Displays 69 (2021) 102053

Contents lists available at ScienceDirect

Displays

journal homepage: www.elsevier.com/locate/displa

Review of multi-view 3D object recognition methods based on deep learning

Shaohua Qi [a,b], Xin Ning [a,c,e,*], Guowei Yang [b], Liping Zhang [a,c], Peng Long [a], Weiwei Cai [d], Weijun Li [a,e]

[a] *Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China*
[b] *College of Electronic Information, Qingdao University, Qingdao 266071, China*
[c] *Cognitive Computing Technology Joint Laboratory, Wave Group, Beijing 102208, China*
[d] *School of Logistics and Transportation, Central South University of Forestry and Technology, Changsha 410004, China*
[e] *Center of Materials Science and Optoelectronics Engineering & School of Microelectronics, University of Chinese Academy of Sciences, Beijing 100049, China*

ARTICLE INFO

Keywords:
Multi-view
Deep Learning
3D Object Classification
3D Object Retrieval

ABSTRACT

Three-dimensional (3D) object recognition is widely used in automated driving, medical image analysis, virtual/augmented reality, artificial intelligence robots, and other areas. Deep learning is increasingly being used to solve 3D vision problems. Multi-view 3D object recognition based on the deep learning technique has become one of the rigorously researched topics because it can directly use the pretrained and successful advanced classification network as the backbone network, and views from multiple viewpoints can complement each other's detailed features of the object. However, some challenges still exist in this area. Recently, many methods have been proposed to solve the problems pertaining to this research topic. This paper presents a comprehensive review and classification of the latest developments in the deep learning methods for multi-view 3D object recognition. It also summarizes the results of these methods on a few mainstream datasets, provides an insightful summary, and puts forward enlightening future research directions.

1. Introduction

With the rapid developments in automated driving [1], medical image analysis, virtual reality/augmented reality [2,3], artificial intelligence robots, and other areas, the demand for object recognition in various environments has become urgent and important. As a result, the area of object recognition has developed rapidly. With the increase and maturity of the three-dimensional (3D) acquisition technology [4] and 3D object datasets, 3D object recognition has become one of the important and popular research directions in the area of object recognition. Deep learning is a new field of machine learning [5–10]. Object recognition has entered a new stage because of the development of convolutional neural networks (CNNs) [11]. Deep learning and CNNs have been widely used in 3D object classification and have achieved excellent performance.3D object recognition methods based on deep learning can be divided into three research directions depending on their input modes [12], namely, voxel-based methods [13–22], point-set-based methods [23–29], and view-based methods [30–48,12,49–55]. An object is represented as a 3D mesh in the voxel-based methods and is analyzed by a 3D network. In the point-set-based methods, an object is represented as a set of unordered points, and the point cloud is used for prediction [12]. These two methods can also be collectively referred to as model-based methods [41]. The model-based methods use a 3D convolution filter to convolute a 3D shape in 3D space, thus generating a 3D representation directly from the 3D data [34,56]. Although they utilize the spatial information of 3D objects, their practical applications are limited by the computational cost [34]. The view-based methods render 3D objects to 2D images from different viewpoints and convolute these views using a 2D convolution filter. The view-based methods do not rely on the complex 3D features, and it is easy to capture the input view in these methods. They have a large amount of data and can make use of a mature advanced network framework. In the case of occlusion, the views from different viewpoints can complement each other's detail features of the object and achieve excellent recognition performance.

In the past six years since the introduction of the multi-view convolutional neural network (MVCNN) [31], pioneering work in the area of 3D object recognition methods based on the multi-view technique has

* Corresponding author.
E-mail address: ningxin@semi.ac.cn (X. Ning).

https://doi.org/10.1016/j.displa.2021.102053
Received 13 April 2021; Received in revised form 15 June 2021; Accepted 5 July 2021
Available online 14 July 2021

developed rapidly. In contrast to the existing reviews [57,58], our review focuses on deep learning methods for multi-view 3D object recognition, mainly including the latest studies from 2018 to 2021. We have innovatively divided this review into (i) viewpoints setting and input modes, (ii) backbone network and feature extraction, and (iii) feature fusion and viewpoints selection mechanisms, as shown in Fig.1. Among them, the biggest difference compared to the two-dimensional (2D) image classification is the viewpoints setting and input modes, and the feature fusion and the viewpoint selection mechanisms. 2D image classification usually does not require consideration of these two points and processes each image in the dataset separately. In 3D object recognition based on multi-view, the features of an object are captured from multiple perspectives and these are further fused to improve the recognition accuracy. Thus, feature fusion has become the research focus in this area. This paper focuses on viewpoints setting and feature fusion of the existing methods and compares their performance by applying them to the mainstream datasets.

Compared to the existing literature, the major contributions of this article can be summarized as follows:

- To the best of our knowledge, this is the first review that comprehensively covers the detailed analysis of each module of multi-view 3D object recognition using deep learning methods, including viewpoints setting and input modes, backbone network and feature extraction, and feature fusion and viewpoints selection mechanism.
- In contrast to the existing reviews [57,58], this paper focuses on the deep learning methods of 3D object recognition based on multi-view for the first time, excluding other 3D object recognition methods or 3D data representation.
- This paper introduces the latest developments in the deep learning methods corresponding to the multi-view 3D object recognition and provides a summary of the state-of-the-art methods for readers.
- This paper compares many methods, and summarizes the performance of most methods on several mainstream datasets (e.g., Tables 2, 3).

In this paper, we provide a review of the deep learning methods for 3D object recognition based on multi-view, which is organized as follows: Section 2 introduces the mainstream datasets and evaluation metrics. Section 3 provides a survey of the viewpoints setting and input modes used in the existing multi-view 3D object recognition methods. Section 4 presents a review of the 3D object recognition methods based on multi-view according to their feature extraction and fusion strategies. Section 5 summarizes the performance of most methods on several mainstream datasets. Section 6 concludes the paper by presenting the characteristics of 3D object recognition based on multi-view and gives a brief discussion on the future research directions in this topic.

2. Background

This section introduces the mainstream datasets and the evaluation metrics used in the various studies that have been performed in the last six years on 3D object recognition based on multi-view.

2.1. Datasets

Table 1 lists the main datasets that have been used in different studies on multi-view 3D object recognition.

2.1.1. ModelNet

ModelNet [13] is a collection of 3D object computer-aided design (CAD) models from the Princeton ModelNet project. The dataset contains 660 object categories and 151128 models. The 40 staff selected and 10 common categories from the dataset form the core data subset and are represented as ModelNet40 [13] and ModelNet10 [13], respectively, and both datasets have the version of orientation aligned. ModelNet40 [13] consists of 40 categories such as airplane, chair, cup, and guitar without color information, totaling up to 12311 models. The models are divided into 9843 training data and 2468 test data. ModelNet10 [13] consists of 4899 models divided into 10 categories, which are further classified into 3991 training data and 908 test data.

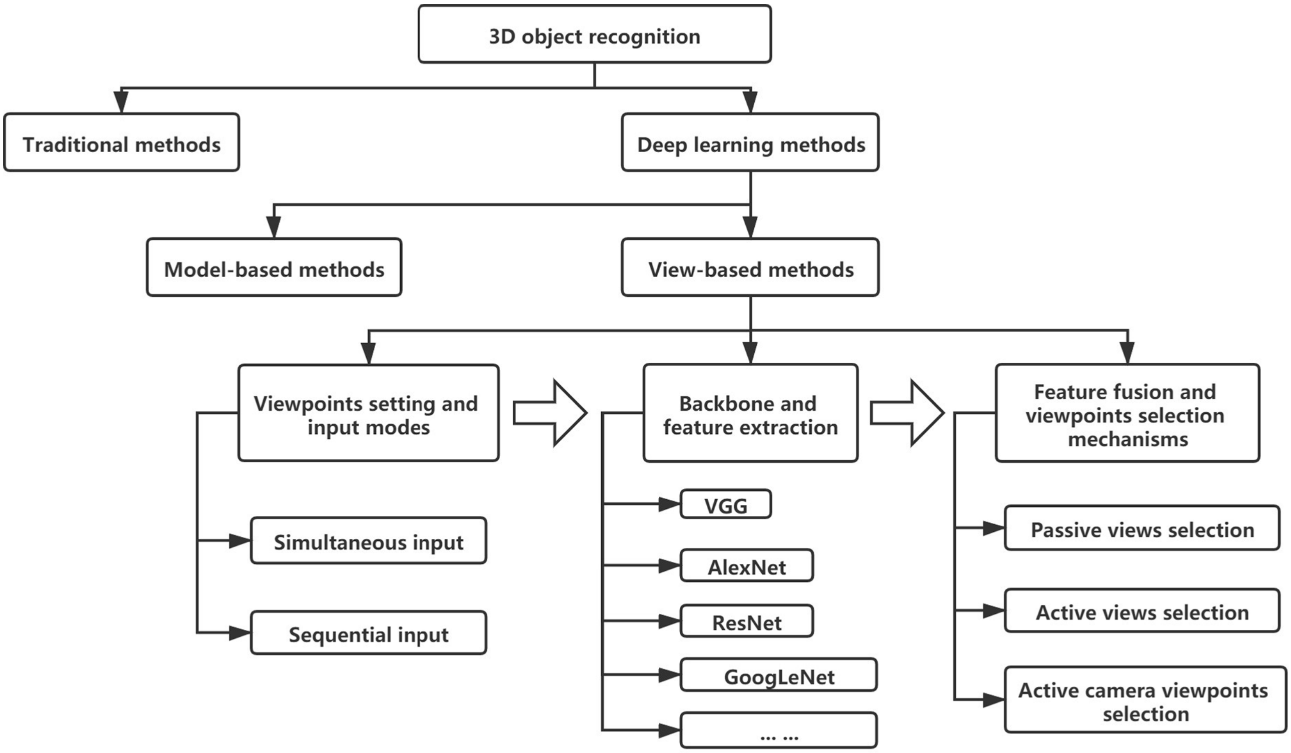

Fig. 1. Various 3D object recognition methods based on the multi-view approach.

