Parts4Feature: Learning 3D Global Features from Generally Semantic Parts in Multiple Views

Zhizhong Han\textsuperscript{1,2}, Xinhai Liu\textsuperscript{1}, Yu-Shen Liu\textsuperscript{1}, Matthias Zwicker\textsuperscript{2}
\textsuperscript{1}School of Software, Tsinghua University, Beijing, China
\textsuperscript{2}Department of Computer Science, University of Maryland, College Park, USA

Abstract

Deep learning has achieved remarkable results in 3D shape analysis by learning global shape features from the pixel-level over multiple views. Previous methods, however, compute low-level features for entire views without considering part-level information. In contrast, we propose a deep neural network, called Parts4Feature, to learn 3D global features from part-level information in multiple views. We introduce a novel definition of generally semantic parts, which Parts4Feature learns to detect in multiple views from different 3D shape segmentation benchmarks. A key idea of our architecture is that it transfers the ability to detect semantically meaningful parts in multiple views to learn 3D global features. Parts4Feature achieves this by combining a local part detection branch and a global feature learning branch with a shared region proposal module. The global feature learning branch aggregates the detected parts in terms of learned part patterns with a novel multi-attention mechanism, while the region proposal module enables locally and globally discriminative information to be promoted by each other. We demonstrate that Parts4Feature outperforms the state-of-the-art under three large-scale 3D shape benchmarks.

1 Introduction

Learning 3D global features from multiple views is an effective approach for 3D shape understanding. A widely adopted strategy is to leverage deep neural networks to aggregate features hierarchically extracted from pixel-level information in each view. However, current approaches can not employ part-level information. In this paper, we show for the first time how extracting part-level information over multiple views can be leveraged to learn 3D global features. We demonstrate that this approach further increases the discriminability of 3D global features and outperforms the state-of-the-art methods on large scale 3D shape benchmarks.

It is intuitive that learning to detect and localize semantic parts could help classify shapes more accurately. Previous studies on fine-grained image recognition also employ this intuition by combining local part detection and global feature learning together. To learn highly discriminative features to distinguish subordinate categories, these methods try to first detect important parts, such as heads, wings and tails of birds, and then collect these part features into a global feature. However, these methods do not tackle the challenges that we are facing in the 3D domain. First, these methods require ground truth parts with specified semantic labels, while 3D shape classification benchmarks do not provide such kind of labels. Second, the part detection knowledge learned by these methods cannot be transferred for general purpose use, such as non-fine-grained image classification, since it is specified for particular shape classes. Third, these methods are not designed to aggregate part information from multiple images, corresponding to multiple views of a 3D shape in our scenario. Therefore, simultaneously learning part detection and further aggregating part-level information from multiple views become a unique challenge in 3D global feature learning.

To address these issues, we propose Parts4Feature, a deep neural network to learn 3D global features from semantic parts in multiple views. With a novel definition of generally semantic parts (GSPs), Parts4Feature learns to detect GSPs in multiple views from different 3D shape segmentation benchmarks. Moreover, it learns a 3D global feature from shape classification data sets, by transferring the learned knowledge of part detection, and leveraging the detected GSPs in multiple views. Specifically, Parts4Feature is mainly composed of a local part detection branch and a global feature learning branch. Both branches share a region proposal module, which enables locally and globally discriminative information to get promoted by each other.

The local part detection branch employs a novel neural network derived from Fast R-CNN [Girshick, 2015] to learn to detect and localize GSPs in multiple views. In addition, the global feature learning branch incrementally aggregates the detected parts in terms of learned part patterns with multi-attention. We propose a novel multi-attention mechanism to further increase the discriminability of learned features by not only highlighting the distinctive parts and part patterns but also depressing the ambiguous ones. Our novel view aggregation based on semantic parts prevents information loss caused by the widely used pooling, and it can understand each detected part in a more detailed manner. In summary, our contribu-
tions are as follows:

i) We propose Parts4Feature, a novel deep neural network to learn 3D global features from semantic parts in multiple views, by combining part detection and global feature learning together.

ii) We show that the novel structure of Parts4Feature is capable of learning and transferring universal knowledge of part detection, which allows Parts4Feature to leverage discriminative information from another source (3D shape segmentation) for 3D global feature learning.

iii) Our global feature learning branch introduces a novel view aggregation based on semantic parts, where the proposed multi-attention further improves the discriminability of learned features.

