IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 11, 2019, accepted December 28, 2019, date of publication January 1, 2020, date of current version January 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2963511

Two-Stream Network Based on Visual Saliency Sharing for 3D Model Recognition

WEIZHI NIE[1], LU QU[1], MINJIE REN[1], QI LIANG[1], YUTING SU[1], YANGYANG LI[2], AND HAO JIN[2]

[1]School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China
[2]National Engineering Laboratory for Public Safety Risk Perception and Control by Big Data (PSRPC), CAEIT, Beijing 100041, China

Corresponding authors: Minjie Ren (renminjie@tju.edu.cn) and Qi Liang (tjuliangqi@tju.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772359, Grant 61572356, Grant 61872267, and Grant 61502477, in part by the Grant of 2019 Tianjin New Generation Artificial Intelligence Major Program, in part by the Grant of Tianjin New Generation Artificial Intelligence Major Program under Grant 18ZXZNGX00150, in part by the Grant of Elite Scholar Program of Tianjin University under Grant 2019XRX-0035, in part by the Tianjin Science Foundation for Young Scientists of China under Grant 19JCQNJC00500, and in part by the Director Foundation Project of National Engineering Laboratory for Public Safety Risk Perception and Control by Big Data (PSRPC).

ABSTRACT Shape representation for 3D models is an important topic in computer vision, multimedia analysis, and computer graphics. Recent multiview-based methods demonstrate promising performance for 3D model recognition and retrieval. However, most of the multiview-based methods focus on the visual information from the taken views and ignore correlation information among these views, which means the similarity and differentiation of multiple views have lost in their methods. In order to address this issue, we propose a novel two-stream network architecture for 3D model recognition and retrieval. The proposed network includes two sub-networks: a multi-view convolutional neural network (MVCNN) that extracts the view information from the taken views, and an Visual Saliency model that defines the weight of views based on the similarity and differentiation information of multiple views. Special, the weight of views defined by the Visual Saliency model can effectively be used to guide the visual information fusion in MVCNN model. This design can make the MVCNN model save visual information and the correlation information of these views in the learning step. Finally, we employ early-fusion method to fuse the feature vectors from MVCNN model and Visual Saliency model respectively, to generate the shape descriptor for 3D model recognition and retrieval. The experimental result on two public datasets, ModelNet40 and ShapeNetCore55, demonstrates the correlation information of multiple views is crucial for view-based 3D model recognition methods and the proposed method can achieve the state-of-the-art performance on both 3D object classification and retrieval.

INDEX TERMS 3D model, view-based, classification, retrieval, MVCNN, LSTM.

I. INTRODUCTION

In recent years, 3D technologies became popular in the folk gradually with the application in film and television industry. People can see the 3D models almost everywhere, so it's natural and reasonable to explore the more efficient methods to learn the representation of 3D models. Besides, with the development of computer vision and 3D reconstruction technology, 3D model recognition has became a fundamental task in shape analysis which is the most crucial technology for processing and analyzing 3D data. Thanks to the powerful deep learning neural networks and the availability of large-scale labeled 3D model collections, lots of deep networks have been proposed for 3D model recognition, such as MVCNN [1], 3D modelNets [2], PointNet [3], VoxNet [4].

Among the current methods, view-based methods perform best. One well-known example of view-based methods is Multi-View Convolutional Neural Networks [1] (MVCNN). As a combination of multiple 2D projection features learned by CNN within an end-to-end trainable fashion, this method have made the milestone for 3D model recognition and achieve the state-of-the-art performance at the time. Inspired by the success of MVCNN, various researcher have tried to build an unified deep learning model that can benefit from the projected view images to perform the tasks of 3D object classification and retrieval. However, in these methods,

The associate editor coordinating the review of this manuscript and approving it for publication was Li Zhang.

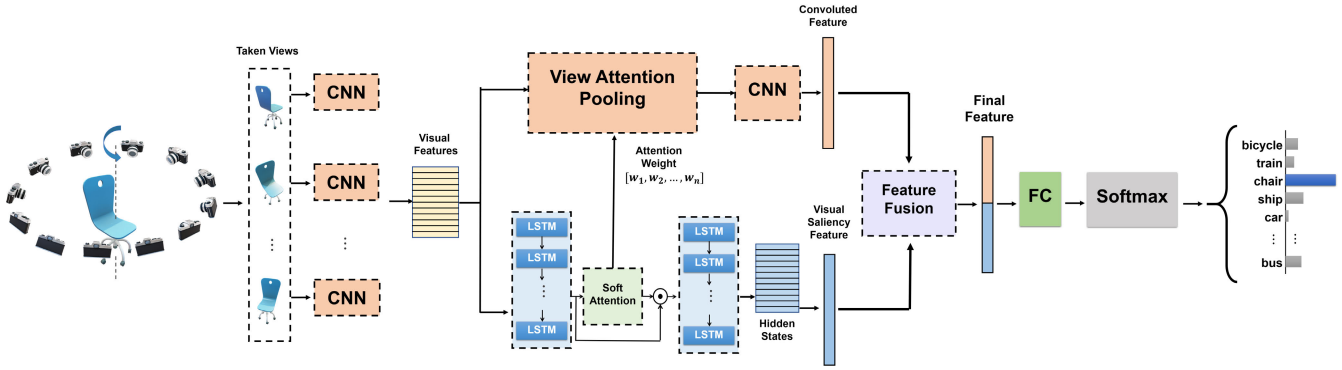


FIGURE 1. The framework of our method, which includes two sub-networks. The first sub-network focus on the multi-view of 3D model to output the visual feature vector and the second sub-network focus on the sequence-view of 3D model to output the serialized feature vector. The attention pooling part is a view-level attention to convoluted feature. Finally, we concatenate two-stream output features as our final feature for 3D model recognition.

we note that all views are treated equally to generate the shape descriptor. The similarity and differentiation between different views of the model are ignored through these networks. For example, in MVCNN, the visual features are fed to a view-pooling layer to generate a descriptor whereas the view-pooling layer only preserves the information from the related view with the maximum values and discards other information of multiple views. Actually, it is important to exploit the similarity and differentiation of the multiple views for analysing the 3D objects.

In order to alleviate this issue, we propose a novel two-stream network architecture based on multiple views taken from the 3D models. The two sub-networks of the proposed network can be briefly described as: a multi-view convolutional neural network (MVCNN) model that extracts the view information from the taken views, and a Visual Saliency model, which is constructed by the LSTM modules and soft-attention module, that defines the weight of views based on the similarity and differentiation information of multiple views. Especially, the weight of views defined by the Visual Saliency model can effectively be used to guide the visual information fusion in MVCNN model. The purpose of this design is to make the MVCNN model both save visual information and the correlation information of these views in the learning step. Finally, we employ early-fusion method to fuse the feature vectors form MVCNN model and Visual Saliency model respectively, to generate the shape descriptor for 3D model recognition and retrieval. The whole network structure can be seen in Fig.1.