Table 1
Datasets used in studies on multi-view 3D object recognition.

Datasets for 3D Shape Classification							
Name	**Year**	**Samples**	**Classes**	**Training**	**validation**	**Test**	**Type**
RGB-D [59]	2011	250000	51	-	-	-	Real-World
ModelNet40 [13]	2015	12311	40	9843	-	2468	Synthetic
ModelNet10 [13]	2015	4899	10	3991	-	908	Synthetic
ShapeNetCore [60]	2015	51162	55	35764	5133	10265	Synthetic
MIRO [30]	2018	160	12	-	-	-	Real-World

2.1.2. *ShapeNet*

The ShapeNet [60] dataset is a large-scale 3D image dataset, including ShapeNetCore [60] and ShapeNetSem [60]. ShapeNetCore [60] is mainly used in general articles. It includes 55 common object categories and 51162 clean models. The training set, validation set, and test set are composed of 35764, 5133, and 10265 shapes, respectively.

2.1.3. *RGB-D*

The RGB-D object dataset [59] consists of 51 categories, 300 common family objects, and 250000 RGB-D images. The dataset has been recorded by placing each object on a turntable and using the Kinect-style 3D cameras of different heights. There are two problems in training a CNN based on multi-view using RGB-D dataset [59,30]: there are only a few object instances per category, and the assumption of the vertical direction is inconsistent. In some cases, the vertical direction assumption is invalid. In some categories, the object instances are not posed consistently relative to the axis of rotation. In addition, this dataset does not include the bottom portion of the object placed on the turntable [30].

2.1.4. *MIRO*

Multi-view images of rotated objects (MIRO) [30] is a self-made dataset of RotationNet [30]. Kanezaki et al. captured multi-view images of rotating objects and obtained 160 images. The dataset consists of 120 object instances of 12 categories, and each object category has 10 object instances. This dataset has only been used by RotationNet [30] and has not become a mainstream dataset.

2.2. *Evaluation metrics*

With the increasing number and dimensions of visual content, it is an important task for many applications to classify and retrieve visual content quickly [61]. Generally, two types of evaluation metrics are used for 3D object recognition methods based on multi-view, classification task, and retrieval task. 3D object retrieval involves finding the specified query object in the dataset, whereas 3D object classification involves identifying the category of a given 3D object [62].

In the task of classification, two metrics that can be used as the criteria for object recognition accuracy are *instance accuracy* and *classification accuracy*. *Instance accuracy* is the ratio of the number of correctly classified objects to the total number of objects, whereas *classification accuracy* is the average of the *instance accuracy* in all categories. The *classification accuracy* is objective because the number of test objects of different categories is uneven, whereas *instance accuracy* is intuitive because it reflects the number of objects that are wrongly classified [44]. The two accuracies can be expressed as shown in Eqs.(1) and (2), where TP_i, TN_i, P_i, N_i are the number of samples of true positive, true negative, positive, and negative outcomes corresponding to the *i*th category respectively, and C is the total number of categories in the sample.

$$instanceaccuracy = \frac{\sum_{i=1}^{C} TP_i + TN_i}{\sum_{i=1}^{C} P_i + N_i} \tag{1}$$

$$classificationaccuracy = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i + TN_i}{P_i + N_i} \tag{2}$$

The retrieval task generally uses the mean average precision (*mAP*) as the evaluation standard [31]. Average precision (*AP*) is the area under the *PR* curve, where *P* is the precision, and *R* is the recall value. *mAP* is the average value of *AP* in each category. The expressions for *P*, *AP*, and *mAP* are given in Eqs.(3), (4), and (5), respectively, where *TP* and *FP* represent true positive and false positive outcomes respectively, *N* represents the total number of samples in the category, *P(k)* represents the precision value on each image in the category, Δ*r(k)* represents the change in the recall value from *k*-1 to *k*, and *C* is the total number of categories in the sample.

$$P = \frac{TP}{TP + FP} \tag{3}$$

$$AP = \sum_{k=1}^{N} P(k)\Delta r(k) \tag{4}$$

$$mAP = \frac{\sum_{k=1}^{C} AP(k)}{C} \tag{5}$$

3. Viewpoints setting and views input modes

3D object recognition based on multi-view reduces the complex 3D classification tasks to simple 2D classification tasks, and thus the 3D objects need to be rendered into 2D images first. This step of viewpoints setting is performed to obtain 2D rendered images with different perspectives so that the features captured from the different views can be complementary and interrelated and can help to improve the recognition performance of the network. In existing studies, the camera viewpoints were generally set up at different positions in advance, and rendering of 3D objects to 2D images was performed from these viewpoints. The images thus obtained were then inputted into the network, and convolution of all views from the different viewpoints was done using a 2D convolution filter. Depending on the view inputs in the existing studies, we can divide them into simultaneous input and sequential input.

3.1. *Simultaneous input*

MVCNN, proposed by Su et al. [31], is a pioneering work in the field of 3D object recognition based on multi-view in which two camera settings were tested. In the first camera setup, the object was assumed to be oriented vertically along a consistent axis, such as the Z-axis. By setting a virtual camera at 30° intervals around the object, 12 rendering views could be collected (Fig.2). The camera was raised 30° from the ground level and pointed at the center of mass of the object. The second camera setup did not take advantage of the assumption that the object is in the same vertical direction. In this case, since the viewpoints that can produce a good representative view of the object was not a priori known, they placed 20 virtual cameras at the 20 vertices of the icosahedron (we think that Su et al. wrote it wrongly in their paper, it should be

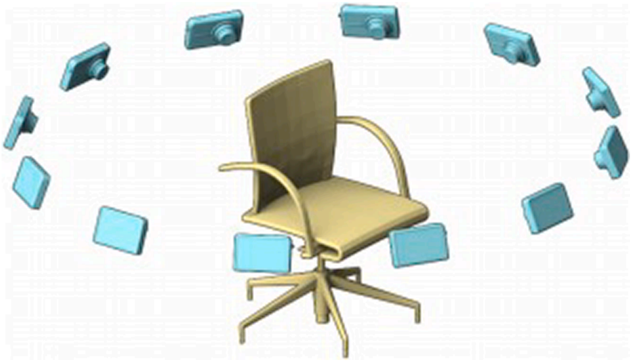

Fig. 2. The first viewpoints setting of MVCNN [31].

dodecahedron as only a dodecahedron has 20 vertices) surrounding the object for rendering. All cameras pointed to the center of mass of the object. Then, each camera was rotated by 0°, 90°, 180°, and 270° along the axis passing through the centroid of the camera and the object respectively to generate four rendering views, providing a total of 80 views. Simultaneous input was used in training the network; any number of views could be inputted from all cameras into the network at the same time.

As an improvement to MVCNN [31], Feng et al. [36] proposed the group-view convolutional neural network (GVCNN). They designed two predefined camera arrays with regular views. The first was a 45° horizontal circle that consisted of 8 cameras and generated 8 views, whereas the second was a 30° horizontal circle that consisted of 12 cameras and generated 12 views. Its views input mode was simultaneous input. Although GVCNN [36] improves the performance of MVCNN [31], its viewpoints setting is a distribution only along the horizontal circle, lacking many views at other angles of the 3D target, and this limits its performance.

The perspective of GIFT proposed by Bai et al. [54] is more comprehensive. In their work, the center of mass of each 3D object was placed at the origin of the spherical coordinate system, and the maximum polar distance of the points on the object surface was adjusted to unit length. The angular range [0, 2π] was divided into eight parts in the horizontal direction, whereas [0, π] was divided into eight parts in the vertical direction.

All aforementioned methods are regular view configurations based on the uniform space assumption [33]. Wei et al. proposed the view-based graph convolutional neural network (View-GCN) [33], the viewpoints setting of which are shown in Fig. 3, where the configurations shown in Figs.3(a) and 3(b) are consistent with the MVCNN [31] viewpoints setting. With different settings shown in Fig.3(c), the viewpoints can be flexibly extended to an irregular-view configuration based on 12 views randomly perturbed in coordinates that are randomly selected from those shown in Fig.3(b). Its views input mode is also given as an input at the same time. Wei et al. found that the per-class and per-instance accuracies of the model having an irregular view configuration were 4.2% and 1.9% higher, respectively than those reported by Su et al. [49]. However, in the case of a randomly selected irregular view configuration, the positions of the viewpoints might be similar, and the resulting views may or may not be identical. This will lead to some views making little contribution to the network and some viewpoints becoming invalid.

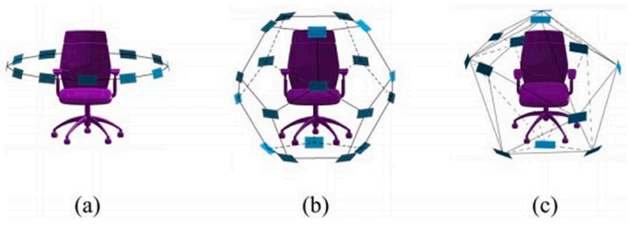

Fig. 3. Different view configurations in View-GCN [33].