2 Related work

Mesh-based deep learning models. To directly learn 3D features from 3D meshes, different novel concepts, such as circle convolution [Han and others, 2016], mesh convolution [Han and others, 2017] were proposed to perform in deep learning models. These methods aim to learn global or local features from the geometry and spatial information on meshes to understand 3D shapes.

Voxel-based deep learning models. Similar to images, voxels have regular structure to be learned by deep learning models, such as CRBN [Wu and others, 2015], fully convolutional autoencoders [Sharma et al., 2016], CNNs [Qi et al., 2016], GAN [Wu and others, 2016]. These methods usually employ 3D convolution to better capture the contextual information in local regions. Moreover, Tags2Parts [Muralikrishnan et al., 2018] discovered semantic regions that strongly correlate with user-prescribed tags by learning from voxels using a novel U-Net.


View-based deep learning models. Similar to the light field descriptor (LFD), GIFT [Bai and others, 2017] measured the difference between two 3D shapes using their corresponding view feature sets. Moreover, pooling panorama views [Shi and others, 2015; Sifakis and others, 2017] or rendered views [Su and others, 2015; Han and others, 2019] is more widely used to learn global features. Different improvements from camera trajectories [Johns et al., 2016], view aggregation [Wang et al., 2017; Han and others, 2019a], pose estimation [Kanezaki et al., 2018] are also presented. However, these methods can not leverage part-level information. In contrast, Parts4Feature learns and transfers universal knowledge of part detection to facilitate 3D global feature learning.

3 Parts4Feature

Overview. Parts4Feature consists of three main components as shown in Fig. 1: a local part detection branch L, a global feature learning branch G, and a region proposal module R, where R is shared by L and G and receives multiple views of a 3D shape as input. We train Parts4Feature simultaneously under a local part detection benchmark Φ and a global feature learning benchmark Ψ. The local part detection branch L learns to identify GSPs in multiple views under Φ, while G learns a global feature from the detected GSPs in multiple views under Ψ.

For a 3D shape m in either Φ or Ψ, we capture V views v^i around it, forming a view sequence v = {v^i | i ∈ [1, V]}. First, the region proposal module R provides the features f_j of regions r_j proposed in each view v^i, where j ∈ [1, J]. Then, by analyzing the region features f_j in v, branch L learns to predict what and where GSPs are in multiple views. Finally, by aggregating the features f_j of the top K region proposals r_j in each v^i in v, the global feature learning branch G produces the global feature f of shape m. Our approach to aggregating region proposal features is based on N semantic part patterns θ_n with multi-attention for 3D shape classification, where θ_n are learned across all views in the global feature learning benchmark Ψ.

Generally semantic parts. We define a GSP as a local part in any semantic part category of any shape class, such as engines of airplanes or wheels of cars. Although our concept of GSPs simplifies all semantic part categories into a binary label by only determining whether a part is semantic or not, this allows us to exploit discriminative, part-level information from several different 3D shape segmentation benchmarks for global feature learning.

We use three 3D shape segmentation benchmarks involved in [Kalogerakis and others, 2017], including ShapeNetCore, Labeled-PSB, and COSEG to construct the local part detection benchmark Φ and provide ground truth GSPs. We also split the 3D shapes in each segmentation benchmark into training and test sets according to [Kalogerakis and others, 2017]. Fig. 2 shows the construction of ground truth GSPs. For each view v^i of a 3D shape m shown in Fig. 2(a), we obtain its ground truth segmentation visualized in Fig. 2(b) from the shape segmentation benchmark. Then, we can isolate each part category to precisely locate GSPs, as shown from Fig. 2(c) to Fig. 2(f). We emphasize each isolated part category in blue, where we locate the corresponding GSPs by computing the bounding box (red) of the colored regions. Finally, we show all GSPs in view v^i in Fig. 2(g). We collect all GSPs of shape m by repeating these procedures in all its V views. Note that we omit small GSPs (for example the landing gear in Fig. 2(f)) whose bounding boxes are smaller than 0.45 of the max bounding box in the same part category.