The contributions of this paper can be summarized as follows:

- We propose a novel two-stream network based on visual differentiation information, consisted of a MVCNN model and a Visual Saliency model. Unlike existing view-based methods, we both save the visual information and the correlation information by updating the weights of different views from the Visual Saliency model.
- We effectively utilize the weight of views defined by the Visual Saliency model to guide the visual information fusion in MVCNN model. The design of this network architecture aims at allowing the MVCNN model to save both visual information and the correlation information from the taken views.
- We design different experiments to verify the significance of the correlation information in the multiview-based methods and the effectiveness of our network. The comparison with recent effective methods on the public dataset turns out we achieve the state-of-the-art performance, which means our network obtains a better representation for the 3D models. The final experimental result also demonstrates the superiority of our methods.

We organize the rest of this paper as follows: in Section 2, we do a coarse review of the current related work. In Section 3, we describe our networks architecture in detail. We provide related experiments, results and analysis in Section 4. Finally, we conclude this paper in Section 5.

II. RELATED WORKS

Recently, a large number of 3D model analysis methods based on deep learning neural networks have been proposed. In general, these methods could be roughly categorized into two classes: 3D model-based methods and view-based methods. We briefly review some typical methods on the 3D model representation problem in this section.

A. MODEL-BASED METHODS

Model-based methods learn the representations of models directly from 3D data formats, such as voxel meshes [2], [4]–[7], polygon meshes or surfaces [8]–[11], and point clouds [3], [12]. For example, Chopra *et al.* [13] proposed a unsupervised method to learn the 3D local features. In addition, these features are represented by surface patterns that capture common geometries and structures in a large number of 3D local regions. For 3D meshes feature learning, Mesh convolutional restricted Boltzmann machines (MCRBMs) is proposed by Han *et al.* [14] in order to learn the global features of meshes. MeshNet [15] is proposed by Feng et al., which can alleviate the problem

of the complexity and irregularity from the mesh format and perform three-dimensional shape representation by employing the face units and feature splitting. Besides, it is recommended that the DLAN [16] network directly process the local area of the 3D model and aggregate the local 3D rotation invariant features on the retrieval task. Klokov and Lempitsky [17] proposed Kd networks, which can handle unstructured point clouds and use learning features to perform retrieval tasks. Wang *et al.* [18] put forward a 3D CNN network based on the octree representation, which greatly improves the computational efficiency compared with the traditional full-voxel-based representations. Recently, DGCNN [19] was proposed by Wang et al., which focus on the point cloud feature learning. To both maintain permutation invariance and capture the local geometric featuresof point cloud, they proposed a new neural network module named EdgeConv. Xie *et al.* [20] proposed ShapeContextNet for point cloud recognition. Unlike previous works, the ShapeContextNet focuses on the concept of shape context and develops a new representation of point cloud. Moreover, they achieve competitive results on several benchmark datasets.

B. VIEW-BASED METHODS

In view-based methods, input data are the views taken from different angles of the 3D object, and these views can be easily captured compared to other methods such as point cloud structures. Recently, view-based methods attracted more attention due to the 3D models can be simply realized by the view representations [21]. And View representations using deep learning schemes usually refers to the use of mature models such as VGG [22], GoogLeNet [23] and ResNet [24]. Based on the structure of MVCNN, a compact shape descriptor can be extracted from multiple rendered views of an object using CNN with a view-pooling layer. MVCNN is significantly superior to the hand-crafted based methods and 3D modelNets on the ModelNet40 dataset. To take advantage of the structural information in the views of 3D objects, Sfikas *et al.* [25] proposed a method for capturing PANORAMA view features, which aims to achieve 3D model continuity and minimize data preprocessing by constructing augmented 3D model representation. In [26], an inductive multi-hypergraph learning algorithm is proposed. The goal of this algorithm is to learn the optimal projection of multi-model training data, and obtain the combination weight of optimal multi-hypergraph and the projection matrices simultaneously. Besides, Bai *et al.* [27] proposed a real-time 3D model search engine GIFT to accelerate with the GPU and two inverted files. In several shape benchmarks, GIFT is significantly better than the most advanced methods in terms of retrieval accuracy. Kanezaki *et al.* [28] using a cylindrical panorama around the main axis of the 3D model. Feng *et al.* [29] proposed GVCNN which considers capturing the hierarchical correlation of views to produce a more discerning 3D model description and also achieve better performance. In [30], a siamese CNN-BiLSTM network was proposed for 3D model representation learning. In order to aggregate information from all the views, the bidirectional LSTM is adopted after extracting the view features from the CNN model. Finally, the contrastive loss function is also employed for minimizing the distance of shapes with the same label, otherwise maximizing. Inspired by n-gram models in natural language processing, He *et al.* [31] proposed VNN for effecient aggregating all the view features to one discriminative shape descriptor. The spatial information across multiple views is captured by VNN, which divides the view sequence into a set of visual n-grams. VNN achieves outperforming results on several benchmark datasets.

However, one thing we should pay more attention to is that most of the existing multi-view based methods treat all the views equally, ignoring the correlated and discriminate information of multiple views, which limits the performance for the classification and retrieval task. Moreover, compare to the multi-view based methods employing the RNN architectures, our two-stream network not only use the LSTM to aggregate all the view features but also fuse the visual information from the CNN model and correlated information from LSTM model for the 3D model recognition. The view weights from the soft-attention module are also not just used for the attentional pooling in the CNN model but effectively work in the correlated information learning combined with the LSTM. The advantages of the CNN and LSTM are both leveraged to learn a more robust and discriminative shape descriptor.

III. OUR APPROACH

In this section, we give a detailed introduction to our method. The input of our two-stream network is a sequence of 2D views, which are rendered images of 3D models captured by predefined camera array. The camera array was set up around the z-axis with the interval of 30°. Therefore a sequence of 12 views are rendered from the models, which is the input of our network. The input views are firstly passed through CNN to get the visual features. Then we feed the visual features into two branches: the MVCNN branch and the Visual Saliency branch. Due to the superiority of the sequential representation learned from LSTM network structure, we utilize two LSTM layers and soft-attention mechanism for feature learning in our Visual Saliency branch. As the Fig.1 shown, the first LSTM module and soft-attention mechanism are adopted to generate the view weights for the attentional pooling and the correlated information learning. Next the Visual Saliency features from the Visual Saliency branch are fused with the convoluted features for obtaining the final feature of our network, which is used both for classification and retrieval tasks. We give a detailed description according to the following parts: (1) attention based view weight calculation; (2) view attention pooling; (3) final shape descriptor generation.