3.2. Sequential input

In order to solve the problem of the relationship between the views in content and space being ignored, Han et al. proposed the sequential views to sequential labels (SeqViews2SeqLabels) model [37]. It is similar to the viewpoints setting of GVCNN [36] in that it captures continuous views in a circle around each 3D object, forming a view sequence composed of V views evenly distributed on the circle in order. The camera is set at an angle of 30° above the ground and points at the center of mass of the 3D object. A position on the circle is randomly selected as the start of the view sequence, and subsequent views are acquired in the same sequence direction at $360^\circ/V$ intervals. Although the top or bottom of 3D object cannot be completely covered as is done in GVCNN [36], the sequential view method adopted by Han et al. enables the low-level features to be captured more effectively, and the spatial information between the views is retained for 3D global feature learning at the same time [37].

The abovementioned methods are based on all views used for training and inference. However, due to occlusion in the real world, objects can only be viewed from limited perspectives, which makes it difficult to rely on all views captured in an entire circle [30]. In order to address this problem, Kanezaki et al. proposed RotationNet [30]. In this model as well, a few predefined camera viewpoints are set initially, as shown in Fig. 4, in which case (i) and case (ii) are consistent with MVCNN [31]. Case (iii) is an extension of case (i), which has multiple elevations. In case (iii), Kanezaki et al. set up virtual cameras at intervals of ϕ in the angular range of [-90°, 90°]. There are $M = M_a \times M_e$ viewpoints, where $M_a = 360^\circ/\ \theta$ and $M_e = (180^\circ/\ \phi) + 1$. RotationNet [30] allows sequential input of views and updating of possible categories of the target objects. It explores a more general view configuration in addition to a regular view configuration [33]. However, it has the limitation of requiring each image to be viewed from one of the predefined viewpoints, which is very limited when there are fewer predefined viewpoints.

RotationNet [30] has few predefined viewpoints and requires inputting all views into the network in the training stage. However, processing all views requires a high degree of computation, and not every view is helpful for recognition [44]. Thus, in order to input the network with as few views as possible, Chen et al. proposed the view-enhanced recurrent attention model (VERAM) [44]. It places the object at the center of the observation sphere. The camera points at the center of mass of the object, whose position is expressed in latitude and longitude and can move on the observation sphere. The predefined camera positions of VERAM [44] consist of discrete viewpoints sampled at 30° intervals in latitude and longitude. All views collected from an object can be arranged in a 12 × 12 grid, represented by 1 to 12 in the vertical and horizontal directions, respectively. Chen et al. believe that the advantage of this configuration is that it provides a continuous space for agents to deploy cameras in the coordinate system [44]. There are a

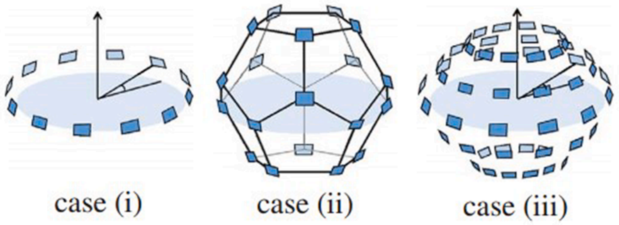

Fig. 4. Illustration of the three viewpoints setups considered in RotationNet [30].

sufficient number of predefined viewpoints in VERAM [44], and the viewpoint configuration in VERAM is more comprehensive than that in RotationNet [30]. The input mode of VERAM [44] also belongs to the sequential input category, which has also been used by other studies [38,43,63].

4. Deep learning methods for multi-view 3D object recognition

Benefiting from the advancements in CNN and its success in 2D image classification, we can learn a large number of general features of 2D image classification by using advances in image descriptors [64,65] and fine-tuning the details of the 3D model projection [31]. The deep learning methods for multi-view 3D object recognition can directly use the successful advanced classification network, trained by the excellent large-scale 2D public datasets (such as ImageNet [66]), as the backbone network. This saves the time of training the network, reduces the complex 3D classification task to a simple 2D classification task, and improves the network accuracy. In this method, the CNN, which performs well in 2D classification task, is used for extracting the features [67] from 2D rendered images, and the extracted features are fused to obtain the shape descriptor, which is used for classification. The time axis of the deep learning methods developed for multi-view 3D object recognition in recent years is shown in Fig. 5.

4.1. Backbone and feature extraction

The backbone network [33] is a CNN that performs well in 2D classification tasks. It is used in the 3D recognition task for feature extraction at the beginning of the network and classification at the end of network; thus, making it the skeleton of the method. The 3D recognition task first uses a mature classification network trained by large-scale 2D datasets to extract the view-level features in different views.

In MVCNN [31], Su et al. used the VGG-M network proposed in previous work [68] as the backbone network [33] for feature extraction. It is mainly composed of five convolutional layers, $conv_{1,\dots,5}$, three fully connected layers, $fc_{6,\dots,8}$, and a softmax classification layer, in which fc_7 is used as an image descriptor. The network is pretrained by ImageNet [66], and all views are fine-tuned in the training set. Su et al. have proved that fine-tuning can significantly improve the network performance because the performance of the CNN features in classification and retrieval is better than that of the popular 3D shape descriptors (such as SPH [69], LFD [70]), and 3D ShapeNets [13]. However, due to the earlier publication of this article, only the VGG-M [68] network has been used as the backbone, and other mature classification CNNs have not been used and compared (Su et al. proposed these improvements based on MVCNN [31] and named the new network MVCNN-new [49], which will be discussed in Section 5.3.).

GVCNN [36] uses GoogLeNet [71] as the backbone network [33]. The preliminary feature extraction part in this network is the fully convolutional network (FCN) designed by them, which comprises the first five convolution layers of GoogLeNet [71]. Compared to deep CNN, a shallow FCN can provide more location information [36].

RotationNet [30] uses AlexNet [36] as the backbone network [33] for feature extraction, which is smaller than the VGG-M [68] network architecture used in MVCNN [31]. It can achieve competitive performance for 3D object retrieval and classification [72] with fewer parameters. The ILSVRC 2012 [73] dataset is used for fine-tuning the weight of pretraining.

SeqViews2SeqLabels [37] uses VGG19 [74] to extract the underlying features of every single view. VGG19 [74] consists of 19 weight layers, including 16 convolution layers and three fully connected (FC) layers. After ImageNet [66] pretraining, VGG19 [74] is fine-tuned through all continuous views in the training set. Each view is divided into an object class by a softmax layer, and the low-level features of the view are extracted from the last FC layer.

View-GCN [33] uses ResNet-18 [75], which has been pretrained by ImageNet [66], for feature extraction. The network performs fine-tuning on the mixed multi-view images, and the features obtained from the different views are vectorized into view features before the last FC layer, which is used as the initialization of the node features [33].

4.2. Feature fusion

The view-level features extracted by the mature 2D classification network require feature fusion to generate the global descriptors and provide an accurate object classification. Developing a procedure for aggregating multiple view features into a differentiated global representation is a key challenge for these methods. The original methods used all views collected from all viewpoints. However, because processing all views requires a high degree of computation, and not every view is helpful for recognition, some methods introduced representative views selection mechanism and others introduced the next viewpoint selection mechanism. According to the different selection mechanisms of existing feature fusion methods, we can divide them into passive views selection mechanism, active views selection mechanism and active camera viewpoints selection mechanism. Active selection mechanism can enable the network to select the best views automatically, thus saving the computational cost and improving the network performance.

4.2.1. Passive views selection

In MVCNN [31], Su et al. found that the 3D shape classification constructed from a 2D image rendering of a 3D object can be much more efficient than the classifier constructed directly from a 3D representation. The structure of MVCNN [31] is shown in Fig. 6. Each image in the multi-view representation generates the descriptors of each 2D view using the first CNN of the network. These descriptors need to be aggregated in the view-pooling layer and merged into a single compact 3D shape descriptor in the view-pooling manner and then sent through the second CNN of the network for the object recognition task. Su et al. used the cross-view element-wise maximum operation in the view-pooling layer, but the mean operation was not effective. Experiments show that the classification retrieval performance is the best when the view-pooling layer is close to the last convolutional layer. The difference between the view-pooling layers, the max-pooling layers, and the maxout layers [76] is the dimension in which they perform the pooling

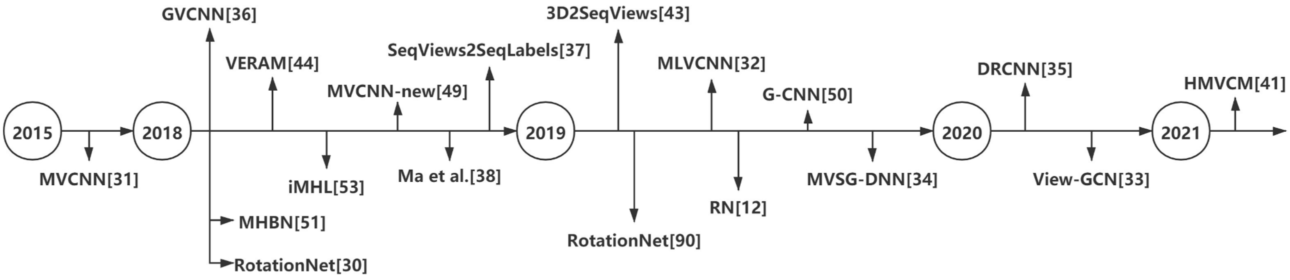

Fig. 5. Time axis of the deep learning methods developed for multi-view 3D object recognition in recent years.