Region proposal module R. R provides region proposals r_j in all views v^i and their features f_j, which are then forwarded to the local part detection and global feature learning branches. Shared by all views v^i in v, R is composed of a Deep Convolutional Network (DCN), and a Region Proposal Network (RPN) with Region of Interest (RoI) pooling [Girshick, 2015].

DCN is modified from a VGG-CNN-M-1024 network [Chatfield et al., 2014], and it produces feature f_i for each view v^i as 512 feature maps of size 12 × 12. Based on f_i, RPN then proposes regions r_j in a sliding-window man-
At each sliding-window location centered at each pixel of $f^i$, a region $r^i_j$ is proposed by determining its location $t_R$ and predicting GSP probabilities $p_R$ with an anchor. The location $t_R$ is a four dimensional vector representing center, width and height of the bounding box. We use 6 scales and 3 aspect ratios to yield $6 \times 3 = 18$ anchors, which ensures a wide range of sizes to accommodate region proposals for GSPs that may be partially occluded in some views. The 6 scales relative to the size of the views are $[1, 2, 4, 8, 16, 32]$, and the 3 aspect ratios are $1 : 1$, $1 : 2$, and $2 : 1$. Altogether, this leads to $J = 2592 = 12 \times 12 \times 18$ regions $r^i_j$ in each view $v^i$.

To train RPN to predict GSP probabilities $p_R$, we assign a binary label to each region $r^i_j$ indicating whether $r^i_j$ is a GSP. We assign a positive label if the Intersection-over-Union (IoU) overlap between $r^i_j$ and any ground-truth GSP in $v^i$ is higher than a threshold $t_R$, and we use a negative label otherwise. In each view $v^i$, we apply RoI pooling over regions given by $\{r^i_j\}$ on feature maps $f^i$. Hence, the features $f^i_j$ of all $J$ region proposals $r^i_j$ are $512 \times 7 \times 7$ dimensional vectors, which we forward to the local part detection branch $L$.

In addition, we provide the features $f^i_k$ of the top $K$ regions $r^i_j$ according to their GSP probability $p_R$ to the global feature learning branch $G$.

**Local part detection branch** $L$. The purpose of this branch is to detect GSPs from the $J$ region proposals $r^i_j$ in each view $v^i$. We employ $L$ as an enhancer of RPN, where $L$ aims to learn what and where GSPs are in $v^i$ without anchors in a more precise manner. The intuition is that this in turn pushes RPN to propose more GSP-like regions, which we provide to the global feature learning branch $G$.

We feed the region features $f^i_j$ of $r^i_j$ into a sequence of fully connected layers followed by two output layers. The first one estimates the GSP probability $p_L$ that $r^i_j$ is a GSP using a sigmoid function as an indicator. The second one predicts the corresponding part location $t_L$ using a bounding box regressor, where $t_L$ represents the same bounding box parameters as $t_R$ in RPN. Similar to the threshold $t_R$, $L$ employs another threshold $t_{t_L}$ to assign positive and negative labels for training. Denoting the ground truth probabilities and locations of positive and negative samples in RPN $R$ as $p^0$ and $t^0$, and similarly for $L$ as $p^\prime$ and $t^\prime$, the objective function of Parts4Feature for GSP detection is formed by the loss in $R$ and $L$, which is defined for each region proposal as follows,

$$
O_{R+L}(p^R, p^L, p^\prime, p^\prime\prime, t_R, t_L, t^0, t^\prime) = O_p(p_R, p^\prime) + O_L(t_R, t^0) + O_p(p_L, p^\prime\prime) + O_L(t_L, t^\prime),
$$

where $O_p$ measures the accuracy in terms of GSP probability by the cross-entropy function of positive labels, while $O_L$ measures the accuracy in terms of location by the robust $L_1$ function as in Ren and others, 2015. The parameter $\lambda$ balances $O_p$ and $O_L$ in both $R$ and $L$. It works well in our experiments with a value of 1. In summary, Parts4Feature has the powerful ability to detect GSPs by simultaneously leveraging the view-level features $f^i$ in $R$ and the part-level features $f^i_j$ in $L$, which addresses the difficulty of GSP detection from multiple views caused by rotation and occlusion effects.