A. ATTENTION BASED VIEW WEIGHT CALCULATION

In order to use a set of views to represent each 3D model, the NPCA [32] method and the visual tool developed by OpenGL are used to normalize each 3D model and extract a set of views from each 3D model, respectively. Here, the extracted views wrapped around the model were extracted through 30 degrees on the Z axis. Therefore, there are 12 views, which can be seen as a sequence of images, are extracted to represent the visual and structure information of the 3D model. Since ResNet18 [24] achieves relatively better trade-off between accuracy and memory cost among several classical CNN models (e.g., AlexNet [33], VGG-Net [22]) [34], we then employ ResNet18 to extract the feature vector of each view. With the residual connections between standard convolution layers, ResNets can effectively improve and accelerate the optimization process for very deep networks. There are 17 convolutional layers $conv_1$ - $conv_{17}$ and a fully connected layer fc_{18} in ResNet18 network. In our work, the output of $conv_{17}$ is used as the feature vector for each view with the dimension of 4096.

Based on the sequence structure of the views, we propose a general approach based purely on neural networks to assign the weight for each taken view. Since LSTM has been successfully used in many fields [35]–[37], we utilize LSTM and soft-attention mechanism to weight each visual feature vector $V = \{v_1, \ldots, v_n\}$ in the Visual Saliency branch. This approach has been used successfully by Xu *et al.* [38] for exploiting spatial structure underlying an image. The reason for adapting attention mechanism is to give importance to those features which hold more significance. The multiplication of this attention weights with the extracted CNN feature vectors are used for the view attention pooling in our MVCNN branch. Next we will detail the calculation of our attention based view weight.

After we got the visual feature vector $V = \{v_1, \ldots, v_n\}$, a long short-term memory (LSTM) network is adopted in our Visual Saliency model to exploit structure information of these visual features. The details of the LSTM we use are described in Zaremba *et al.* [39].

The LSTM mainly maintains the hidden state h_t and an internal memory state c_t of an RNN. The correlation between the hidden state h_t and the memory state c_t is computed by an output gate:

$$h_t = o_t \bigodot \tanh(c_t). \tag{1}$$

where $\bigodot$ represents the element-wise multiplication. We can calculate the o_t as follows:

$$o_t = \sigma(W_o h_{t-1} + U_o v_{i,t} + b_o) \tag{2}$$

where σ is a logistic sigmoid function and $v_{i,t}$ is feature vector of the i-th view in the t time. W_o, U_o, b_o are respectively, the weight matrices for the previous hidden state and the bias. The current memory state c_t is determined by the previous memory state c_{t-1} and the updated memory $\tilde{c}_t$:

$$c_t = f_t \bigodot c_{t-1} + i_t \bigodot \tilde{c}_t \tag{3}$$

where f_t, i_t are, in order, the forget gates and the input gates, which are computed by:

$$\begin{aligned} f_t &= \sigma(W_f h_{t-1} + U_f v_{i,t} + b_f) \\ i_t &= \sigma(W_i h_{t-1} + U_i v_{i,t} + b_i) \end{aligned} \tag{4}$$

The current updated memory $\tilde{c}_t$ is computed by:

$$\tilde{c}_t = \tan(W_c v_{i,t} + U_c h_{t-1} + b_c) \tag{5}$$

where W_c, U_c, b_c are, respectively represents the weight matrices and the bias.

In order to discard irrelevant information and minimize the task complexity, attention mechanism is widely used in lots of fields, which makes neural networks focus on some particular portions of the input image. Therefore, in this work, soft attention mechanism is employed to compute the weight α_i based on the previous hidden state h_{t-1}.

$$\begin{aligned} e_i &= \mathbf{w}^\top tanh(W_a v_{i,t} + U_a h_{t-1} + b_a) \\ \alpha_i &= exp\{e_i\} / \sum_{j=1}^{n} exp\{e_j\} \\ \sum_{i=1}^{n} \alpha_i &= 1 \end{aligned} \tag{6}$$

where w, W_a, U_h and b_a are the parameters that are estimated together with the whole network.

B. VIEW ATTENTION POOLING

Unlike the max pooling used in MVCNN [1], we consider that a soft attention pooling will help the network achieve better representations of 3D models. According to the above steps, we have calculated soft-attention weight α_i, which reflects the relevance of the i-th temporal feature in the input images. In our MVCNN branch, we utilize the average of the dynamic weighted sum of the multi-view feature vectors such that

$$\psi(V) = (\sum_{i=1}^{N} \alpha_i v_i)/N \tag{7}$$

where N is the number of input views and $V = \{v_1, \ldots, v_n\}$ is the visual feature sets of the 3D object. After attention pooling, the output $\psi(V)$ then passed through the CNN2 to obtained the feature vector of our MVCNN branch.

C. FINAL SHAPE DESCRIPTOR GENERATION

As stated in [40], LSTMs can be trained to link time intervals which are over 1000 steps even for noisy sequences without losing short-time-lag capabilities. Hence we can easily extract the 3D model representation, including the holistic and correlation information, from the last cell state which takes into account all the views in a 3D model. Besides, MVCNN had proved its effectiveness in utilizing the visual information to address the 3D model analysis problem. We note that not all the views are equally important for discriminating a particular model and therefore attention mechanism helps to calculate the relative importance of the

