3D Part-based Segmentation via Deep Neural Network

xufei guo 36920221153080,
weijian ma 23020221154103,fuhang li 23020221154093,
qiuyue qin 23020221154106,jiawei yao 36920221153138

Abstract

Numerous frameworks on the job of 3D segmentation have been published recently, although the majority of these methods rely on the separator characteristic of geometric models. For example, the point cloud approach will result in the loss of geometric shape connections. As a result, in order to preserve the structure, we picked a mesh model to extract the shape feature. Because various grid models need to extract distinct shape representations and classic component segmentation is difficult to handle with diverse 3D models. In this study, we suggested a Deep Neural Network-based 3D Part-based Segmentation frame. In this method, We first select and augment the data obtained from the ABC data set to get the input feature. Then we utilize ResUNet to improve efficiency and generalization. The input feature is from the grid model, and the output is the segmentation result of the 3D industrial model. We can successfully solve the partial segmentation problem of the complicated and irregular three-dimensional model in this manner. We demonstrate the effectiveness of our framework by applying it to publicly available CAD mesh model data. Furthermore, the output of the CAD segment mesh will assist in editing the input more quickly and conveniently.

Introduction

Recent years, with the rapid development of 3D-imaging technology, low-cost miniaturized 3D sensors such as Microsoft Kinect, Intel's RealSense and Google's Tango can capture 3D information of the scene very well, helping intelligent devices better perceive and understand the world, and at the same time, to a large extent, promoting the development of people's exploration of acquiring real world information in a three-dimensional way [2]. The segmentation of 3D scene is a basic and challenging problem in Computer Vision and Computer Graphics. Its goal is to establish a computer technology that can predict the fine-grained labels of objects in 3D scene, which is widely used in the fields of automatic driving, mobile robots, industrial control, augmented reality and medical image analysis. 3D segmentation can be divided into three types: semantic segmentation, instance segmentation and partial segmentation. Among them, semantic segmentation aims to predict the tags of object classes, such as tables and chairs. Instance segmentation distinguishes different in-stances of the same type of label, such as table 1 or 2 and chair 1 or 2. Partial segmentation aims to further decompose the instance into different components, such as arm-rests, legs and backrests of a chair. Compared with 2D segmentation, 3D segmentation provides a more comprehensive understanding of the scene, because 3D data (such as RGB-D, point clouds, projected images, voxels, and meshes) contain more geometric, shape, and scale information, while the background noise is less [8]. With the iterative updating of GPU computing power and the emergence of large 3D model data, the idea of deep learning gradually occupies an absolute dominant position in 3D model classification, retrieval and other tasks. Recently, deep learning technology has dominated many research fields, including Computer Vision, speech recognition and Natural Language Processing. Due to the success in learning effective features, the deep learning method for 3D segmentation has also attracted more and more interest in the research community in the past decade. However, 3D deep learning methods still face many unresolved challenges. For example, the features of RGB and depth channels are difficult to fuse, the irregularity of point clouds makes it difficult to use local features, and the conversion of high-resolution voxels requires huge computing resources, which makes the technology of efficient, accurate and direct processing of 3D data a widespread demand.

In this paper, we proposed the 3D Part-based Segmentation frame based on Deep Neural Network named ResUNet. The faces are taken into account as the basic unit in mesh data processing, and connections between faces that share similar edges are formed. This approach allows us to address the complexity and irregularity issues using perface processes and a symmetry function. In addition, the features of faces are divided into structural and spatial features. Based on these assumptions, we construct the network architecture for learning the primary features and combining surrounding features. In this way, we can solve the part segment of the complex and irregular three dimension model well.