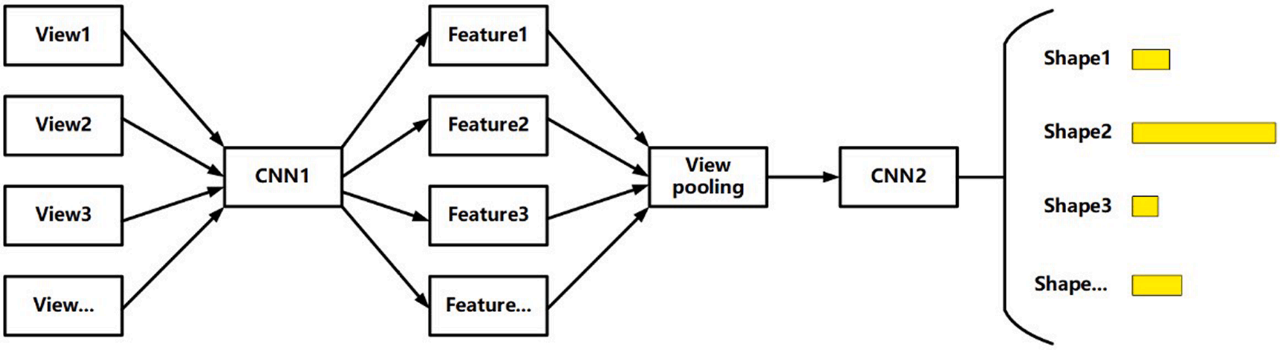

Fig. 6. Architecture of MVCNN [31].

operations. Training or fine-tuning of MVCNN [31] can be achieved by random gradient descent with backpropagation. Su et al. found that using fc_7 as an aggregate shape descriptor can lead to high performance, and this aggregate descriptor provides a significant acceleration. They also found a method that can significantly improve the retrieval performance of the network. They learned a Mahalanobis metric, W, so that the distance, l_2, in the projection space is very small between the same type of objects. Metric learning can be easily used for outputting the shape descriptors, which is an advantage of the CNN structure [31]. In addition, Su et al. used MVCNN [31] on the sketch recognition benchmark [77], and achieved good network performance. They introduced a 2D image classification method into a 3D object recognition area and achieved good performance, which greatly enriched the content and simplified the solution in this area.

Su et al. later conducted an in-depth analysis of MVCNN [31]. By changing the rendering background to black and improving the centering of the object, the performance of MVCNN-new [49], which has a deeper architecture, was further improved. However, they did not improve on feature fusion.

Multiple views of each 3D object have different importance in shape description. However, MVCNN [31] shares the information equally among all views and ignores the uncommon information contained in the different views, which limits the performance of the network. In terms of the distinguishability of different views, GVCNN [36] introduces a hierarchical view-group-shape framework. Feng et al. first generated group-level descriptions from view-level descriptors. In this step, the grouping module learns the group information, considers the correlation between the different views, helps in mining the relationship between the different views, and calculates the weight of the different groups to distinguish the view groups. After grouping, the view contents of the same group are similar. Here, all views in the same group are pooled by the view-pooling layer to obtain a group-level description. These group-level descriptions are weighted and fused to generate the shape-level descriptions. By using a hierarchical view-group-shape description framework, unique visual content can be found at the group level and thus emphasized in the shape descriptor [36]. But the views in the same group are so similar that some of them are undoubtedly redundant.

Although GVCNN [36] consider different information from different views and refer to the idea of a group to process the views, their view aggregation process still depends on pooling. However, pooling only retains the maximum or average value of all views, and it has a permutation invariance [33] which ignores the content information of almost all views and the spatial information between the views. Expressing the 3D graphics efficiently is still a challenging problem. To solve the problem of invariant, Esteves et al. proposed the group-convolutional network (G-CNN) [50]. Some papers call it EMV (equivariant multi-view network). It is also based on the idea of “group”. In particular, Esteves et al. pointed out in their paper that GVCNN [36] refers to the combination of view-level features as ”group”, and the ”group” of G-CNN [50] is an algebraic definition. Inspired by the work of Cohen et al. [78], Esteves et al. designed a multi-view aggregation group-convolutional approach to learn the equivalent representation of group transformation. In this method, convolution is performed on discrete subgroups of the rotation group, so that all views can infer jointly in an equivariant way until the last level. Further, Esteves et al. operated on the smaller discrete homogeneous space of the rotation group. G-CNN [50] has a local support filter in the group, which can learn complex representations from the superposition of layers and an increase in the receptive field. Finally, invariant descriptors are obtained by pooling the last layer of G-CNN [50]. Esteves et al. introduced excellent mathematical methods into 3D object recognition, which improved the theoretical level of this field.

In order to solve the problem of pooling, Yang et al. proposed the relation network (RN) [12] from the perspective of modeling the relationship from region to region and view to view. RN [12] connects the corresponding regions of different viewpoints through the self-attention modules to enhance the information of a single viewpoint image. The relationship between different views is used for view integration, and different 3D shape representations are realized.

Similarly, in order to solve the problem of pooling, Han et al. proposed the SeqViews2SeqLabels [37] network inspired by the recurrent neural networks (RNNs) [79]. It consists of an encoder RNN and a decoder RNN, and learns the 3D global features from sequential views. Han et al. used an encoder RNN to aggregate the sequence views, enabling the network to effectively learn the global features with semantics. Then, they used a decoder RNN to predict the sequence tags based on the learned global features. The decoder RNN can capture more and better identification information, reduce overfitting effectively, and achieve better performance. An attention mechanism is often used for extracting the parts of interest from the images [80]. Thus, Han et al. added an attention mechanism to the decoder RNN, which assigns more weight to the specific view-level features of each object, improves the discrimination ability, and can significantly reduce the influence of the first view position, which helps the encoder RNN learn the semantics of the view sequences. They also proposed another multi-view 3D object recognition method, called 3D2SeqViews [43], based on the attention mechanism. Using a new hierarchical attention gathering method, the content information in all continuous views and the continuous space between views is effectively gathered.

Further, based on RNNs [79], by introducing long short-term memory (LSTM) [79], Jiang et al. proposed the multi-loop-view convolutional neural network (MLVCNN) [32], which further improves the feature fusion mechanism. MLVCNN [32] introduces a hierarchical view-loop-shape structure, namely, the view level, the loop level, and the shape level, to represent a 3D shape. Firstly, the view features are extracted from a given multi-loop-view data to obtain the view-level descriptor. Among them, Jiang et al. proposed the concept of loop normalization (LN) [32]. In LN [32], only the pixels that share the same channel and loop are normalized together. It can represent the information efficiently because it can prevent local saliency from being weakened by global normalization and can be regarded as local normalization in the loop dimension. Based on LN [32], Jiang et al. designed the LoopNorm Block (LN Block) [32] with reference to the residual blocks in ResNet18 [75], and LN [32] was added into it. By inputting view-level descriptors into the superimposed LN Blocks [32], superior view-level features can be obtained. After passing through the LN Block [32], the features in the same loop are sent to the LSTM [79] network as sequence elements, and maxpooling is applied to the output in order to obtain the loop-level descriptors. LSTM [79] improves the recognition ability of the loop-level features because it considers the relationship between different views and enhances the relationship between the features obtained from them. Global features are formed by the features extracted from each loop. Finally, the FC layer uses the global features to extract the shape-level descriptors for 3D shape retrieval. Ma et al. [38] also used LSTM [79].

Liu et al. combined the ideas of the group and LSTM [79] in the above paper [32,36] and proposed the hierarchical multi-view context modeling method (HMVCM) [41]. It includes view-level context learning, a multi-view grouping module, group-level context learning, and a group fusion module. Firstly, the visual context features [81] of a view and its neighborhood are learned through the view-level context learning module. Liu et al. designed a bidirectional LSTM (Bi-LSTM) [41] by combining two LSTM [79] modules with the opposite input order for learning the visual conversion between the adjacent views and generate the view-level context features. The module is based on the CNN and Bi-LSTM [41] network, which can simulate the human behavior of looking back and forth. Then, using the multi-view grouping module, the grouping features are obtained by the weighted summation of the grouped views. Bi-LSTM [41] is used as a group-level context learning module for generating group-style context features from the context between the adjacent groups for visually distinguishing between the adjacent groups. Finally, the group context fusion module adaptively fuses a compact 3D shape descriptor according to the importance of all features.

In contrast to the view-based methods pooling the view-wise features, Yu et al. solved this problem from the perspective of patches-to-patches similarity measurement. They used harmonized bilinear pooling as a layer of the network, forming the multi-view harmonized bilinear network (MHBN) [51]. They used the relationship between the polynomial kernel and bilinear pooling to obtain an effective 3D shape representation. Zhang et al. put forward the multi-hypergraph learning in the label propagation methods [82]. They proposed an inductive multi-hypergraph learning algorithm (iMHL) [53], which uses the high-order correlation between 3D objects, represents all training data in a multi-hypergraph based on features, and simultaneously learns the weight of the projection matrix and optimal multi-hypergraph combination via inductive learning.