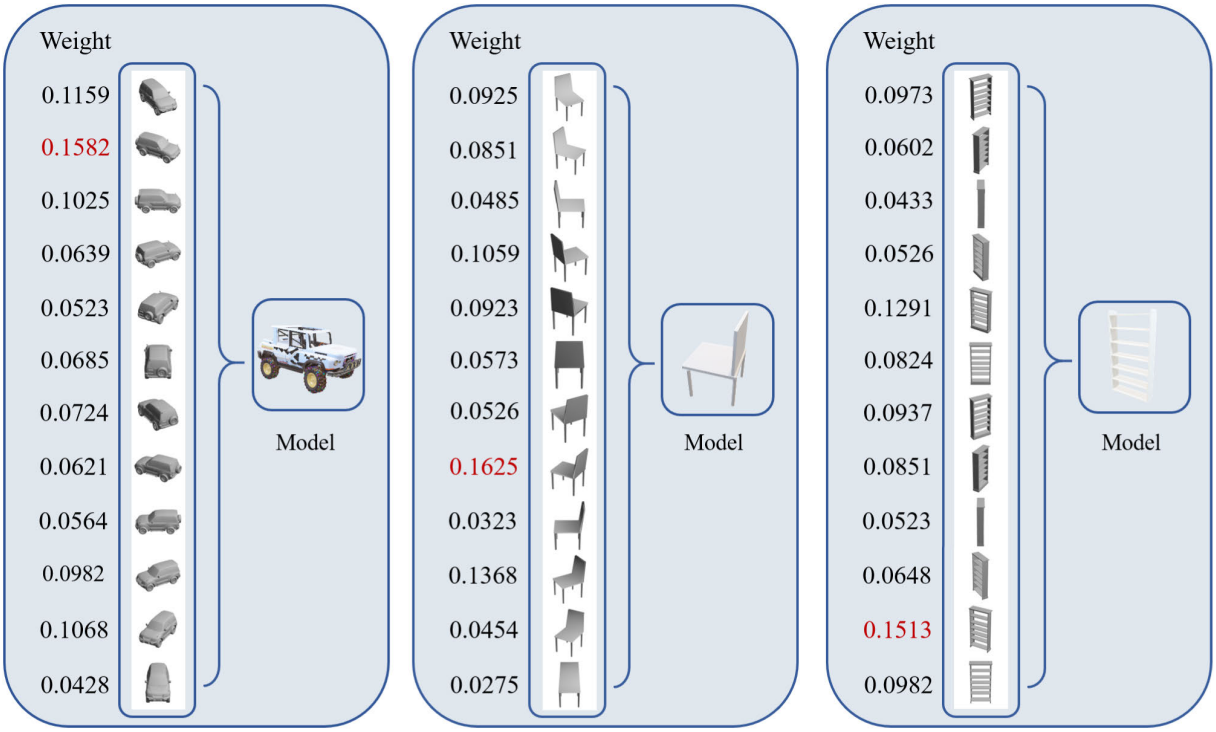


FIGURE 2. The red value is the highest attention weight, which means the characteristic view is selected by the soft-attention model.

3D model views by assigning weights to all the views, which is making the model focus on more representative extracted views of the 3D objects. To both retain the advantages of these two models, a comprehensive feature representation is obtained from a concatenation of feature vectors form MVCNN model and Visual Saliency model respectively. This is our final shape descriptor of the entire sequence of views for the classification and retrieval task.

IV. EXPERIMENTS

In this section, the experiment for proving the reasonableness of our design network is firstly provided, and then we compare the proposed network with the current effective methods to verify the superiority of our method. We also investigate the influence of two important parameters, including the input views' order and number, on the performance of 3D model recognition. The experiments are performed on two public datasets, ModelNet40 and ShapeNetCore55. Next, we will present the related experiments, results and analysis in detail.

A. DATASET

The ModelNet40 and ShapeNetCore55 datasets are used to evaluate the performance of the proposal method on the 3D model recognition task.

- ModelNet40 is a subset of ModelNet, which consists of 12,311 CAD models and these models are divided into 40 categories. In ModelNet40, the training subset and test subset respectively consist 9843 and 2468 models, These models ware cleaned manually, but pose normalization was not performed.
- ShapeNetCore55 is a subset of ShapeNet, which contains about 51,300 3D models in 55 common categories, and each category is subdivided into several subcategories. There are three subsets in the ShapeNetCore, consisting of 70%/10%/20% training / validation / testing splits. Two dataset versions of ShapeNetCore are available: consistent alignment (regular dataset), and a more challenging dataset where the model is disturbed by random rotation. The models in ShapeNetCore55 dataset are provided in OBJ format.

Due to the 3D models are represented as polygonal meshes in these datasets, following the MVCNN [1], we render them into multiple views to obtain the multi-view training and testing sets. There are 12 view images are rendered for each model by placing 12 virtual cameras around the mesh. These virtual cameras were pointed towards the centroid of the model, and elevated 30 degrees from the ground plane.

B. EVALUATION CRITERIA

In our experiments, the classification accuracy is employed to evaluate the classification performance of the proposed method. As for the 3D model retrieval task, several evaluation metrics are utilized, including Precision Recall Curve, NN, FT, ST, F-Measure, DCG, ANMRR, mAP [41]. Here, the lower ANMRR value means better retrieval performance, others the higher the better.

- The Precision-Recall Curve (PR-Curve) can comprehensively demonstrate retrieval performance. To change the threshold, which is used to distinguish the

irrelevance and correlation in the object retrieval, it can jointly consider the accuracy and recall metrics.

- The Nearest Neighbor (NN) indicates the percentage of the closest matching objects.
- The First Tier (FT) is calculated by the recall of the top N matching results, where the N is the number of relevant 3D objects in the dataset.
- The Second Tier (ST) employs the top 2N matching results to calculated the value, which is similar to the FT.
- The F measure (F) is a synthetical measurement, which considers both precision and recall based on the top 20 returned results.
- The Discounted Cumulative Gain (DCG) is a statistic that give more attention on the top matching results.
- The Average Normalized Modied Retrieval Rank (ANMRR) presents the ranking performance of the ranking list, which takes the ranking information of the relevant objects into account from the most frequently retrieved objects.
- The Mean Average Precision (mAP) is a ranking measure, which can solve the single point value limitation of Precision, Recall and F-measure.

C. IMPLEMENTATION DETAILS

Our method is tested on an Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz CPU system, with 32G RAM and a GeForce GTX1080 Ti GPU, with 12G RAM. We code in Matlab to extract the view images on an Intel(R) Core(TM) i5-6500 CPU @3.2GHz, with 8G RAM. The visual representation of the model with 12 viewpoint costs about 2.8s, while the images are sized to 600 $\times$ 600 pixels. The model of MVCNN, pre-trained by ImageNet1K, is utilized in our work and fine-tuned on all view images of models in the training set. Proposed network is fine-tuned on ModelNet40, which consits 9843 models in the trainning set and there are 12 extracted views for each model.

We utilize PyTorch platform to make all experiments. As for the computational cost, we use GTX1080 Ti GPUs and accelerate the training process by the CUDA instruction set on the GPU. It takes about 30 minutes to train each epoch of the whole network. The learning rate is set as 0.0001 at the beginning of joint training and decreases to 0.000001 after about 36_{th} epochs with the batch size set as 16. We obtain the best result of the whole network at about the 36_{th} epoch of joint training. Finally, we make a discussion on the view order and number. Meanwhile, we choose the best result to compare with current state-of-the-art methods.

D. EXPERIMENT FOR VALIDATING THE EFFECTIVENESS OF OUR METHOD

To validate the effectiveness of our method, we design the experiment for every components of our network. As the Tab. 1 shown, we use different parts of our network to perform the 3D model classification on the ModelNet40 dataset. The attention weight is employed to guide the visual features pooling in the MVCNN model and its effectiveness

TABLE 1. Comparison on the different components of our network for classification task on the ModelNet40 dataset.

