label distribution vector $\hat{\mathbf{y}}_{w,h}^m \in \mathbb{R}^{(C+1)}$, where the c-th element $\hat{\mathbf{y}}_{w,h,c}^m = 1$ if pixel (w, h) has label c and all the other elements are 0.

The proposed SPFusionNet can be trained end-to-end by minimizing the following balanced cross entropy loss function:

$$\mathcal{L} = -\sum_{m=1}^{M} \sum_{w,h} \lambda_{y_{w,h}} \sum_{c=1}^{C+1} \hat{\mathbf{y}}_{w,h,c}^m \log(\mathbf{s}_{w,h,c}^m), \qquad (2)$$

where $\mathbf{s}_{w,h,c}^m$ is the (w, h, c) element of the m-th sample's predicted score matrix $\mathbf{s}^m$, and λ_c $(c = 0, ..., C)$ is the weight for c-th component label. In this paper, we set $\lambda_0 = 0$ for background, and λ_c $(c = 1, ..., C)$ is reciprocal to the portion of component c in the dataset measured in pixels.

4. EXPERIMENTS

In this section, we will present extensive experimental evaluation and comparison with state-of-the-art methods to verify the effectiveness of our method, and conduct ablation study to analyze the contribution of each component in our SPFusionNet framework.

4.1. Experimental settings

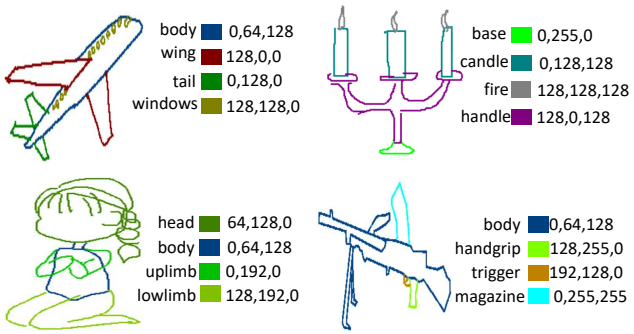

Fig. 4. Examples of sketches and their segmentation labels from the SketchSeg dataset.

Dataset. Existing datasets for sketch segmentation are unsuitable for learning and evaluating deep learning-based models. Therefore, we construct a large dataset SketchSeg, which consists of 10,000 pixel-wisely labeled sketches, for evaluation and comparison. The SketchSeg dataset contains 10 categories (1,000 sketches for each) with 24 semantic component labels. Sketch examples with their segmentation labels are exhibited in Fig. 4. We make the dataset open access on the Google Drive to help promote research in related communities (`https://drive.google.com/open?id=1WucqueaSFozOXYgPxplKfzZuB89IMwVS`).

Evaluation metrics. We follow existing works [8] to adopt two metrics for evaluation: Pixel-based accuracy (P-metric) that reflects the percentage of correctly labeled pixels, and Component-based accuracy (C-metric) that reflects the percentage of correctly labeled segments, which has 75% of its pixels correctly labeled.

Implementation details. Our SPFusionNet is implemented in PyTorch. We train our model for 50 epochs with SGD optimizer, where the size of mini-batch is 5, initial learning rate is 0.01 and momentum is 0.9. Three cutting-edge methods U-Net [21], LinkNet [22] and FCN [11] are used for comparison. We use the open source codes of U-Net and LinkNet and implement FCN. The U-Net is based on the VGG model, while Link-Net and FCN are based on the ResNet-34 model as ours. Moreover, we use the same skip-connections in our SketchNet for FCN, so actually it only differs from SketchNet by having no spatial transformer. All the compared methods are trained on a training set of 7,500 sketches from the SketchSeg dataset and tested on the rest. We perform the experiments on a PC with Intel i7 CPU, 32 GB RAM and GTX 1080ti GPU.

4.2. Experimental results and comparison

We report the experimental results of the compared methods in Tables 1 and 2. As can be observed, our SPFusionNet achieves an average segmentation performance of 92.9% in P-metric and of 90.7% in C-metric, ranking first in both evaluation metrics. Among the competitive methods, FCN performs the second best, but our method can outperform it by a large margin, with 11.2% increase in P-metric and 13.6% increase in C-metric. For each category of sketches, our SPFusionNet also shows obvious performance boost in both metrics compared with other methods. Some examples of the segmentation results are exhibited in Fig. 5. It can be seen that our SPFusionNet produces more accurate segmentation results, while the other compare methods generate many more wrongly labeled fragments. These quantitative and qualitative results strongly demonstrate the effectiveness of our proposed method. The advantage comes from the fact that our method well exploits the characteristics of sketch from multiple modes of data by a well-designed deep neural network framework. We will take further analysis in the next subsection.