The application of the capsule network [83] in the feature fusion of different views is also a good attempt. In order to improve the defects of CNN, Sabour et al. proposed the capsule network [83], and one of the participants in this work is Hinton, the father of deep learning. Inspired by the dynamic routing algorithm of the capsule network [83], Sun et al. [35] modified the dynamic routing algorithm and constructed a dynamic routing layer (DRL). In this method, rearrangement and affine transformation are performed to transform the features. Then, the improved dynamic routing algorithm is used for adaptively selecting the transformed features, instead of ignoring the features, except the most active features in the view-pooling layer. As a result, the features of each view are fused effectively. Sun et al. also showed that the traditional view-pooling layer method is a special case of DRL. On the basis of DRL, they further proposed a dynamic routing convolutional neural network (DRCNN [35]) for multi-view 3D object recognition.

4.2.2. Active views selection

Not every view is essential for recognition. In order to fully mine the context information of multi-view and discover the distinctive structure of multi-view sequences, Zhou et al. proposed the multi-view saliency guided deep neural network (MVSG-DNN) [34] from the perspective of view selection. They designed a multi-view saliency mechanism, in particular, a saliency LSTM [34] for representative view selection. In this method, using the extracted 2D visual feature set, representative views are adaptively selected by exploring the multi-view context. Saliency LSTM [34] can fully extract the features of a view sequence and determine the view that contributes the most to the 3D objects instead of treating each view equally.

Similarly, based on the idea of view selection, Wei et al. proposed view-GCN [33] from the perspective of graph convolution [84–87]. Structure information is a very important type of information. A graphic structure can be used for representing higher-level information [88]. The nearest neighbor (NN) algorithm is a booming field in computer vision and machine learning [89]. Wei et al. proposed a view-graph representation of multi-views in 3D shape. Each view corresponds to a graph node, and the graph edges between the nodes are determined by the k nearest neighbors of the camera coordinates. This view-graph-based representation can model different view configurations, including regular and irregular camera positions. View-GCN [33] is a hierarchical network based on view-graph representation. In each layer, it performs local graph convolution and non-local message passing operations on the relationships between adjacent views and long-range paired views and considers the multi-view features of graph node relationships. Wei et al. used a selective view sampling strategy to sample representative views through the view selector. Finally, the learned features of different levels are combined into a 3D object descriptor. By introducing graph convolution and a selective view-sampling strategy, view-GCN [33] achieves excellent performance.

Although the view selection mechanism has been introduced in the study by Wei et al. [33], it still needs to input all views collected from all viewpoints into the network.

4.2.3. Active camera viewpoints selection

In the abovementioned methods, all viewpoints are set in advance, and all views are inputted into the network, which is a simple and effective strategy. However, the best performance can only be obtained by using all views, and the processing of all views is computationally demanding; not every view is vital for the network [44]. In real-world scenarios such as robot active recognition, we hope to realize object recognition using only a few views in order to solve the occlusion problem [30] and reduce the cost of mobile robot [44]. It is very important for practical applications to use only a few views to obtain accurate inferences. Thus, some multi-view 3D object recognition methods introduced the active camera viewpoints selection mechanism in their networks.

RotationNet [30] is differentiable, and the viewpoint variables, v_i, are set as latent variables that are to be optimized in the training process. For a multi-view input, the parameters of the network and v_i are optimized alternately to output the maximum probability, y. When the input image belongs to the v_ith viewpoint, the category likelihood value, P, related to v_i should be close to 1; otherwise, P might not be large. In other words, v_i is determined according to the output value of P. In order to obtain a stable solution of v_i, Kanezaki et al. added an " incorrect view" class to the target category to represent the negative samples that cannot judge the category. The parameters of the network can be updated iteratively by standard backpropagation. In the reasoning stage, RotationNet [30] inputs a small number of rendered images, outputs the corresponding probability, and then estimates the category of the objects depending on the output, and v_i is also determined by the output P. Therefore, RotationNet [30] can estimate the pose of an object and its category label. Kanezaki et al. regarded v_i as the latent variables of optimization in the training process and found the correct viewpoint by rotation in predicting the object labels and gaining the ability to found the corresponding viewpoints for the input views. In the inference stage, the viewpoints selection mechanism was introduced, which enables RotationNet [30] to achieve excellent performance.

Kanezaki et al. republished their paper on RotationNet [90] this year, and added the contents of self-alignment and viewpoint augmentation in Section 3 of their paper. The concept of self-alignment emphasize that the most distinctive feature of RotationNet [90] is that it can align the objects in the dataset during the training process and determine the base axis of the object in an unsupervised way depending on the appearance of the object. The concept of viewpoint augmentation is based on the defect where each object can only be viewed from a limited number of predefined viewpoints in RotationNet [30]. In order to reduce this limitation, view enhancement has been introduced in the training and inference stages.

Although the idea of viewpoints selection mechanism existed in RotationNet [90], it only seeks the corresponding viewpoints for input views. In addition, a complete multi-view image set of all the objects captured by the predefined viewpoints is still needed in the training process [30], and a small number of views are used only in the inference stage. Therefore, it does not constitute a complete active camera viewpoints selection. In terms of the viewpoints selection mechanism, VERAM [44] is an excellent network. Inspired by the visual attention model based on RNN [91,92], Chen et al. introduced the active camera viewpoints selection mechanism in the real sense. VERAM 44] is an agent that interacts with 3D objects through camera sensors. The architecture includes a virtual camera sensor, an observation subnetwork, a recurrent subnetwork, a view estimation subnetwork, and a classification subnetwork. The observation subnetwork encodes the position and content of the observation, extracts the view features, and updates the internal state of the next module with its feature vectors. The recurrent subnetwork encodes the environmental knowledge of the agent and summarizes the historical information of past observations. In this step, by virtue of the internal state generated by the RNN [44], the agent can specify the position of the camera sensor, i.e., the next viewpoint. The view estimation subnetwork uses the current internal state as a controller to guide attention. The classification subnetwork outputs the classification probability according to the final internal state to be classified. The historical information of the agent and 3D object interaction is integrated into the final internal state. Because of its hybrid architecture, it is easy to encounter problems such as dealing with non-differentiable components, keeping the balance between the subnetworks, and overfitting caused by huge number of parameters, while using VERAM [44]. Chen et al. used the stochastic gradient descent (SGD) [93] and REINFORCE [94] methods to solve these problems. Training imbalance among subnetworks is a common problem in the RNN-based attention models. Hence, Chen et al. proposed three schemes to adjust the gradient backpropagation information flow from the view estimation subnetwork to the hidden unit. The enhanced gradient information flow can keep the balance between the subnetworks, and effectively solve the problem of estimating the view stuck in the view parameter space boundary. In the reinforcement learning process of the attention model, Chen et al. realized the estimation of the next viewpoint by integrating the classification confidence of the current view into the gradient calculation of view reward. In addition, they proposed a new loss function to avoid duplicate estimation views. VERAM [44] can independently select the shortest view sequence and input it into the network for 3D shape classification that has the ability of active camera viewpoints selection. This not only saves the computational cost but also combines multi-view 3D object recognition with active vision, thus further improving the intelligence of this field.

4.3. Summary

A 3D recognition task uses a mature classification network, such as VGG-M [68], VGG19 [74], GoogLeNet [71], AlexNet [95], ResNet-18/50 [75], DenseNet [96], etc., which is pretrained by the large-scale 2D datasets for view-level feature extraction and can be used as the backbone network [33]. View-level features require feature fusion to generate global descriptors and provide accurate object classification. The original method involves a simple maxpooling or average-pooling of the features extracted from each view while ignoring the relationship between the view features. By introducing the concepts of the group [36], RNN [91,92], LSTM [79], dynamic routing [83], GCN [33], and other methods, full use of the content and spatial relationship between the different views have been made in recent studies to efficiently aggregate multi-view features into global 3D shape descriptors. The passive views selection method sets all viewpoints in advance and inputs all views into the network, which is not only computationally expensive but also achieves the best performance only when all views are used. Although the active views selection methods judge and use representative views, all views still need to be inputted into the network, which is not only costly but also difficult in terms of solving practical problems such as occlusion. Active camera viewpoints selection methods can achieve object recognition with as few views input network as possible, solve the occlusion problem, and reduce the cost of mobile robot [44], which is very important for real-world applications.

5. Performance

Various datasets are involved in the experiments of multi-view 3D object recognition, such as ModelNet40/10 [13], ShapeNetCore [60], RGB-D [59], MIRO [30], and other datasets, including classification and retrieval. Because most methods use ModelNet40/10 [13], we classify and summarize the performance of the different networks achieved using the ModelNet40/10 [13] dataset. The experimental results of the methods have been mainly summarized with the image as the modality. The tables of the experimental results provided in different papers that contain voxel, point cloud, and other modalities will also be summarized together.

5.1. ModelNet40

The performance summary of each method on ModelNet40 [13] is shown in Table 2.

It can be seen from the results in Table 2 that with the viewpoints selection mechanism and the representative views selection mechanism, the highest performance is achieved by RotationNet [30] and View-GCN [33] among all the classification task methods listed in the table. G-CNN [50] and DRCNN [35] are the two best retrieval task methods, which shows that excellent mathematical methods and dynamic routing methods can also greatly improve recognition performance.

5.2. ModelNet10

The performance summary of each method on ModelNet10 [13] is shown in Table 3.

It can be seen from the results in Table 3 that the highest performance is achieved by RotationNet [30] and DRCNN [35] among all the classification task methods listed in the table. Further, the two best-performing networks among the retrieval task methods are G-CNN [50] and DRCNN [35]. This conclusion is similar to that obtained in Section 5.1.

In addition, Kanezaki et al. published a paper reporting a revised version of RotationNet [90]. The experimental content in this publication is richer than the earlier version of RotationNet [30], and from the results given in Tables 2,3, it is emphasized that their revised method achieves excellent performance.