Related work

Mesh segmentation is a fundamental research topic in geometry processing and computer graphics. Mesh segmentation aims to decompose a mesh, representing a 3D object, into parts.In general, there are two major categories of mesh seg-

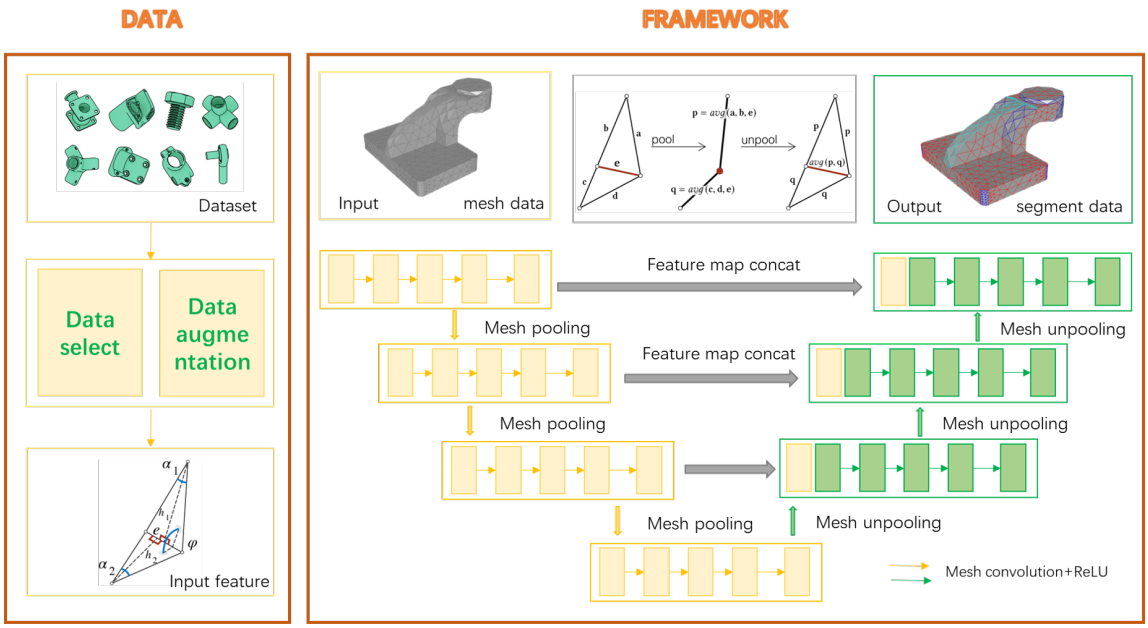

Figure 1: Baseline: Using the five special features as the input, we first select the varied data from the ABC dataset and then utilize data augmentation to boost the diversity of the data. The embedding data will be added to the ResUNet using a convolution and pooling method similar to MeshCNN.

mentation algorithms:

*Chart-based segmentation.*In this case, a given mesh surface decomposes into charts, with the geometric entities (i.e. vertices, edges and facets) of each chart satisfying similar values of a specific shape feature or descriptor (e.g. planarity, convexity and curvature) within a threshold or range.

*Part-based segmentation.*The primary objective of part-based segmentation is to decompose a given 3D object into meaningful parts (e.g. the fingers of a hand), though their meaningfulness depends on the application at hand. In fact, most algorithms to segment anthropomorphic, zoomorphic and hand-made objects take advantage of the minima rule. The aim in using this rule is to mimic the way the human visual system distinguishes segments from one another along bound-aries defined by negative minima of the principal curvatures[10, 1, 25]; In this latter case, parts are geometric primitives, that is, a 3D object decomposes into primitives such as planes, cylindrical patches, spherical patches and so forth. In Part-Based Mesh Segmentation[17]they have identified three main categories of part-based segmentation techniques.

- Volume-based segmentation. In this case, the segments are volumes. The input is a 3D volumetric mesh, which is then partitioned into 3D volumetric sub-meshes. These sub-meshes possibly correspond to meaningful parts. In fact, as argued by Hoffman and Richards[9], the volumetric convexity is oftenrelated to the human perception of the shape and, consequently,shape seg-mentation.
- Surface-based segmentation. In this technique, the segments are2D sub-meshes or regions of a 2D triangle mesh. Each regionconsists of a set of connected facets that have similar geometricproperties (e.g. convexity, curvature).
- Skeleton-based segmentation. In this technique, also known asskeletonization, the segments are line segments. The input is either a 3D volumetric mesh or a 2D surface mesh, but the output is a 1D skeleton that represents the structural shape of the mesh.