Method	Accuracy
MVCNN	89.9%
MVCNN + attention weight	92.35%
Visual Saliency model	91.44%
Ours	**93.02%**

is demonstrated by the related experimental results, which are listed in the first row and second row of Tab. 1. "MVCNN" denotes the typical view based method proposed by Su *et al.* [1]. And "MVCNN + attention weight" is a modified version of MVCNN, which replaces the view pooling part of the typical MVCNN method with the view attention pooling part. Obviously, the classification result of typical MVCNN model has the lower accuracy comparing to MVCNN model with the attention weights. The result demonstrates our attention weight can make the model focus on more representative views to obtain a better performance on the 3D model recognition. In our proposed method, we introduce the Visual Saliency model, including two LSTM layers and a soft-attention network, to consider the structure and correlation information from the multiple extracted views and guide the visual features pooling in the MVCNN branch. As Fig.1 shown, from Visual Saliency model, the last hidden state of the second LSTM layer are used as the Visual Saliency Feature for the classification task and obtaining an accuracy of 91.12%. Compare to the typical MVCNN model, Visual Saliency model wins by 1.22% more gains in terms of classification accuracy, which means the design of seeming the extracted views as a view sequence and exploiting their structure information is reasonable and feasible for 3D model recognition. Our method, which both retain the correlated information and visual information from multiple extracted views, outperforms the other parts of our network, bringing a 2.79%, 0.96%, 1.57% improvement over each part of our network respectively. From the above experiment, the experimental results demonstrate our proposed network architecture is effective to obtain a better 3D model representation.

E. COMPARISON WITH STATE-OF-THE-ART METHODS ON THE MODELNET40 DATASET

In order to validate the superiority of the proposed network, 3D model classification and retrieval experiments have been conducted on the ModelNet40 dataset. As for the dataset, we follow the same training and test split setting in [2]. In the Tab. 2, "Ours(GoogLeNet)" means we utilize the GoogLeNet to extract the features of views, which are used as the input of our two-stream network. In experiments, we have compared our two-stream model with various models based on different representations, including volumetric based models (3D modelNets by Wu *et al.* [2]), hand-craft descriptors for multi-view data (SPH by Kazhdan *et al.* [42] and LFD by Chen *et al.* [43]), deep learning models for multi-view

TABLE 2. Comparisons of classification and retrieval experimental results on the ModelNet40 dataset.

Method	Train Config		Data Representation	Classification	Retrieval
	Pre train	Fine tune	#Number of Views	(Overall Accuracy)	(mAP)
(1)SPH[42]	-	-	-	68.2%	33.3%
(2)LFD[43]	-	-	-	75.5%	40.9%
(3)3D modelNets[2]	ModelNet40	ModelNet40	Volumetric	77.3%	49.2%
(4)VoxNet[4]	ModelNet40	ModelNet40	Volumetric	83.0%	-
(5)VRN[44]	ModelNet40	ModelNet40	Volumetric	91.3%	-
(6)MVCNN-MultiRes[7]	-	ModelNet40	Volumetric	91.4%	-
(7)MVCNN,12×[1]	ImageNet1K	ModelNet40	12 Views	89.9%	70.1%
(8)MVCNN,metric,12×[1]	ImageNet1K	ModelNet40	12 Views	89.5%	80.2%
(9)MVCNN,80×[1]	ImageNet1K	ModelNet40	80 Views	90.1%	70.4%
(10)MVCNN,metric,80×[1]	ImageNet1K	ModelNet40	80 Views	90.1%	79.5%
(11)MVCNN(GoogLeNet),12×	ImageNet1K	ModelNet40	12 Views	92.2%	74.1%
(12)MVCNN(GoogLeNet),metric,12×	ImageNet1K	ModelNet40	12 Views	92.2%	83.0%
(13)PointNet[3]	-	ModelNet40	Point Cloud	89.2%	-
(14)PointNet++[12]	-	ModelNet40	Point Cloud	90.7%	-
(15)KD-Network[17]	-	ModelNet40	Point Cloud	91.8%	-
(16)PointCNN[45]	-	ModelNet40	Point Cloud	91.8%	-
(17)DGCNN[19]	-	ModelNet40	Point Cloud	92.2%	-
(18)PANORAMA-NN[46]	-	ModelNet40	PANORAMA-Views	90.7%	83.4%
(19)CNN-BiLSTM[30]	-	ModelNet40	12 views	-	83.3%
(20)VNN[31]	-	ModelNet40	12 views	-	**89.3%**
(21)MHBN[47]	-	ModelNet40	12 views (6RGB + 6Dep)	93.1%	-
(22)TCL[48]	-	ModelNet40	12 views	-	88%
(23)RED[49]	-	ModelNet40	-	-	86.3%
(24)LMRL[50]	-	ModelNet40	12 views	91.1%	84.3%
(25)Ours	-	ModelNet40	12 Views	93.0%	85.3%
(26)Ours(GoogLeNet)	-	ModelNet40	12 Views	**93.4%**	87.2%

data (MVCNN by Su *et al.* [1], MVCNN-MultiRes by Qi *et al.* [7], CNN-BiLSTM by Dai *et al.* [30], VNN by He *et al.* [31], MHBN by Yu *et al.* [47], TCL by He *et al.* [48], RED by Bai *et al.* [49] and LMRL by Ma *et al.* [50]), point cloud based models (PointNet by Qi *et al.* [3], PointNet++ by Qi *et al.* [12], Kd-Network by Klokov *et al.* [17], PointCNN by Li *et al.* [45] and DGCNN by Wang *et al.* [19]) and panorama views based model(PANORAMA-NN by Sfikas *et al.* [46]).

From the experimental results in Tab. 2, we can see "Ours(GoogLeNet)" outperforms all the comparison methods in terms of classification accuracy on the ModelNet40 dataset. Note that, the classification results for CNN-BiLSTM [30], VNN [31] and RED [49] are not provided in their papers. MHBN achieves the second best classification performance, it not only uses the information of extracted views but also utilizes the depth images of each model for boosting the performance, resulting in more complicated network architecture. When compared to "MVCNN(GoogLeNet)", two-stream network outperforms it by 1.2% on the classification task, which can be attributed to the exploitation of the correlated information among the views. For the 3D model retrieval task, we employ the Euclidean distance to rank the reference models in our method and the above referenced methods are set as the comparison to our network. As Tab. 2 presented, our method outperforms MVCNN, which is the baseline of our network, and achieves increments of 7% in mAP on the ModelNet40 dataset. This result shows the correlation among the views can contribute to obtaining the discriminative shape descriptor. However, our retrieval performance is not the best compare to the recent view-based 3D model retrieval methods, such as VNN [31] and TCL [48]. VNN achieves the best retrieval result for a mAP of 89.3%, it captures the local spatial information across the extracted views by dividing the view sequence into a set of visual n-grams. Moreover, VNN introduces an attentional feature aggregation module, which allows the network to focus on the more discriminate view features. TCL [48] achieves the second best retrieval result with a mAP of 88%, which demonstrates TCL can effectively further enhance the discriminate power of the features. For each class of 3D models, TCL can learn a center