5.3. Backbone

Su et al. [49] conducted comparative experiments on MVCNN [31] based on four different backbone networks (Table 4). Their results show that MVCNN [31] is robust even when different backbone networks [49]

Table 2
Classification and retrieval performance of the different methods achieved using the ModelNet40 [13]dataset.

Method	Modality	Views	Classification (Accuracy)		Retrieval
			Class (%)	Instance (%)	mAP (%)
(1) SPH [69]	Mesh	-	68.23	-	33.3
(2) LFD [70]	Mesh	10	75.47	-	40.9
(3) RED [97]	Multi-Modality	-	-	-	86.3
(4) Beam Search [98]	-	-	81.26	-	-
(5) ECC [99]	-	-	83.2	-	-
(6) Fisher vector [31]	-	12	84.8	-	43.9
(7) DLAN [100]	-	-	-	-	85.0
(8) AniProbing [20]	-	-	85.6	89.9	-
(9) SubVolSup [20]	-	-	86.0	89.2	-
(10) Multiple Depth Maps [101]	-	-	87.8	-	-
(11) Set-convolution [102]	-	-	90	-	-
(12) FusionNet [103]	-	-	-	90.8	-
(13) T-L Network [15]	Voxel	-	74.4	-	-
(14) 3D ShapeNets [13]	Voxel	12	77.3	-	49.2
(15) VoxNet [17]	Voxel	-	83.0	-	-
(16) 3DGAN [18]	Voxel	-	83.3	-	-
(17) SliceVoxel [19]	Voxel	1	-	85.73	-
(18) FPNN [21]	Voxel	-	88.4	-	-
(19) LightNet [104]	Voxel	-	88.9	-	-
(20) ORION [16]	Voxel	-	89.7	-	-
(21) O-CNN [22]	Voxel	12	90.6	-	-
(22) VRN [14]	Voxel	-	-	91.33	-
(23) MVCNN-MultiRes [20]	Voxel	20	91.4	93.8	-
(24) VRN Ensemble [14]	Voxel	-	-	95.5	-
(25) LP-3DCNN [105]	Voxel	-	92.1	-	-
(26) FoldingNet [24]	Points	1	-	88.4	-
(27) PointNet [25]	Points	1	86.2	89.2	-
(28) 3D Point CapsuleNet [106]	Points	-	-	89.3	-
(29) KD-Network [29]	Points	-	88.5	91.8	-
(30) PointCNN [107]	Points	-	-	91.8	-
(31) PointNet++ [26]	Points	1	-	91.9	-
(32) SO-Net [23]	Points	1	90.8	93.4	-
(33) DGCNN [108]	Points	-	-	93.6	-
(34) RS-CNN [27]	Points	-	-	93.6	-
(35) DeepPano [39]	Image	1	82.5	-	76.8
(36) Geometry image [40]	Image	1	83.9	-	-
(37) PyramidHoG-LFD	Image	20	87.2	90.5	-
(38) PANORAMA-NN [45]	Image	6	-	90.7	-
(39) Ma et al. [38]	Image	-	-	91.05	-
(40) MVCNN [31]	Image	12	89.5	-	80.2
(41) GIFT [42,54]	Image	64	89.5	-	81.94
(42) MVCNN [20]	Image	20	89.7	92.0	-
(43) MVCNN(GoogLeNet) [36]	Image	12	-	92.2	83.0
(44) DeepCCFV [109]	Image	-	-	92.5	-
(45) Pairwise [46]	Image	12	90.7	-	-
(46) GVCNN [36]	Image	8	90.7	93.1	84.5
(47) MVCNN with TCL [47]	Image	-	-	-	88.0
(48) SeqViews2SeqLabels [37]	Image	12	91.1	93.3	89.0
(49) 3D2SeqViews [43]	Image	-	91.5	93.4	-
(50) VERAM [44]	Image	-	92.1	93.7	-
(51) MVSG-DNN [34]	Image	12	92.3	-	83.7
(52) RN [12]	Image	12	92.3	94.3	86.7
(53) MLVCNN [32]	Image	12		94.16	92.84
(54) MVCNN-new [49]	Image	-	92.4	95.0	-
(55) G-CNN [50]	Image	20	92.6	94.67	93.56
(56) DomSetClust [48]	Image	-	92.8	93.8	-
(57) MHBN [51]	Image	6	93.1	94.7	-
(58) PANORAMA-ENN [55]	Image	1	-	95.56	86.34
(59) HGNN [52]	Image	-	-	96.7	-
(60) HMVCM [41]	Image	12	94.57	-	92.8
(61) DRCNN [35]	Image	12	94.86	96.84	93.9
(62) iMHL [53]	Image	-	-	97.2	-
(63) RotationNet [30]	Image	20	96.29	97.37	
(64) View-GCN [33]	Image	20	96.5	97.6	-

are used.

Kanezaki et al. used 11 camera system directions in their experiment to compare the performance of RotationNet [30] for three different backbone networks (Table 5). The results thus obtained show that the impact of changing the direction of the camera system is far greater than the impact of the performance differences between the different architectures. This shows that viewpoint setting is an extremely important factor in multi-view 3D object recognition [30].

Table 3
Classification and retrieval performance of each method achieved by using the ModelNet10 [13]dataset.

Method	Modality	Views	Classification (Accuracy)		Retrieval
			Class (%)	Instance (%)	mAP (%)
(1) SPH [69]	Mesh	-	79.79	-	44.05
(2) LFD [70]	Mesh	10	79.87	-	49.82
(3) RED [97]	Multi-Modality	-	-	-	92.15
(4) Beam Search [98]	-	-	88	-	-
(5) FusionNet [103]	-	-	-	93.11	-
(6) Multiple Depth Maps [101]	-	-	91.5	-	-
(7) 3D ShapeNets [13]	Voxel	12	83.54	-	68.26
(8) 3DGAN [18]	Voxel	-	91.0	-	-
(9) VoxNet [17]	Voxel	-	92	-	-
(10) SliceVoxel [19]	Voxel	1	-	91.4	-
(11) 3DDescriptorNet [110]	Voxel	-	-	92.4	-
(12) ORION [16]	Voxel	-	93.8	-	-
(13) LightNet [104]	Voxel	-	93.9	-	-
(14) VRN [14]	Voxel	24	-	93.8	-
(15) VRN-Ensembel [14]	Voxel	-	-	97.14	-
(16) LP-3DCNN [105]	Voxel	-	94.4	-	-
(17) PointNet [25]	Points	1	77.6	-	-
(18) KD-Network [29]	Points	-	93.5	94.0	-
(19) SO-Net [23]	Points	1	93.9	94.1	-
(20) FoldingNet [24]	Points	1	-	94.4	-
(21) Geometry image [40]	Image	1	88.4	-	-
(22) DeepPano [39]	Image	1	88.7	-	-
(23) PANORAMA-NN [45]	Image	6	-	91.12	-
(24) VIPGAN [111]	Image	12	-	-	90.69
(25) GIFT [42,54]	Image	64	91.5	-	91.12
(26) Pairwise [46]	Image	12	-	94.0	-
(27) MVSG-DNN [34]	Image	12	94.0	-	93.9
(28) SeqViews2SeqLabels [37]	Image	12	94.8	94.82	91.43
(29) 3D2SeqViews [43]	Image	12	-	-	92.12
(30) MHBN [51]	Image	6	95.0	95.0	-
(31) Ma et al. [38]	Image	12	-	95.29	93.19
(32) RN [12]	Image	12	95.1	95.3	-
(33) VERAM [44]	Image	-	95.3	95.5	-
(34) G-CNN [50]	Image	20	-	96.78	96.18
(35) PANORAMA-ENN [55]	Image	1	-	96.85	93.28
(36) HMVCM [41]	Image	12	95.7	-	93.9
(37) RotationNet [30]	Image	20	-	98.9	-
(38) DRCNN [35]	Image	12	99.3	99.34	96.15

Table 4
MVCNN [31] performance using different backbone networks [49].

Model	ModelNet40 [13]	
	Per class (%)	Per instance (%)
VGG-11 [74]	92.4	95.0
ResNet 18 [75]	92.8	95.6
ResNet 34 [75]	93.4	**95.9**
ResNet 50 [75]	**94.0**	95.5

Table 5
Comparison of the classification accuracy (%) using RotationNet [30] based on different architectures [30].

Model	ModelNet40 [13]		ModelNet10 [13]	
	Mean	Max	Mean	Max
AlexNet [95]	93.70 ±1.07	96.39	94.52 ±1.01	97.58
VGG-M [68]	94.68 ±1.16	**97.37**	**94.82 ±1.17**	**98.46**
ResNet-50 [75]	**94.77 ±1.10**	96.92	94.80 ±0.96	97.80

6. Conclusion and prospects

In the 3D object recognition task, benefiting from the advancements in CNN and its success in 2D image classification, the view-based methods can directly employ the successful advanced classification network, trained by the excellent large-scale 2D public datasets, as the backbone network. This saves the training network time, reduces the complex 3D classification task to a simple 2D classification task, and ensures network accuracy. The performance indicators of the existing 3D object recognition methods based on multi-view have fully demonstrated their advantages, but some aspects still remain in need of improvement. For example, efficient utilization of the relationship between multi-view images and improving feature fusion has been the mainstream research direction of recent research works. As an example, the passive views selection methods in multi-view 3D object recognition are computationally intensive and have limited performance, whereas the active views selection methods still require all views to be inputted, which suffers from practical problems such as occlusion. Advances in computer vision require the network to be able to automatically select the best viewing angle of the target, i.e., active camera viewpoints selection. These best views would have the most abundant image information and the highest distinguishability, which will help the network achieve the best recognition performance with as few input views as possible and reduce the cost of mobile robot in addition to solving the occlusion problem. In the future, active camera viewpoints selection will be related to robot active vision, so that the robot can find the best viewpoint to observe and recognize objects. It will be a good development direction in the field of multi-view 3D object recognition and will have very important application prospects.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (No. 61901436, 61772277), and supported by the Key Research Program of the Chinese Academy of Sciences (No. XDPB22).