**Global feature learning branch** $G$. This branch learns to map the features $f^i_k$ of the top $K$ region proposals $r^i_j$ in each view $v^i$ in $v$ to the 3D global feature $f$. To avoid information loss caused by widely used pooling for aggregation, $G$ incrementally aggregates all $V \times K$ region features $f^i_k$ in terms of semantic part patterns $\theta_{t_{L_k}}$ with multi-attention, where we learn the parameters $\theta_{t_{L_k}}$ across all training data in the global feature learning benchmark $\Psi$. The motivation for learning part patterns to aggregate regions is that the appearance of GSPs
is so various that it would limit the discriminability of global features \( f \). Our multi-attention mechanism includes attention weights for view aggregation on the part-level and the part-pattern-level, denoted by \( \alpha \) and \( \beta \), respectively. Here, \( \alpha \) models how each of the \( N \) patterns \( \theta_n \) weights each of the \( V \times K \) regions \( r_{ik} \), while \( \beta \) measures how the final, global feature \( f \) weights each of the \( N \) patterns \( \theta_n \).

Specifically, we employ a single-layer perceptron to learn \( \theta_n \), where \( \theta_n \) has the same dimension as \( f_k \). \( \alpha \) is a \( (V \times K) \times N \) matrix, where each entry \( \alpha(i, k, n) \) is the attention paid to each of the \( (V \times K) \) regions \( r_{ik} \) by the \( n \)-th pattern \( \theta_n \). \( \alpha(i, k, n) \) is measured by a softmax function as \( \exp(w_n^T f_k + b_n)/\sum_{n'}^{N} \exp(w_n^T f_k + b_n') \). With \( \alpha \), we first aggregate all \( (V \times K) \) region features \( f_k \) into a pattern specific aggregation \( \varphi_n \) in terms of each pattern \( \theta_n \) by computing \( \sum_{i \in [1, V], k \in [1, K]} \alpha((i, k, n)) (\theta_n - f_k) \). Then, we further aggregate all \( N \) pattern specific aggregations \( \varphi_n \) into the final, global feature \( f \) of 3D shape \( m \). This is performed by linear weighting with the \( N \) dimensional vector \( \beta \), such that \( f = \sum_{n \in [1, N]} \beta(n) \varphi_n \). For clarity of exposition, we explain the details of how we obtain \( \beta \) further below.

Finally, we use \( f \) to classify \( m \) into one of \( C \) shape classes by a softmax function after a fully connected layer, where the softmax function outputs the classification probabilities \( p \), such that each probability \( p(c) \) is defined as \( \exp(u_c^T f + a_c)/\sum_{c' \in [1, C]} \exp(u_{c'}^T f + a_{c'}) \). The objective function of \( G \) is the cross entropy between \( p \) and the ground truth probability \( p' \),

\[
O_G(p, p') = -\sum_{c \in [1, C]} p'(c) \log p(c).
\] (2)

The intuition behind modelling part-pattern-level attention is to enable Parts4Feature to weight the pattern specific aggregations \( \varphi_n \) according to the 3D shape characteristics that it has learned. This leads Parts4Feature to differentiate shapes in greater detail. To implement this, \( \beta \) is designed to capture the similarities between each of the \( N \) pattern specific aggregations \( \varphi_n \) and the \( C \) shape classes. To represent the characteristics of \( C \) shape classes, we propose to employ the weights \( u_c \) in the fully connected layer before the last softmax function, as illustrated in Fig. 1. We first project \( \varphi_n \) and \( u_c \) into a common space using matrices \( W_1 \) and \( W_2 \). Then we compute normalized similarities using a linear mapping with \( w \) and \( g \) as follows, \( \beta = \text{softmax}(W_1 [\varphi^T_n]_N + W_2 [u^T_c]_N w + g) \), where learnable parameters \( W_1 \) and \( W_2 \) are \( N \times N \) and \( N \times C \) dimensional matrices, \( w \) and \( g \) are \( d \) and \( N \) dimensional vectors, \( \bullet^T \) means stacking all \( \circ \) vectors \( \bullet \) into a matrix row by row.