Table 1. Segmentation accuracy (%) on the SketchSeg dataset in P-metric

Method	airplane	bicycle	candelabra	chair	fourleg	human	lamp	rifle	table	vase	Average
U-Net	68.9	68.1	89.3	84.0	74.1	71.9	92.2	54.8	79.6	89.9	77.3
LinkNet	78.0	65.3	88.3	89.1	76.7	74.5	91.2	59.9	82.5	93.8	79.9
FCN	78.2	71.4	90.8	86.9	80.3	75.6	92.8	65.2	81.4	94.4	81.7
SketchNet	92.1	92.7	93.4	88.2	87.4	84.4	89.9	90.1	88.1	92.2	89.9
SPointNet	76.7	75.4	73.6	83.1	71.9	65.4	83.7	78.8	69.8	79.0	75.7
Ours(SPFusionNet)	**95.1**	**95.0**	**96.0**	**92.8**	**91.4**	**86.1**	**93.9**	**92.7**	**91.3**	**94.9**	**92.9**

Table 2. Segmentation accuracy (%) on the SketchSeg dataset in C-metric

Method	airplane	bicycle	candelabra	chair	fourleg	human	lamp	rifle	table	vase	Average
U-Net	52.6	49.7	90.3	81.9	54.5	62.6	92.4	38.9	70.1	90.7	68.4
LinkNet	67.7	55.7	89.0	89.2	67.2	67.9	92.4	44.5	80.3	96.6	75.0
FCN	66.5	59.2	94.5	84.8	73.5	72.1	92.5	54.7	75.3	98.1	77.1
SketchNet	86.8	84.7	93.7	88.2	84.4	82.7	90.5	82.2	84.2	93.2	87.1
SPointNet	53.3	44.6	55.8	75.5	52.8	45.4	84.9	58.1	55.7	64.4	59.1
Ours(SPFusionNet)	**91.0**	**87.2**	**96.3**	**93.9**	**90.2**	**82.1**	**94.4**	**86.6**	**89.6**	**95.7**	**90.7**

4.3. Ablation study

We conduct experiments to evaluate the contribution of each component in our framework, and meanwhile compare with the baseline models used for implementing our method. The experimental evaluation are reported in the lower parts of Tables 1 and 2. As can be observed, when removing the SPointNet component from our framework, the SketchNet alone can achieve an average segmentation performance of 89.9% in P-metric and 87.1% in C-metric, with 3.0% and 3.6% decrease from those of SPFusionNet, respectively. On the contrast, when using the SPointNet alone, more obvious performance drop is witnessed, with 17.2% and 31.6% decrease in P-metric and C-metric, respectively. These results show that SketchNet plays a more critical role in our framework, but incorporating SPointNet to explore the point set mode of sketch data can help boost the performance favorably. Because SPointNet uses the architecture of PoinNet [10], it also reveals that our SPFusionNet performs largely better on sketch segmentation when compared with PoinNet. We further evaluate the effectiveness of spatial transformer used in the SIER blocks of SketchNet. As mentioned in Section 4.1, FCN is in fact the SketchNet with spatial transformers removed, and the comparison results show that it has a performance drop of 8.2% and 10.0% in P-metric and C-metric from the SketchNet, which indicates that incorporating the spatial transformer does help learn more robust feature representation to achieve evident performance gain.

5. CONCLUSION

In this work, we have proposed a unified deep learning based framework that exploits multi-modal data fusion with end-to-end joint optimization for sketch segmentation, and demonstrated the effectiveness of using image and point set modes of sketch data in conjunction by conducting extensive evaluation and comparison with peer methods and analyzing the contribution of each single modal network component on our constructed large SketchSeg dataset. In future, we would like to explore the use of other modes like spectrum and additional information (e.g., strokes order), extend our SketchSeg dataset and generalize our framework to other related tasks.

6. REFERENCES

[1] Qian Yu, Yongxin Yang, Feng Liu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales, "Sketch-a-net: A deep neural network that beats humans," *International journal of computer vision*, vol. 122, no. 3, pp. 411–425, 2017.