Recently, deep learning techniques have dominated many areas of research. Deep learning for 3D segmentation has also attracted increasing interest in the research community over the past decade due to its success in learning powerful features.However, 3D deep learning approaches still face many unsolved challenges. For example, features from RGB and depth channels are difficult to use. The irregularity of point clouds makes the low-heat features difficult to utilize, and the computational burden of converting them into high-resolution voxels is enormous.

Deep learning technology has also recently become the tool of choice for 3D task segmentation. This has led to an influx of methods in the literature that have been evaluated on different baseline datasets. Semantic segmentation[3],[5], [19] aims to predict object class labels such as table and chair. Instance segmentation[6], [16], [20] also distinguishes between different instances of the same class labels e.g. table one/two and chair one/two. Part segmentation [11], [22], [24] aims to further decompose instances into their different components such as arms, legs and backrest of the same chair. It is the next more elaborate level after instance segmentation, and its purpose is to mark different parts of the instance.

Data

we use the obj data formation of the ABC dataset[12]. It is a collection of one million computer aided design (CAD) models used to study deep learning methods and applications for geometry.

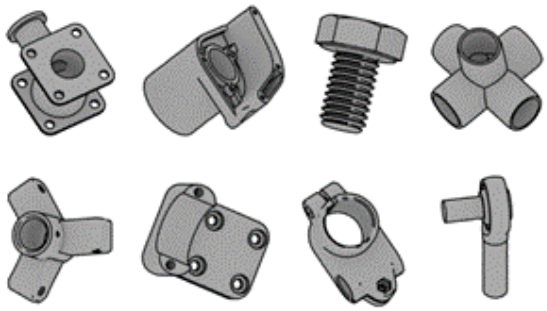

Figure 2: Examples from the ABC-Dataset. Most models are mechanical parts with sharp edges and well defined surfaces.

Data process.

In order to increase the classification diversity and keep the network stabilized we process the ABC dataset by data selection and data augmentation.

- data selection: fist we abandon the no-manifold mesh data and in order to keep the model pooling work we only chose the CAD structure in which the mesh is larger than its least pooling threshold.
- data augmentation: because the three dimensions dataset focuses on the easy structure and has some simple models which can not be segmented. As well, to increase the data multiplicity we do data augmentation on gmesh to comprise our existing data.

Gmsh is an open source 3D finite element mesh generator with a built-in CAD engine and post-processing program. It was designed to provide a fast, lightweight and user-friendly grid tool with parameter input and flexible visualization capabilities. Gmsh is built around four modules (geometry, mesh, solver and post-processing), that can be controlled from the command line via a graphical user interface using text files written in Gmsh's own scripting language(.geo files), or application programming interfaces such as C++, C, Python, Julia and Fortran.

Part-based segmentation framework

Input feature

In order to perform convolution on the grid better, we design the input edge features as a 5-dimensional vector for each edge: the dihedral angle of each face, the ratio of two side lengths, and the two interior angles. The edge ratio ranges between the perpendicular to each adjacent face and the length of the edge. By sorting the side length ratio and interior angle of two face-based features, the ambiguity and invariance of the sorting are ensured(see Figure 3). Since the observed features are relative, they are made invariant to rotation, uniform scaling, and translation.