References

[1] T. Pylvanainen, K. Roimela, R. Vedantham, J. Itaranta, R. Grzeszczuk, Automatic alignment and multi-view segmentation of street view data using 3d shape priors, in: Symposium on 3D Data Processing, Visualization and Transmission (3DPVT), volume 737, 2010, pp. 738–739.

[2] N. Hagbi, O. Bergig, J. El-Sana, M. Billinghurst, Shape recognition and pose estimation for mobile augmented reality, IEEE transactions on visualization and computer graphics 17 (2010) 1369–1379.

[3] Y. Zhang, Y. Chen, X. Bai, S. Yu, K. Yu, Z. Li, K. Yang, Adaptive unimodal cost volume filtering for deep stereo matching, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 12926–12934.

[4] C. Wang, X. Bai, X. Wang, X. Liu, J. Zhou, X. Wu, H. Li, D. Tao, Self-supervised multiscale adversarial regression network for stereo disparity estimation, IEEE Transactions on Cybernetics (2020).

[5] X. Ning, F. Nan, S. Xu, et al., Multi-view frontal face image generation: a survey, Concurr. Comput. Pract. Exp. 3 (2020), https://10.1002/cpe.6147.

[6] L. Zhang, W. Li, L. Yu, X. Dong, L. Sun, X. Ning, J. Xu, H. Qin, Gmface: A mathematical model for face image representation using multi-gaussian, Displays (2021), https://doi.org/10.1016/j.displa.2021.102022.

[7] C. Yan, G. Pang, X. Bai, Z. J, L. Gu, Beyond triplet loss: Person re-identification with fine-grained difference-aware pairwise loss, IEEE Trans. Multimedia (2021), https://doi.org/10.1109/TMM.2021.3069562.

[8] N. Xin, G. Duoduo, D. Xiaoli, T. Weijuan, Y. Lina, W. Chuansheng, Conditional generative adversarial networks based on the principle of homologycontinuity for face aging, Concurr. Comput. Pract. Exp. (2020), https://doi.org/10.1002/cpe.5792.

[9] C. Peng, X. Qi, X. Jian, D. Xiaoli, S. Linjun, L. Weijun, N. Xin, W. Guojun, C. Ziheng, Harnessing semantic segmentation masks for accurate facial attribute editing, Concurr. Comput. Pract. Exp. (2020). https://10.1002/cpe.5798.

[10] G. Zhai, Y. Zhu, X. Min, Comparative perceptual assessment of visual signals using free energy features, IEEE Trans. Multimedia 99 (2020) 1.

[11] X. Ning, K. Gong, W. Li, L. Zhang, X. Bai, S. Tian, Feature refinement and filter network for person re-identification, IEEE Trans. Circuits Syst. Video Technol. (2020), https://doi.org/10.1109/TCSVT.2020.3043026.

[12] Z. Yang, L. Wang, Learning relationships for multi-view 3d object recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 7505–7514.

[13] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3d shapenets: A deep representation for volumetric shapes, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1912–1920.

[14] A. Brock, T. Lim, J.M. Ritchie, N. Weston, Generative and discriminative voxel modeling with convolutional neural networks, arXiv preprint arXiv:1608.04236 (2016).

[15] R. Girdhar, D.F. Fouhey, M. Rodriguez, A. Gupta, Learning a predictable and generative vector representation for objects, in: European Conference on Computer Vision, 2016, pp. 484–499.

[16] N. Sedaghat, M. Zolfaghari, E. Amiri, T. Brox, Orientation-boosted voxel nets for 3d object recognition, arXiv preprint arXiv:1604.03351 (2016).

[17] D. Maturana, S. Scherer, Voxnet: A 3d convolutional neural network for real-time object recognition, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 922–928.

[18] J. Wu, C. Zhang, T. Xue, W.T. Freeman, J.B. Tenenbaum, Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling, arXiv preprint arXiv:1610.07584 (2016).

[19] R. Miyagi, M. Aono, Sliced voxel representations with lstm and cnn for 3d shape recognition, in: 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017, pp. 320–323.

[20] C.R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, L.J. Guibas, Volumetric and multi-view cnns for object classification on 3d data, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 5648–5656.

[21] Y. Li, S. Pirk, H. Su, C.R. Qi, L.J. Guibas, Fpnn: Field probing neural networks for 3d data, 2016 arXiv preprint arXiv:1605.062 40.

[22] P.S. Wang, Y. Liu, Y.X. Guo, C.Y. Sun, X. Tong, O-cnn: Octree-based convolutional neural networks for 3d shape analysis, ACM Transactions on Graphics (TOG) 36 (2017) 1–11.

[23] J. Li, B.M. Chen, G.H. Lee, So-net: Self-organizing network for point cloud analysis, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 9397–9406.

[24] Y. Yang, C. Feng, Y. Shen, D. Tian, Foldingnet: Point cloud auto-encoder via deep grid deformation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 206–215.

[25] C.R. Qi, H. Su, K. Mo, L.J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017a, pp. 652–660.

[26] C.R. Qi, L. Yi, H. Su, L.J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, arXiv preprint arXiv:1706.02413 (2017b).

[27] Y. Liu, B. Fan, S. Xiang, C. Pan, Relation-shape convolutional neural network for point cloud analysis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8895–8904.

[28] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.H. Yang, J. Kautz, Splatnet: Sparse lattice networks for point cloud processing, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 2530–2539.

[29] R. Klokov, V. Lempitsky, Escape from cells: Deep kd-networks for the recognition of 3d point cloud models, in: iProceedings of the IEEE International Conference on Computer Vision, 2017, pp. 863–872.

[30] A. Kanezaki, Y. Matsushita, Y. Nishida, Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5010–5019.

[31] H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, Multi-view convolutional neural networks for 3d shape recognition, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 945–953.

[32] J. Jiang, D. Bao, Z. Chen, X. Zhao, Y. Gao, Mlvcnn: Multi-loop-view convolutional neural network for 3d shape retrieval, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 8513–8520.

[33] X. Wei, R. Yu, J. Sun, View-gcn: View-based graph convolutional network for 3d shape analysis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1850–1859.

[34] H.Y. Zhou, A.A. Liu, W.Z. Nie, J. Nie, Multi-view saliency guided deep neural network for 3-d object retrieval and classification, IEEE Trans. Multimedia 22 (2019) 1496–1506.

[35] K. Sun, J. Zhang, J. Liu, R. Yu, Z. Song, Drcnn: Dynamic routing convolutional neural network for multi-view 3d object recognition, IEEE Trans. Image Process. 30 (2020) 868–877.

[36] Y. Feng, Z. Zhang, X. Zhao, R. Ji, Y. Gao, Gvcnn: Group-view convolutional neural networks for 3d shape recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 264–272.

[37] Z. Han, M. Shang, Z. Liu, C.M. Vong, Y.S. Liu, M. Zwicker, J. Han, C.P. Chen, Seqviews2seqlabels: Learning 3d global features via aggregating sequential views by rnn with attention, IEEE Trans. Image Process. 28 (2018) 658–672.

[38] C. Ma, Y. Guo, J. Yang, W. An, Learning multi-view representation with lstm for 3-d shape recognition and retrieval, IEEE Trans. Multimedia 21 (2018) 1169–1182.

[39] B. Shi, S. Bai, Z. Zhou, X. Bai, Deeppano: Deep panoramic representation for 3-d shape recognition, IEEE Signal Process. Lett. 22 (2015) 2339–2343.

[40] A. Sinha, J. Bai, K. Ramani, Deep learning 3d shape surfaces using geometry images, in: European conference on computer vision, 2016, pp. 223–240.

[41] A.A. Liu, H. Zhou, W. Nie, Z. Liu, W. Liu, H. Xie, Z. Mao, X. Li, D. Song, Hierarchical multi-view context modelling for 3d object classification and retrieval, Inf. Sci. 547 (2021) 984–995.

[42] S. Bai, X. Bai, Z. Zhou, Z. Zhang, L. Jan Latecki, Gift: A real-time and scalable 3d shape search engine, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 5023–5032.

[43] Z. Han, H. Lu, Z. Liu, C.M. Vong, Y.S. Liu, M. Zwicker, J. Han, C.P. Chen, 3d2seqviews: Aggregating sequential views for 3d global feature learning by cnn with hierarchical attention aggregation, IEEE Trans. Image Process. 28 (2019) 3986–3999.

[44] S. Chen, L. Zheng, Y. Zhang, Z. Sun, K. Xu, Veram: View-enhanced recurrent attention model for 3d shape classification, IEEE transactions on visualization and computer graphics 25 (2018) 3244–3257.

[45] K. Sfikas, T. Theoharis, I. Pratikakis, Exploiting the panorama representation for convolutional neural network classification and retrieval, 3DOR 6 (2017) 7.

[46] E. Johns, S. Leutenegger, A.J. Davison, Pairwise decomposition of image sequences for active multi-view recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3813–3822.