**Training.** We train \( R \) and \( L \) together under a local part detection benchmark \( \Phi \) and \( G \) under a global feature learning benchmark \( \Psi \). The Parts4Feature objective is to simultaneously minimize Eq. 1 and Eq. 2, which leads to the loss

\[
O = \frac{1}{\|\varphi\|} \sum_{(p, p') \in \Phi} O_{R+L}(pR, pL, p', p'', tR, tL, t', t'') + \frac{\eta}{\|\Psi\|} \sum_{p \in \Psi} O_G(p, p').
\] (3)

where the number of samples \( ||\cdot|| \) is a normalization factor and \( \eta \) is a balance parameter. Since \( R \) and \( L \) are based on the object detection architecture of Fast R-CNN [Girshick, 2015], we adopt the approximate approach in [Ren and others, 2015] to jointly train \( R \) and \( L \). First, we simultaneously update \( u_c \) in the softmax classifier in \( G \) by \( \partial O_G/\partial u_c \) and \( \partial O_G/\partial \beta \). This enables \( u_c \) to be learned more flexibly for optimization convergence, which is a connection across \( G \). For the \( \Phi = \Psi \) case, parameters in \( R \) and \( L \) can be simultaneously updated, otherwise, they are updated alternatively. For example, parameters in \( R \) and \( L \) are first updated under \( \Phi \), then, parameters in \( R \) (except RPN) and \( G \) are updated under \( \Psi \), and this process is iterated until convergence. In our following experiments we use \( \eta = 1 \).

**4 Experiments and analysis**

**Parameters.** We investigate how some important parameters affect Parts4Feature in shape classification under ModelNet [Wu and others, 2015].

We first explore the IoU thresholds \( S_R \) in \( R \) and \( S_L \) in \( L \) that are used to establish positive GSP samples using ModelNet40 [Wu and others, 2015] as \( \Psi \), as shown in Table 1, where we initially use \( V = 12 \) views, \( K = 20 \) regions, and \( N = 256 \) patterns. With \( S_R = 0.7 \) and increasing \( S_R \) from 0.5 to 0.8, the mean Average Precision (mAP) under the test set of \( \Phi \) decreases, and accordingly, the average instance accuracy under the test set of \( \Psi \) decreases, compared to the highest classification accuracy 93.40%. With \( S_L = 0.5 \), we also decrease \( S_R \) to 0.5 and increase it to 0.8 respectively. The mAP only slightly drops from 77.28 to 75.39 and 72.32, although the corresponding accuracy decreases too. However, the mAP and the accuracy are not strictly positive correlated, as shown by “(0.6,0.6)”, which has lower mAP but higher accuracy than “(0.8,0.5)” and “(0.5,0.5)”. This comparison also implies that \( S_L \) affects part detection more than \( S_R \).

Next, we apply the parameters setting “(0.7,0.5)” under ModelNet10 [Wu and others, 2015], as shown by the first accuracy of 95.26% in Table 2. Increasing \( S_L \) to 0.7 leads to an even better result of 96.15%. We also find the slight effect of \( K \), \( N \), and \( V \) on the performance.

We visualize part detection and multi-attention involved in our best result under ModelNet10 in Fig. 3 and Fig. 4, respectively. Although there are no ground truth GSPs under ModelNet10, Parts4Feature still successfully transfers the part-
Table 1: The effects of $S_R$ and $S_L$ on the performance of Parts4Feature under ModelNet40.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>$(0.7, 0.5)$</th>
<th>$(0.7, 0.6)$</th>
<th>$(0.7, 0.7)$</th>
<th>$(0.7, 0.8)$</th>
<th>$(0.5, 0.5)$</th>
<th>$(0.8, 0.5)$</th>
<th>$(0.6, 0.6)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>mAP</td>
<td>77.28</td>
<td>69.35</td>
<td>66.12</td>
<td>56.97</td>
<td>75.39</td>
<td>72.32</td>
<td>69.51</td>
</tr>
<tr>
<td>Acc</td>
<td>93.40</td>
<td>93.15</td>
<td>92.38</td>
<td>92.50</td>
<td>92.67</td>
<td>92.71</td>
<td>92.95</td>
</tr>
</tbody>
</table>

Table 2: The effects of $K$, $N$ and $d$ on the performance of Parts4Feature under ModelNet10.

<table>
<thead>
<tr>
<th>Metric</th>
<th>$S_L$ = 0.5</th>
<th>0.7</th>
<th>0.8</th>
<th>$K = 10$</th>
<th>30</th>
<th>$N = 128$</th>
<th>512</th>
<th>$V = 3$</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acc</td>
<td>95.26</td>
<td>96.15</td>
<td>94.38</td>
<td>94.38</td>
<td>94.93</td>
<td>94.49</td>
<td>95.04</td>
<td>94.27</td>
<td>94.93</td>
</tr>
</tbody>
</table>

Table 3: The view aggregation and attention comparison.