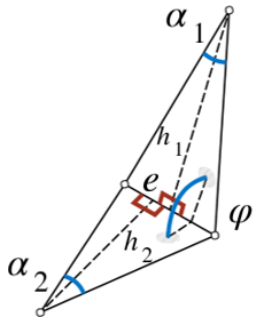

Figure 3: Select input features for every mesh

.

$$\vec{v}_e = \left(\varphi, \alpha_1, \alpha_2, \frac{|e|}{|h_1|}, \frac{|e|}{|h_2|}\right) \in R^5 \tag{1}$$

convolution

This paper assumes that all shapes are represented by manifold networks, possibly carrying boundary edges. Such a setting ensures that each edge is incident on at most two triangle faces and will be adjacent to two or four other edges. Since face vertices are arranged counterclockwise, there are two possible orders for the four adjacent vertices of each edge. For example, in Figure 4, the 1-ring neighbors of e can be sorted as (c,d,a,b) or (a,b,c,d). The order of arrangement depends on which face is set as the first neighbor. Different orderings make the receptive field of the convolution ambiguous, hindering the generation of invariant features.

In response to this problem, we take two measures to ensure the invariance of similarity transformations (scaling, rotation and translation) in the network. Edge input descriptors are first designed by only containing relative geometric features fixed to the similarity transformation. Second, the four 1-ring edges are fused into two pairs of ambiguous edges, and the new symmetric features are convolved to remove all order ambiguities

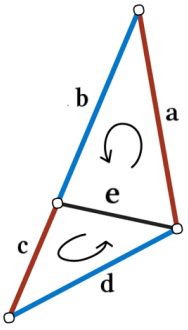

Figure 4: The four adjacent nodes of each edge have two possible orders

.

pooling

On each side of the collapsed edge, only one pair of edges must be merged. This can be checked by counting the joint neighbour vertices of the two merging vertices (there must be exactly two). Consider the following example where the red edge is being col-lapsed: he triangle between the orange and cyan edge is not manifold anymore.

We use a priority queue to prioritize the folding order of edges by the size of the edge features, allowing the network to choose the part that is relevant to the solving task. This allows the network to choose to non-uniformly collapse regions that contribute little to the loss. If you fold an edge that is adjacent to both faces, then three edges are deleted (Figure 6), since two faces merge into one edge[7]. Each face has three sides: the two smallest sides are adjacent. By taking the average value of each feature channel, the features on the three selected edges in each face are fused into a new edge feature.

The priority of edge collapse is distinguished by the strength of edge features, which are considered as l2-norm. There are two merge operations in the aggregation feature. A merge is performed on the triangles of each face of the minimum edge feature e and produces two new feature vectors. After each update of Pooling, the structural features will be saved, so that the global features can be preserved, which can reduce the running time and improve the efficiency when extracting and comparing the feature structure next time, but this will also increase a certain amount of storage space.

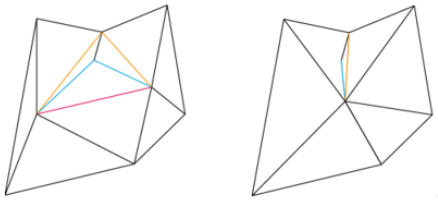

Figure 5: Mesh simplification technique edge collapse

.

unpooling

Unpooling can be understood to some extent as the re-verse of pooling[18].It is commonly used for recovering the virgin resolution lost in the pooling operation[21]. While pooling layers reduce the resolution of the feature activations (encoding or compressing information), unpooling layers increase the resolution of the feature activations (decoding or uncompressing information). Pooling operation also keep detailed records from merge operations (e.g., maximum position) and we will use them to expend the features in unpooling. The unpooling and pooling layers are in one-to-one correspondence, so that the mesh topology and edge features can be recovered by up-sampling,by storing the connectivity prior to pooling. Note that upsampling the connectivity is a reversible operation (just as in images).[7] After unpooling operation of the features extracted from the edges, we regain a graph with the adjacency from the original edge (before pooling) to the new edge (after pooling). Every edge feature that has undergone the unpooling operation is a weighted combination of edge features that have undergone the pooling operation.