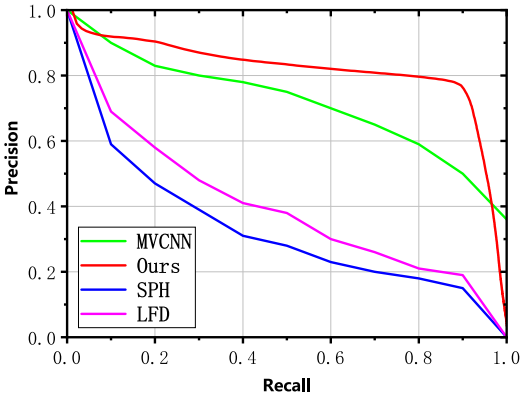


FIGURE 3. Precision-recall curves for our network and other methods on the ModelNet40 dataset.

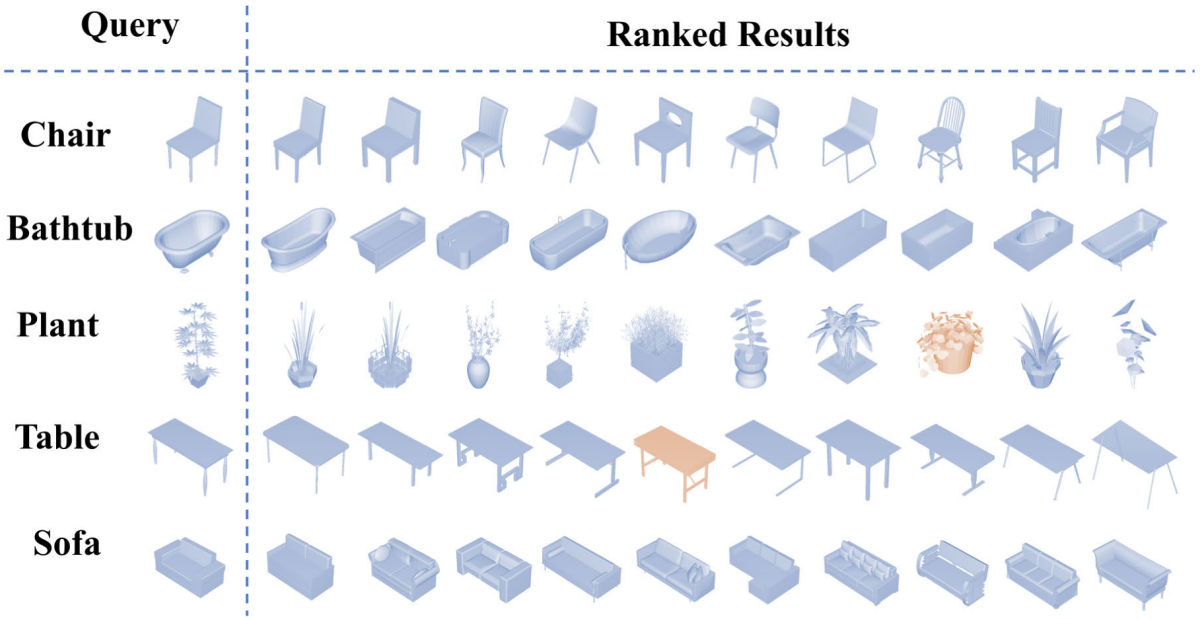


FIGURE 4. Illustration of the 3D model retrieval results on the ModelNet40 dataset. The retrieved top 10 models are selected according to the pairwise similarity. From left to right, the query 3D models are listed on the first left column, and the ranked results are listed on the right side. The blue and orange color indicate the correc and incorrect retrieval models, respectively.

TABLE 3. Retrieval accuracy measured via mAP, F-score, and NDCG on ShapeNetCore55 data set.

Method	Micro-averaged			Macro-averaged		
	F-score	mAP	NDCG	F-score	mAP	NDCG
RotationNet	0.798	0.722	0.865	0.819	-	-
Improved GIFT	0.767	0.722	0.827	0.581	0.575	0.657
ReVGG	0.772	0.749	0.828	0.519	0.496	0.559
DLAN	0.712	0.663	0.762	0.505	0.477	0.563
SHREC16-Bai_GIFT	0.689	0.640	0.765	0.454	0.447	0.548
SHREC16-Su_MVCNN	0.764	0.735	0.815	0.575	0.566	0.640
Ours	0.773	**0.864**	**0.872**	0.589	**0.742**	**0.859**

and require that the distances between samples and centers from the same class are closer than those from different classes to further boost the retrieval performance. Our method with GoogLeNet achieves the third best retrieval performance among these comparison methods for a mAP of 87.2%. Other than this, from Tab. 2, bidirectional LSTM is employed in CNN-BiLSTM [30] and LMRL [50] to aggregate the multiple view features and obtains better performance than typical pooling methods. RED [49] also achieves competitive performance on the ModelNet40 dataset, it introduces an automatic weight learning paradigm to surpass the negative impacts of noisy similarities.

As Fig.3 presented, the precision-recall curves we achieved on the ModelNet40 dataset demonstrates the effectiveness of our method. Obviously, when the retrieval recall is under 0.9, the highest retrieval precision and the best overall retrieval performance among all the methods have achieved by the proposed network. These results have demonstrated the promising discriminative capacity of our method for 3D model retrieval task. As Fig.4 presented, there are several retrieval examples on the ModelNet40 dataset and we can see that the highly relevant 3D models are retrieved for the query models by our method.

TABLE 4. Performance(%) on ModelNet40 with different view numbers.

View	NN	FT	ST	F_measure	DCG	ANMRR	ACC
2	88.57	77.21	87.26	32.40	80.92	19.73	89.34
4	90.64	82.66	92.29	33.57	85.74	14.66	91.57
6	91.41	83.40	91.99	33.69	86.33	13.94	91.93
8	89.99	81.97	90.19	33.45	85.04	15.27	90.31
10	90.56	81.68	89.50	33.60	85.02	15.45	91.57
12	90.72	80.77	89.81	33.47	84.27	16.24	**92.45**
20	90.48	82.63	91.22	33.55	85.63	14.72	91.04

F. RETRIEVAL RESULTS ON THE SHAPENETCORE55 DATASET

On the ShapeNetCore55 dataset, there are two kinds of versions of the above evaluation metrics to be used in the experiment. The macro-averaged version is used to provide the unweighted average of the entire dataset and the

TABLE 5. Performance (%) with different view orders on the ModelNet40 dataset.