<table>
<thead>
<tr>
<th>Pooling</th>
<th>Attention</th>
<th>Methods</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxPool</td>
<td>NoAtt</td>
<td>90.97</td>
<td>92.84</td>
</tr>
<tr>
<td>MeanPool</td>
<td>PrAtt</td>
<td>92.18</td>
<td>93.17</td>
</tr>
<tr>
<td>NetVLAD</td>
<td>PrAtt</td>
<td>93.50</td>
<td>93.72</td>
</tr>
<tr>
<td>No L</td>
<td>MultiAtt</td>
<td>93.61</td>
<td>96.15</td>
</tr>
</tbody>
</table>

Figure 4: Multi-attention is visualized under the test set of ModelNet10. $\alpha$ and $\beta$ are shown as matrices and numbers.

Figure 5: The classification confusion matrix under ModelNet10.

**Classification.** Table 4 compares Parts4Feature with the state-of-the-art in 3D shape classification under ModelNet. The comparison are conducted under the same condition\(^1\).

Under both benchmarks, Parts4Feature outperforms all its competitors at the same condition, where “Our” are obtained with the parameters of our best accuracy under ModelNet40 in Table 1 and the ones under ModelNet10 in Table 2. This comparison shows that Parts4Feature effectively employs part-level information to significantly improve the discriminability of learned features. Parts4Feature is also outperforming under ShapeNet55 with the same parameters of our best results under ModelNet10, as shown by the comparison in the last three rows in Table 7.

To better demonstrate our classification results, we visualize the confusion matrix of our classification result under ModelNet10 and ShapeNet55 in Fig. 5 and Fig. 6, respectively. In each confusion matrix, an element in the diagonal line means the classification accuracy in a class, while other elements in the same row means the misclassification accuracy. The large diagonal elements show that Parts4Feature is good at classifying large-scale 3D shapes.

We also conduct experiments with reduced number of segmented shapes for training under ModelNet10. As shown in Table 5, trained by randomly sampled \(\{0\%, 1\%, 5\%, 10\%, 25\%, 50\%\}\) of 6,386 shapes, our results

\(^1\)We use the same modality of views from the same camera system for the comparison, where the results of RotationNet are from Fig. 4 (a) and (b) in https://arxiv.org/pdf/1603.06208.pdf. Moreover, the benchmarks are with the standard training and test split.
increase accordingly. The good results with 0% segmented shapes show that we not only learn from pixel-level information in 3D classification benchmarks, similar to existing methods, but also improve performance further by absorbing part-level information from 3D segmentation benchmark.

**Retrieval.** We further evaluate Parts4Feature in shape retrieval under ModelNet and ShapeNetCore55 by comparing with the state-of-the-art methods in Table 6 and Table 7. These experiments are conducted under the test set, where each 3D shape is used as a query to retrieve from the rest of the shapes, and the retrieval performance is evaluated by mAP. The compared results include LFD, SHD, Fisher vector, 3D ShapeNets [Wu and others, 2015], Pano [Shi and others, 2015], MVCN [Su and others, 2015], GIFT [Bai and others, 2017], RAMA [Sfikas and others, 2017] and Trip [He and others, 2018] under ModelNet.

As shown in Table 6, Table 7, our results outperform all the compared results in each benchmark. Besides Taco [Cohen and others, 2018] in Table 7, the compared micro-averaged results in Table 7 are from SHREC2017 shape retrieval contest [Savva and others, 2017] with the same names. In addition, the available PR curves under ModelNet40 and ModelNet10 are also compared in Fig. 7, which also demonstrates our outperforming results in shape retrieval.

Table 7: Retrieval and classification comparison in terms of Micro-averaged metrics under ShapeNetCore55.