[2] Rosália G Schneider and Tinne Tuytelaars, "Example-based sketch segmentation and labeling using crfs," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 5, pp. 151, 2016.

[3] Fei Wang, Shujin Lin, Xiaonan Luo, Hefeng Wu, Ruomei Wang, and Fan Zhou, "A data-driven approach for sketch-based 3d shape retrieval via similar drawing-style recommendation," in *Computer Graphics Forum*. Wiley Online Library, 2017, vol. 36, pp. 157–166.

[4] Kun Xu, Kang Chen, Hongbo Fu, Wei-Lun Sun, and Shi-Min Hu, "Sketch2scene: sketch-based co-retrieval and co-placement of 3d models," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, pp. 123, 2013.

[5] Yi Li, Yi Zhe Song, Timothy M. Hospedales, and Shaogang Gong, "Free-hand sketch synthesis with de-

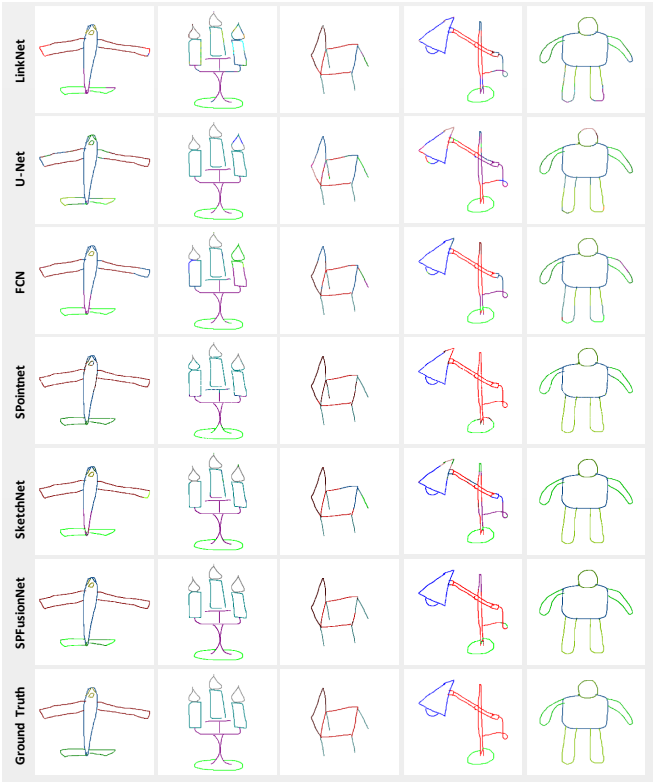

Fig. 5. Visualization of example sketch segmentation results of different methods.

formable stroke models," *International Journal of Computer Vision*, vol. 122, no. 1, pp. 169–190, 2017.

[6] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR*, 2005, pp. 886–893.

[7] Greg Mori, Serge J. Belongie, and Jitendra Malik, "Efficient shape matching using shape contexts," *TPAMI*, vol. 27, no. 11, pp. 1832–1837, 2005.

[8] Zhe Huang, Hongbo Fu, and Rynson WH Lau, "Data-driven segmentation and labeling of freehand sketches," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, pp. 175, 2014.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[10] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *CVPR*, 2017.

[11] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[12] Jiantao Pu and David Gur, "Automated freehand sketch segmentation using radial basis functions," *Computer-Aided Design*, vol. 41, no. 12, pp. 857–864, 2009.

[13] 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," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[14] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun, "Learning hierarchical features for scene labeling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[16] Daniel Maturana and Sebastian Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *IROS*. IEEE, 2015, pp. 922–928.

[17] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas, "Volumetric and multi-view cnns for object classification on 3d data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5648–5656.

[18] Li Yi, Hao Su, Xingwen Guo, and Leonidas Guibas, "Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation," in *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] Lei Li, Hongbo Fu, and Chiew Lan Tai, "Fast sketch segmentation and labeling with deep learning," *arXiv preprint arXiv:1807.11847*, 2018.

[20] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al., "Spatial transformer networks," in *NIPS*, 2015, pp. 2017–2025.

[21] Vladimir Iglovikov and Alexey Shvets, "Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation," *arXiv preprint arXiv:1801.05746*, 2018.

[22] Abhishek Chaurasia and Eugenio Culurciello, "Linknet: Exploiting encoder representations for efficient semantic segmentation," in *Visual Communications and Image Processing (VCIP), 2017 IEEE*. IEEE, 2017, pp. 1–4.