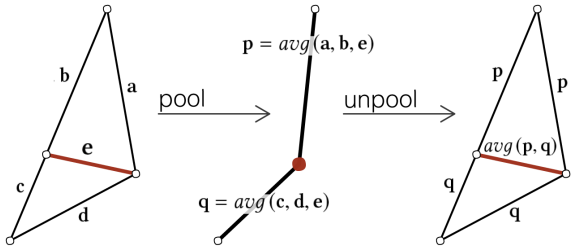

Figure 6: The case of pooling and unpooling

.

ResUnet

Resunet retains the advantages of both residual networks and UNet. The residual network simplifies the network training, makes the gradient explosion problem alleviated, and enables to build a deeper network structure; the constant mapping designed in the residual unit can facilitate the transfer of information from the lower to the higher layers of UNet, which enables to achieve better segmentation with fewer parameters, while the connection between the encoder and decoder corresponding to UNet can help the upper sampling layer to better recover image details[23].

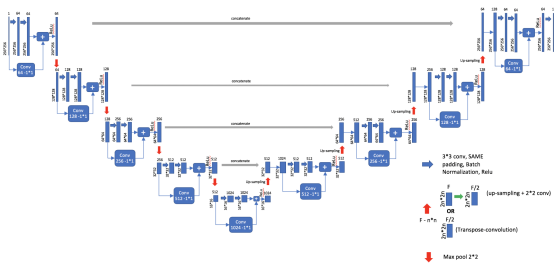

Figure 7: The architecture of ResUnet

.

And in order to avoid the data from augmentation prefer to predict the one special type, we use the AdamW opti-mizer.AdamW is an improved algorithm based on Adam+L2 regularization, including decoupled weight decay regularization.Using Adam to optimize the loss function with L2 regularization is not effective. If the L2 regularization term is introduced[4], the result of finding the gradient against the regularization term is added to the calculation of the gradient.Therefore if some weights are inherently larger, the corresponding gradient will also be larger, and since the subtraction term in the Adam calculation step will have the accumulation of dividing by the squared gradient, making the subtraction term small. Common sense says that the larger the weight should be penalized, but this is not the case in Adam.Instead, the weight decay is updated us-

ing the same coefficients for all weights, and the larger the weight the larger the penalty is obviously[14].

$$\theta_t \leftarrow \theta_{t-1} - \eta_t \left(\frac{\alpha \hat{\mathbf{m}}_t}{\sqrt{\mathbf{v}_t} + \epsilon} + \lambda \theta_{t-1} \right) \tag{2}$$

Discussion

The experiment (Figure 8) shows the pooling update helps us keep the structure because the former only pooling method focus on the local classification information, so it may be interesting for us to discover that it will be pooling a person end with its body. But by using the pooling update the human will keep its limb structure.

Max pooling and average pooling are widely used pooling techniques[10]. They are used in local and global pooling layers, the suitability of which depends on the application. Max pooling considers only the most activated elements in each feature map and treats all other activations as insignificant. This active element can be noisy.

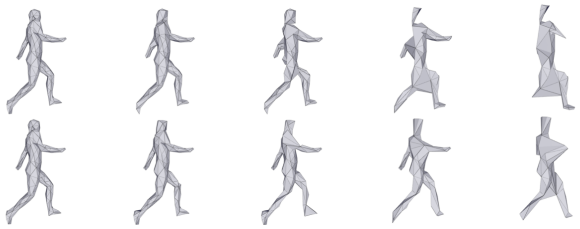

Figure 8: The pooling step on human body(The first line in the picture is the pooling process of meshCNN, and the following is our pooling process, which retains the global structural features).

And to show the data augmentation help some special type of structure to get more precise classification.As can be seen from the chart(Figure 9), for plane processing, the accuracy of segmentation and augmentation segmentation is very high and the difference is not large; the accuracy of

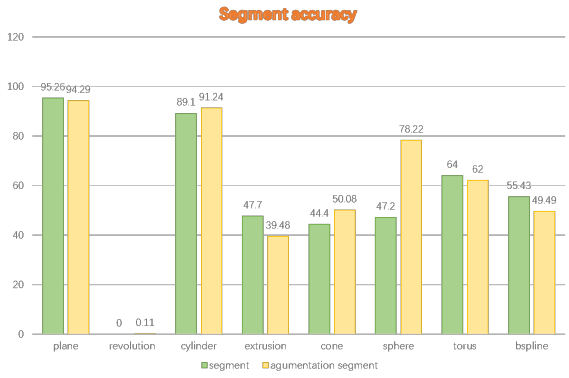

Figure 9: Accuracy of segment and augmentation segment

.