View Order	NN	FT	ST	F_measure	DCG	ANMRR	ACC
Disordered	91.70 ± 0.32	83.03 ± 1.29	90.97 ± 0.97	33.79 ± 0.16	86.19 ± 1.03	14.21 ± 1.13	93.02 ± 0.2
Ordered	90.72	80.77	89.81	33.47	84.27	16.24	92.45

micro-averaged version is used to adjust the size of the model categories to provide a representative performance metric across categories. In addition, on the official website of the SHREC competition, the evaluation code for all these indicator calculations are provided by the organizer.

The retrieval experimental results on the ShapeNetCore dataset, including the pose normalized and perturbed versions, are provided in Table.3. We compare our method with other methods, which have demonstrated their superiority and exhibited higher performance in the ShapeNetCore55 tracks. According to listed results in Table3, for the macro-averaged version and micro-averaged version metrics, the proposed method outperforms other methods by a large margin, except for the RotationNet. RotationNet achieves the highest F-score on the both Micro-averaged version and macro-averaged version metrics, it utilizes a partial set of the extract view images to jointly estimate the pose and category of the model. Although it achieves higher score in the term of F-score compared to the proposed method, it ignores the correlated information among the extracted views, which has been taken into consideration in our method. On the other two evaluation metrics, mAP and NDCG, we achieve the best performance, which further demonstrates the superiority of the proposed method.

G. SENSITIVITY ANALYSIS ON VIEW NUMBER

Due to the number of extracted view images may have direct influence on the 3D model recognition performance, we performed comparative experiment to select the best number of the view images. Concretely, a virtual camera array is set up around the z-axis with the intervals of angle θ. The θ is set to $\{180°, 90°, 60°, 45°, 36°, 30°, 18°\}$ respectively, which means there are $\{2, 4, 6, 8, 10, 12, 20\}$ view images are generated for each model.

The experimental results are presented in Table.4. As the number of view images increasing, the performance of our network keeps improving until the view number increase to 12. At the beginning, there are only 2 view images are used as the input of our network, which do not provide enough information for the proposed network to effectively learn the representation of the 3D model. So before the optimal number coming, the performance of proposed network can be improved by increasing the number of view images. As shown in Table 4 and Fig.5, when the number of views increased over 12, excessive views images produce redundant information and lead to worse performance. Compared with other number of view images, when view number is set to 12, the performance is the best so we select the optimal number as 12.

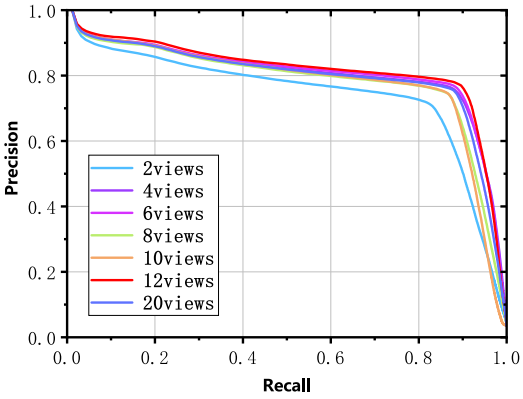


FIGURE 5. Precision-recall curves of the experiment on the different view numbers, which is conducted on the ModelNet40 dataset.

H. SENSITIVITY ANALYSIS ON VIEW ORDER

Intuitively, the order of input views can directly influence the effect of 3D model feature learning. To demonstrate the robustness of our network, we conduct the experiment on the ModelNet40 dataset by upsetting the view order number of multi-view sequence 50 times in the testing procedure. The related retrieval and classification results are presented in Table 5. We can observe that the result of disordered input views is even better than ordered input views method's result. The reason of this consequence can be summarized to the parameter $\mathbf{w}^\top$ from the Equation 6 will learn more structure information from the taken views when the input views is disordered. Obviously, according to these results, our network has been demonstrated it can adaptively calculate the importance of individual views, and with these information, the structure and visual information from multiple taken views are well exploited. So, for the 3D model representation learning, proposed method can be free of the camera setting and achieve robust performance.

V. CONCLUSION

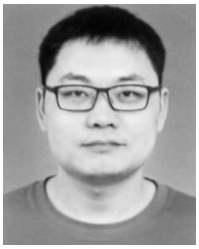
In this paper, we have presented a novel two-stream network based on visual differentiation information of multiple views, consisting of a MVCNN model and a Visual Saliency model. The MVCNN model extracts the view information from the taken views, and the Visual Saliency model defines the weight of views based on the similarity and differentiation information of multiple views. Especially, the weight of views defined by the Visual Saliency model can effectively be used to guide the visual information fusion in MVCNN model. Here, the correlation information among the views for each model is taken into consideration in the learning step. Finally, the early-fusion method is employed to fuse

the feature vectors from MVCNN model and Visual Saliency model respectively, in order to generate the shape descriptor for 3D model recognition and retrieval. Compared with current effective methods, our method not only utilizes the visual information from the taken views, but also takes the correlation information into consideration for 3D model recognition. Experimental results on the two public model dataset have demonstrated that the effectiveness of the proposed network, which means the correlation information is crucial for view-based 3D model recognition methods. Related experimental results also showed that the proposed method can achieve a robust and discriminative representation for the 3D model.