[47] X. He, Y. Zhou, Z. Zhou, S. Bai, X. Bai, Triplet-center loss for multi-view 3d object retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1945–1954.

[48] C. Wang, M. Pelillo, K. Siddiqi, Dominant set clustering and pooling for multi-view 3d object recognition, arXiv preprint arXiv:1906.01592 (2019).

[49] J.C. Su, M. Gadelha, R. Wang, S. Maji, A deeper look at 3d shape classifiers, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018.

[50] C. Esteves, Y. Xu, C. Allen-Blanchette, K. Daniilidis, Equivariant multi-view networks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1568–1577.

[51] T. Yu, J. Meng, J. Yuan, Multi-view harmonized bilinear network for 3d object recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 186–194.

[52] Y. Feng, H. You, Z. Zhang, R. Ji, Y. Gao, Hypergraph neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 3558–3565.

[53] Z. Zhang, H. Lin, X. Zhao, R. Ji, Y. Gao, Inductive multi-hypergraph learning and its application on view-based 3d object classification, IEEE Trans. Image Process. 27 (2018) 5957–5968.

[54] S. Bai, X. Bai, Z. Zhou, Z. Zhang, Q. Tian, L.J. Latecki, Gift: Towards scalable 3d shape retrieval, IEEE Trans. Multimedia 19 (2017) 1257–1271.

[55] K. Sfikas, I. Pratikakis, T. Theoharis, Ensemble of panorama-based convolutional neural networks for 3d model classification and retrieval, Computers & Graphics 71 (2018) 208–218.

[56] Y. Feng, J. Xiao, Y. Zhuang, X. Yang, J.J. Zhang, R. Song, Exploiting temporal stability and low-rank structure for motion capture data refinement, Inf. Sci. 277 (2014) 777–793.

[57] L. Carvalho, A. von Wangenheim, 3d object recognition and classification: a systematic literature review, Pattern Anal. Appl. 22 (2019) 1243–1292.

[58] A.S. Gezawa, Y. Zhang, Q. Wang, L. Yunqi, A review on deep learning approaches for 3d data representations in retrieval and classifications, IEEE Access 8 (2020) 57566–57593.

[59] K. Lai, L. Bo, X. Ren, D. Fox, A large-scale hierarchical multi-view rgb-d object dataset, in: 2011 IEEE international conference on robotics and automation, 2011, pp. 1817–1824.

[60] A.X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al., Shapenet: An information-rich 3d model repository, arXiv preprint arXiv:1512.03012 (2015).

[61] L. Zhou, X. Bai, X. Liu, J. Zhou, E.R. Hancock, Learning binary code for fast nearest subspace search, Pattern Recogn. 98 (2020) 107040.

[62] M. Savva, F. Yu, H. Su, A. Kanezaki, T. Furuya, R. Ohbuchi, Z. Zhou, R. Yu, S. Bai, X. Bai, et al., Large-scale 3d shape retrieval from shapenet core55: Shrec`17 track, in: Proceedings of the Workshop on 3D Object Retrieval, 2017, pp. 39–50.

[63] X. He, T. Huang, S. Bai, X. Bai, View n-gram network for 3d object retrieval, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 7515–7524.

[64] D.G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the seventh IEEE international conference on computer vision, volume 2, 1999, pp. 1150–1157.

[65] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: European conference on computer vision, 2010, pp. 143–156.

[66] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, 2009, pp. 248–255.

[67] X. Ning, W. Li, B. Tang, H. He, Buldp: biomimetic uncorrelated locality discriminant projection for feature extraction in face recognition, IEEE Trans. Image Process. 27 (2018) 2575–2586.

[68] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: Delving deep into convolutional nets, arXiv preprint arXiv:1405.3531 (2014).

[69] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3 d shape descriptors, in: Symposium on geometry processing, volume 6, 2003, pp. 156–164.

[70] D.Y. Chen, X.P. Tian, Y.T. Shen, M. Ouhyoung, On visual similarity based 3d model retrieval, Computer graphics forum 22 (2003) 223–232.

[71] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[72] A.A. Liu, W.Z. Nie, Y. Gao, Y.T. Su, View-based 3-d model retrieval: A benchmark, IEEE transactions on cybernetics 48 (2017) 916–928.

[73] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International journal of computer vision 115 (2015) 211–252.

[74] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[75] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[76] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, Y. Bengio, Maxout networks, in: International conference on machine learning, 2013, pp. 1319–1327.

[77] M. Eitz, J. Hays, M. Alexa, How do humans sketch objects? ACM Transactions on graphics (TOG) 31 (2012) 1–10.

[78] T. Cohen, M. Welling, Group equivariant convolutional networks, in: International conference on machine learning, 2016, pp. 2990–2999.

[79] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.

[80] X. Ning, K. Gong, W. Li, L. Zhang, Jwsaa: Joint weak saliency and attention aware for person re-identification, Neurocomputing 453 (2021) 801–811, https://doi.org/10.1016/j.neucom.2020.05.106.

[81] S. Bai, Z. Zhou, J. Wang, X. Bai, L. Jan Latecki, Q. Tian, Ensemble diffusion for retrieval, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 774–783.

[82] Y. Gao, M. Wang, D. Tao, R. Ji, Q. Dai, 3-d object retrieval and recognition with hypergraph analysis, IEEE Trans. Image Process. 21 (2012) 4290–4303.

[83] S. Sabour, N. Frosst, G.E. Hinton, Dynamic routing between capsules, arXiv preprint arXiv:1710.09829 (2017).

[84] J. Bruna, W. Zaremba, A. Szlam, Y. LeCun, Spectral networks and locally connected networks on graphs, arXiv preprint arXiv:1312.6203 (2013).

[85] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, arXiv preprint arXiv:1606.09375 (2016).

[86] M. Henaff, J. Bruna, Y. LeCun, Deep convolutional networks on graph-structured data, arXiv preprint arXiv:1506.05163 (2015).

[87] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907 (2016).

[88] B. Xiao, E.R. Hancock, R.C. Wilson, Graph characteristics from the heat kernel trace, Pattern Recogn. 42 (2009) 2589–2606.

[89] X. Bai, C. Yan, H. Yang, L. Bai, J. Zhou, E.R. Hancock, Adaptive hash retrieval with kernel based similarity, Pattern Recogn. 75 (2018) 136–148.

[90] A. Kanezaki, Y. Matsushita, Y. Nishida, Rotationnet for joint object categorization and unsupervised pose estimation from multi-view images, IEEE transactions on pattern analysis and machine intelligence 43 (2019) 269–283.

[91] J. Ba, V. Mnih, K. Kavukcuoglu, Multiple object recognition with visual attention, arXiv preprint arXiv:1412.7755 (2014).

[92] V. Mnih, N. Heess, A. Graves, K. Kavukcuoglu, Recurrent models of visual attention, arXiv preprint arXiv:1406.6247 (2014).

[93] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533–536.

[94] R.J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine learning 8 (1992) 229–256.

[95] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012) 1097–1105.

[96] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.

[97] C. Wang, X. Wang, X. Bai, Y. Liu, J. Zhou, Self-supervised deep homography estimation with invertibility constraints, Pattern Recogn. Lett. 128 (2019) 355–360.

[98] X. Xu, S. Todorovic, Beam search for learning a deep convolutional neural network of 3d shapes, in: 2016 23rd International Conference on Pattern Recognition (ICPR), 2016, pp. 3506–3511.

[99] M. Simonovsky, N. Komodakis, Dynamic edge- conditioned filters in convolutional neural networks on graphs, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3693–3702.

[100] T. Furuya, R. Ohbuchi, Deep aggregation of local 3d geometric features for 3d model retrieval., in: BMVC, volume 7, 2016, p. 8.

[101] P. Zanuttigh, L. Minto, Deep learning for 3d shape classification from multiple depth maps, in: 2017 IEEE International Conference on Image Processing (ICIP), volume 71, 2017, pp. 3615–3619.

[102] S. Ravanbakhsh, J. Schneider, B. Poczos, Deep learning with sets and point clouds, 2016 arXiv preprint arXiv:1611.04500.

[103] V. Hegde, R. Zadeh, Fusionnet: 3d object classification using multiple data representations, arXiv preprint arXiv:1607.05695 (2016).

[104] S. Zhi, Y. Liu, X. Li, Y. Guo, Toward real-time 3d object recognition: A lightweight volumetric cnn framework using multitask learning, Computers & Graphics 71 (2018) 199–207.

[105] S. Kumawat, S. Raman, Lp-3dcnn: Unveiling local phase in 3d convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4903–4912.

[106] Y. Zhao, T. Birdal, H. Deng, F. Tombari, 3d point capsule networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1009–1018.

[107] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, B. Chen, Pointcnn: Convolution on x-transformed points, arXiv preprint arXiv:1801.07791 (2018).

[108] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph cnn for learning on point clouds, Acm Transactions On Graphics (tog) 38 (2019) 1–12.

[109] Z. Huang, Z. Zhao, H. Zhou, X. Zhao, Y. Gao, Deepccfv: Camera constraint-free multi-view convolutional neural network for 3d object retrieval, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 8505–8512.

[110] J. Xie, Z. Zheng, R. Gao, W. Wang, S.C. Zhu, Y.N. Wu, Learning descriptor networks for 3d shape synthesis and analysis, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8629–8638.

[111] Z. Han, M. Shang, Y.S. Liu, M. Zwicker, View inter-prediction gan: Unsupervised representation learning for 3d shapes by learning global shape memories to support local view predictions, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 8376–8384.