Framework	Segment accuracy
Our	93.23
MeshCNN[7]	92.3
MeshWalker[13]	94.8
PDMesh[15]	91.11

Table 1: Comparison of various frameworks

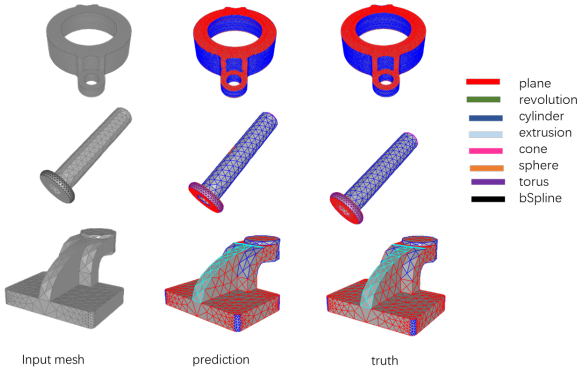

Figure 10: Some abc dataset segmentaion result

.

revolution segmentation is very low, but the accuracy of augmentation segmentation is slightly higher than that of ordinary segmentation; for sphere Processing, the accuracy of the augmentation segmentation is significantly higher than that of the normal segmentation, almost twice it.

And the most important the segmentation between us and other peoples work. Segmentation accuracy for the entire frame. Our method has great advantages.

Our experiments show that max pooling works well when class-specific features (e.g., abnormal regions in medical images) are small compared to the image size. During the learning phase of the network, only the network nodes connected to this maximum activation element will be updated, which makes the learning speed of the network slower. Max pooling is usually applied in the early stages of the network to capture important local image features. This is appropriate when the size of the image is large enough. From abc dataset do data selection and data augumentation ,and then use the mesh processing to input the resemble feature of mesh input the use resunet and meshcnn method to get segmentation result.

The experiment demonstrates that the network we enhance performs more effectively than the model it replaces. This is due to the use of data expansion, in which gmsh builds composite samples of parametric surfaces with uniformly distributed random parameters to enhance classification performance. In order to offer parameters for later editable CAD, we classify divided surfaces. We incorporate an update method throughout each phase of the pooling process to assist us maintain order. Additionally, in order to avoid overfitting, we train our model with ResUNet and the AdamW optimizer.

References

[1] Biederman, I. 1987. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2): 115.

[2] Chen-geng, N.; Yu-jie, L.; Zong-min, L.; and Hua, L. 2019. 3D Object Recognition and Model Segmentation Based on Point Cloud Data. *Journal of Graphics*, 40(2): 274.

[3] Cheng, Y.; Cai, R.; Li, Z.; Zhao, X.; and Huang, K. 2017. Locality-Sensitive Deconvolution Networks with Gated Fusion for RGB-D Indoor Semantic Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 1475–1483. IEEE Computer Society.

[4] Cortes, C.; Mohri, M.; and Rostamizadeh, A. 2012. L2 regularization for learning kernels. *arXiv preprint arXiv:1205.2653*.

[5] Fooladgar, F.; and Kasaei, S. 2020. A survey on indoor RGB-D semantic segmentation: from hand-crafted features to deep convolutional neural networks. *Multim. Tools Appl.*, 79(7-8): 4499–4524.

[6] Han, L.; Zheng, T.; Xu, L.; and Fang, L. 2020. OccuSeg: Occupancy-Aware 3D Instance Segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 2937–2946. Computer Vision Foundation / IEEE.