REFERENCES

[1] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 945–953.
[2] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1912–1920.
[3] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 652–660.
[4] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.
[5] Y. Li, S. Pirk, H. Su, C. R. Qi, and L. J. Guibas, "FPNN: Field probing neural networks for 3D data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 307–315.
[6] N. Sedaghat, M. Zolfaghari, E. Amiri, and T. Brox, "Orientation-boosted voxel nets for 3D object recognition," 2016, *arXiv:1604.03351*. [Online]. Available: https://arxiv.org/abs/1604.03351
[7] C. R. Qi, H. Su, M. Niebner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and Multi-view CNNs for Object Classification on 3D Data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5648–5656.
[8] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein, "Learning shape correspondence with anisotropic convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3189–3197.
[9] D. Boscaini, J. Masci, E. Rodolà, M. M. Bronstein, and D. Cremers, "Anisotropic diffusion descriptors," *Comput. Graph. Forum*, vol. 35, no. 2, pp. 431–441, 2016.
[10] J. Xie, G. Dai, F. Zhu, E. K. Wong, and Y. Fang, "DeepShape: Deep–learned shape descriptor for 3D shape retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1335–1345, Jul. 2017.
[11] F. Zhu, J. Xie, and Y. Fang, "Learning cross-domain neural networks for sketch-based 3D shape retrieval," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016.
[12] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.
[13] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. CVPR*, 2005, pp. 539–546.
[14] Z. Han, Z. Liu, J. Han, C.-M. Vong, S. Bu, and C. L. P. Chen, "Mesh convolutional restricted Boltzmann machines for unsupervised learning of features with structure preservation on 3-D meshes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2268–2281, Oct. 2017.
[15] Y. Feng, Y. Feng, H. You, X. Zhao, and Y. Gao, "MeshNet: Mesh neural network for 3D shape representation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8279–8286.
[16] T. Furuya and R. Ohbuchi, "Deep aggregation of local 3D geometric features for 3D model retrieval," in *Proc. BMVC*, 2016, pp. 1–121.
[17] R. Klokov and V. Lempitsky, "Escape from cells: Deep KD-networks for the recognition of 3D point cloud models," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 863–872.
[18] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-CNN: Octree-based convolutional neural networks for 3D shape analysis," *ACM Trans. Graph.*, vol. 36, no. 4, p. 72, 2017.
[19] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, p. 146, 2019.
[20] S. Xie, S. Liu, Z. Chen, and Z. Tu, "Attentional shapecontextnet for point cloud recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4606–4615.
[21] A.-A. Liu, W.-Z. Nie, Y. Gao, and Y.-T. Su, "Multi–modal clique–graph matching for view–based 3D model retrieval," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2103–2116, May 2016.
[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556
[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[25] K. Sfikas, I. Pratikakis, and T. Theoharis, "Ensemble of PANORAMA-based convolutional neural networks for 3D model classification and retrieval," *Comput. Graph.*, vol. 71, pp. 208–218, Apr. 2018.
[26] Z. Zhang, H. Lin, X. Zhao, R. Ji, and Y. Gao, "Inductive multi–hypergraph learning and its application on view–based 3D object classification," *IEEE Trans. Image Process.*, vol. 27, no. 12, pp. 5957–5968, Dec. 2018.
[27] S. Bai, X. Bai, Z. Zhou, Z. Zhang, and L. Jan Latecki, "Gift: A real-time and scalable 3D shape search engine," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5023–5032.
[28] A. Kanezaki, Y. Matsushita, and Y. Nishida, "RotationNet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5010–5019.
[29] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao, "GVCNN: Group-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 264–272.
[30] G. Dai, J. Xie, and Y. Fang, "Siamese CNN-BiLSTM architecture for 3D shape representation learning," in *Proc. IJCAI*, 2018, pp. 670–676.
[31] X. He, T. Huang, S. Bai, and X. Bai, "View N-gram network for 3D object retrieval," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 7515–7524.
[32] P. Papadakis, I. Pratikakis, S. Perantonis, and T. Theoharis, "Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation," *Pattern Recognit.*, vol. 40, no. 9, pp. 2437–2452, 2007.
[33] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. NIPS*, 2014.
[34] A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," 2016, *arXiv:1605.07678*. [Online]. Available: https://arxiv.org/abs/1605.07678
[35] Y. Yang, J. Zhou, J. Ai, Y. Bin, A. Hanjalic, H. T. Shen, and Y. Ji, "Video captioning by adversarial LSTM," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5600–5611, Jan. 2018.
[36] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting human eye fixations via an LSTM-based saliency attentive model," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5142–5154, Oct. 2016.
[37] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018.
[38] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.
[39] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2014, *arXiv:1409.2329*. [Online]. Available: https://arxiv.org/abs/1409.2329
[40] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[41] A.-A. Liu, W.-Z. Nie, Y. Gao, and Y.-T. Su, "View-based 3-D model retrieval: A benchmark," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 916–928, Mar. 2018.
[42] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Proc. Symp. Geometry Process.*, vol. 6, 2003, pp. 156–164.

[43] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," *Comput. Graph. Forum*, vol. 22, no. 3, pp. 223–232, Sep. 2003.
[44] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "Generative and discriminative voxel modeling with convolutional neural networks," 2016, *arXiv:1608.04236*. [Online]. Available: https://arxiv.org/abs/1608.04236
[45] Y. Li, R. Bu, M. Sun, and B. Chen, "PointCNN: Convolution On $\mathcal{X}$-transformed points," 2018, *arXiv:1801.07791*. [Online]. Available: https://arxiv.org/abs/1801.07791
[46] K. Sfikas, T. Theoharis, and I. Pratikakis, "Exploiting the panorama representation for convolutional neural network classification and retrieval," *3DOR*, vol. 6, p. 7, Apr. 2017.
[47] T. Yu, J. Meng, and J. Yuan, "Multi-view harmonized bilinear network for 3D object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 186–194.
[48] X. He, Y. Zhou, Z. Zhou, S. Bai, and X. Bai, "Triplet-center loss for multi-view 3D object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1945–1954.
[49] S. Bai, Z. Zhou, J. Wang, X. Bai, L. Jan Latecki, and Q. Tian, "Ensemble diffusion for retrieval," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 774–783.
[50] C. Ma, Y. Guo, J. Yang, and W. An, "Learning multi–view representation with LSTM for 3-D shape recognition and retrieval," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1169–1182, May 2019.



WEIZHI NIE received the Ph.D. degree from Tianjin University, Tianjin, China. He was a Visiting Scholar with the NExT Center, National University of Singapore, under the supervision of Prof. T.-S. Chua. He is currently an Assistant Professor with the School of Electrical and Information Engineering, Tianjin University. His research interests include computer vision, machine learning, and social networks.

LU QU is currently pursuing the M.S. degree with the School of Electrical and Information Engineering, Tianjin University. Her research interests include computer vision and 3D model retrieval.

MINJIE REN is currently pursuing the M.S. degree with the School of Electrical and Information Engineering, Tianjin University. Her research interests include 3D shape recognition and cross domain learning.

QI LIANG received the M.S. degree from the School of Electrical and Information Engineering, Tianjin University. His research interests include 3D shape recognition, cross domain learning, and Data mining.

YUTING SU received the M.S. and Ph.D. degrees in electronic engineering from Tianjin University, China. He is currently a Professor with the School of Electrical and Information Engineering, Tianjin University. His research interests include multimedia content analysis and security, multiple object tracking, and multimedia content analysis and security.

YANGYANG LI received the B.S. degree from the Nanjing University of Information and Technology, in 2009, and the Ph.D. degree in computer science from the Beijing University of Posts and Telecommunications, in 2015. He is currently a Senior Engineer at the National Engineering Laboratory for Public Safety Risk Perception and Control by Big Data (PSRPC). His research interests include the mobile internet, content security, and edge computing.

HAO JIN received the B.E. degree from the Hefei University of Technology, in 2013, and the Ph.D. degree in communication and information system from the Institute of Information Engineering, Chinese Academy of Sciences, in 2018. She is currently an Engineer with the National Engineering Laboratory for Public Safety Risk Perception and Control by Big Data (PSRPC). Her research interests include the mobile internet, content security, and mobile malware detection.