<table>
<thead>
<tr>
<th>Methods</th>
<th>P</th>
<th>R</th>
<th>F1</th>
<th>mAP</th>
<th>NDCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canzakii</td>
<td>81.0</td>
<td>80.1</td>
<td>79.8</td>
<td>77.2</td>
<td>86.5</td>
</tr>
<tr>
<td>Zhou</td>
<td>78.6</td>
<td>77.3</td>
<td>76.7</td>
<td>72.2</td>
<td>82.7</td>
</tr>
<tr>
<td>Tatsuma</td>
<td>76.5</td>
<td>80.3</td>
<td>76.7</td>
<td>74.9</td>
<td>82.8</td>
</tr>
<tr>
<td>Furuya</td>
<td><strong>81.8</strong></td>
<td>68.9</td>
<td>71.2</td>
<td>66.3</td>
<td>76.2</td>
</tr>
<tr>
<td>fondes</td>
<td>74.3</td>
<td>67.7</td>
<td>69.2</td>
<td>62.2</td>
<td>73.2</td>
</tr>
<tr>
<td>Deng</td>
<td>41.8</td>
<td>71.7</td>
<td>47.9</td>
<td>54.0</td>
<td>65.4</td>
</tr>
<tr>
<td>Li</td>
<td>53.5</td>
<td>25.6</td>
<td>28.2</td>
<td>19.9</td>
<td>33.0</td>
</tr>
<tr>
<td>Mk</td>
<td>79.3</td>
<td>21.1</td>
<td>25.3</td>
<td>19.2</td>
<td>27.7</td>
</tr>
<tr>
<td>Su</td>
<td>77.0</td>
<td>77.0</td>
<td>76.4</td>
<td>73.5</td>
<td>81.5</td>
</tr>
<tr>
<td>Bai</td>
<td>70.6</td>
<td>69.5</td>
<td>68.9</td>
<td>64.0</td>
<td>76.5</td>
</tr>
<tr>
<td>Taco</td>
<td>70.1</td>
<td>71.1</td>
<td>69.9</td>
<td>67.6</td>
<td>75.6</td>
</tr>
<tr>
<td>Our</td>
<td>62.0</td>
<td>80.4</td>
<td>62.2</td>
<td><strong>85.9</strong></td>
<td><strong>90.2</strong></td>
</tr>
<tr>
<td>SVSL[Han and others, 2019a]</td>
<td></td>
<td></td>
<td></td>
<td>85.5</td>
<td></td>
</tr>
<tr>
<td>VIPG[Han and others, 2019b]</td>
<td></td>
<td></td>
<td></td>
<td>83.0</td>
<td></td>
</tr>
<tr>
<td>Our classification</td>
<td></td>
<td></td>
<td></td>
<td><strong>86.9</strong></td>
<td></td>
</tr>
</tbody>
</table>

5 Conclusions

Parts4Feature is proposed to learn 3D global features from part-level information in a semantic way. It successfully learns universal knowledge of generally semantic part detection from 3D segmentation benchmarks, and effectively transfers the knowledge to other shape analysis benchmarks.

Table 5: The effect of less segmented shapes for training.

<table>
<thead>
<tr>
<th>Acc</th>
<th>0%</th>
<th>1%</th>
<th>5%</th>
<th>10%</th>
<th>25%</th>
<th>50%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
<td>93.0</td>
<td>93.5</td>
<td>93.8</td>
<td>93.8</td>
<td>94.1</td>
<td>94.3</td>
<td><strong>96.15</strong></td>
</tr>
<tr>
<td>Class</td>
<td>92.7</td>
<td>93.1</td>
<td>93.4</td>
<td>93.6</td>
<td>94.0</td>
<td>94.1</td>
<td><strong>96.14</strong></td>
</tr>
</tbody>
</table>
by learning 3D global features from detected parts in multiple views. Parts4Feature makes it feasible to improve 3D global feature learning by leveraging discriminative information from another source. Moreover, our novel view aggregation with multi-attention can also benefit Parts4Feature to learn more discriminative features than widely used aggregation procedures. Our outperforming results show that Parts4Feature is superior to other state-of-the-art methods.

6 Acknowledgments

This work was supported by National Key R&D Program of China (2018YFB0505400) and NSF under award number 1813583. We thank all anonymous reviewers for their constructive comments.

References


[Han et al., 2019] Zhizhong Han, Mingyang Shang, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. Y2seq2seq: Cross-modal representation learning for 3D shape and text by joint reconstruction and prediction of view and word sequences. In AAAI, 2019.