[7] Hanocka, R.; Hertz, A.; Fish, N.; Giryes, R.; Fleishman, S.; and Cohen-Or, D. 2019. Meshcnn: a network with an edge. *ACM Transactions on Graphics (TOG)*, 38(4): 1–12.

[8] He, Y.; Yu, H.; Liu, X.; Yang, Z.; Sun, W.; Wang, Y.; Fu, Q.; Zou, Y.; and Mian, A. 2021. Deep learning based 3D segmentation: A survey. *arXiv preprint arXiv:2103.05423*.

[9] Hoffman, D. D.; and Richards, W. A. 1984. Parts of recognition. *Cognition*, 18(1-3): 65–96.

[10] Hoffman, D. D.; and Singh, M. 1997. Salience of visual parts. *Cognition*, 63(1): 29–78.

[11] Kalogerakis, E.; Averkiou, M.; Maji, S.; and Chaudhuri, S. 2017. 3D Shape Segmentation with Projective Convolutional Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 6630–6639. IEEE Computer Society.

[12] Koch, S.; Matveev, A.; Jiang, Z.; Williams, F.; Artemov, A.; Burnaev, E.; Alexa, M.; Zorin, D.; and Panozzo, D. 2019. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9601–9611.

[13] Lahav, A.; and Tal, A. 2020. Meshwalker: Deep mesh understanding by random walks. *ACM Transactions on Graphics (TOG)*, 39(6): 1–13.

[14] Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

[15] Milano, F.; Loquercio, A.; Rosinol, A.; Scaramuzza, D.; and Carlone, L. 2020. Primal-dual mesh convolutional neural networks. *Advances in Neural Information Processing Systems*, 33: 952–963.

[16] Pham, Q.; Nguyen, D. T.; Hua, B.; Roig, G.; and Yeung, S. 2019. JSIS3D: Joint Semantic-Instance Segmentation of 3D Point Clouds with Multi-Task Pointwise Networks and Multi-Value Conditional Random Fields. *CoRR*, abs/1904.00699.

[17] Rodrigues, R. S.; Morgado, J. F.; and Gomes, A. J. 2018. Part-based mesh segmentation: a survey. In *Computer Graphics Forum*, volume 37, 235–274. Wiley Online Library.

[18] Turchenko, V.; Chalmers, E.; and Luczak, A. 2017. A deep convolutional auto-encoder with pooling-unpooling layers in caffe. *arXiv preprint arXiv:1701.04949*.

[19] Wang, W.; and Neumann, U. 2018. Depth-Aware CNN for RGB-D Segmentation. In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., eds., *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XI*, volume 11215 of *Lecture Notes in Computer Science*, 144–161. Springer.

[20] Wang, X.; Liu, S.; Shen, X.; Shen, C.; and Jia, J. 2019. Associatively Segmenting Instances and Semantics in Point Clouds. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 4096–4105. Computer Vision Foundation / IEEE.

[21] Xu, C.; Yang, J.; Lai, H.; Gao, J.; Shen, L.; and Yan, S. 2019. UP-CNN: Un-pooling augmented convolutional neural network. *Pattern Recognition Letters*, 119: 34–40.

[22] Xu, H.; Dong, M.; and Zhong, Z. 2017. Directionally Convolutional Networks for 3D Shape Segmentation. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2717–2726. IEEE Computer Society.

[23] Xu, J.; Luo, X.; Wang, G.; Gilmore, H.; and Madabhushi, A. 2016. A deep convolutional neural network for segmenting and classifying epithelial and stromal regions in histopathological images. *Neurocomputing*, 191: 214–223.

[24] Xu, Y.; Fan, T.; Xu, M.; Zeng, L.; and Qiao, Y. 2018. SpiderCNN: Deep Learning on Point Sets with Parameterized Convolutional Filters. *CoRR*, abs/1803.11527.

[25] Zhang, J.; Zheng, J.; Wu, C.; and Cai, J. 2012. Variational mesh decomposition. *ACM Transactions on Graphics (TOG)*, 31(3): 1–14.