IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 24, 2020, accepted March 6, 2020, date of publication March 20, 2020, date of current version April 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982196

A Review on Deep Learning Approaches for 3D Data Representations in Retrieval and Classifications

ABUBAKAR SULAIMAN GEZAWA [1], YAN ZHANG [1], QICONG WANG [1,2], AND LEI YUNQI [1]

[1] Department of Computer Science, School of Informatics, Xiamen University, Xiamen 361005, China
[2] Shenzhen Research Institute, Xiamen University, Shenzhen 518000, China

Corresponding author: Lei Yunqi (yqlei@xmu.edu.cn)

This work was supported by the National Natural Science Foundation of China (61671397) and the Shenzhen Science and Technology Projects under Grant JCYJ20180306173210774.

ABSTRACT Deep learning approach has been used extensively in image analysis tasks. However, implementing the methods in 3D data is a bit complex because most of the previously designed deep learning architectures used 1D or 2D as input. In this work, the performance of deep learning methods on different 3D data representations has been reviewed. Based on the categorization of the different 3D data representations proposed in this paper, the importance of choosing a suitable 3D data representation which depends on simplicity, usability, and efficiency has been highlighted. Furthermore, the origin and contents of the major 3D datasets were discussed in detail. Due to growing interest in 3D object retrieval and classification tasks, the performance of different 3D object retrieval and classification on ModelNet40 dataset were compared. According to the findings in this work, multi views methods surpass voxel-based methods and with increased layers and enough data augmentation the performance can still be increased. Therefore, it can be concluded that deep learning together with a suitable 3D data representation gives an effective approach for improving the performance of 3D shape analysis. Finally, some possible directions for future researches were suggested.

INDEX TERMS 3D data representation, 3D deep learning, 3D models dataset, computer vision, classification, retrieval.

I. INTRODUCTION

The increasing availability of 3D models from constructed and captured 3D data from low-cost acquisition devices and other modeling tools requires effective algorithms to perform key tasks such as retrieval [1]–[3], classification [4]–[7], recognition [8]–[10], and other 3D shape analysis tasks [11]–[15]. In 3D deep learning algorithm, there are two key challenges, i.e., the 3D data representation to use and the network structure adopted. Majority of the deep learning methods are deeply rooted in 2D data [15] which makes it more challenging on 3D data, fortunately with increase interest on 3D objects like the yearly 3D shape retrieval contest organized by [16] in order to evaluate the effectiveness of 3D retrieval algorithms and other 3D objects related tasks have increased tremendously the number of deep learning algorithm on different 3D data.

The success of deep learning in image feature extraction over handcrafted methods have attracted interest in raw 3D data like points clouds because of their simplicity and flexibility. [17], [18] process points cloud directly without any transformation to other formats. These approaches prove effective in 3D object classification and segmentation tasks. However, point clouds are permutation invariant and there is uncertainty on how these methods can be processed without affecting output quality. 3D data can also be projected into 2D space and still maintains some of the main properties of the original 3D data [19], [20] but their major limitation is information loss when the 3D objects are very complex, in order to overcome this shortcomings, some approaches adopted reconstruction methods that retrieved objects from a 3D shape database and use a real time scanning of the environment which helps to replace scanned RGB-D data with complete hand-modeled objects from the shape database and finally align and scale retrieved models to the input data to obtain a high quality virtual representation of the

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

real world environment which is very similar to the original geometry. Surfaces mesh is among the most popular 3D data representation which is commonly used in computer graphics and computer vision fields [21]. MeshCNN was Proposed in [22] which utilizes the key features of mesh data by combining convolution and pooling layers to operate on the mesh edges by taking advantages of their intrinsic geodesic connections. The major limitations of mesh data are its complexity and irregularity which makes it less usable in the research community as can be seen in Fig. 2 which shows the history of 3D data representation using different types of data from 2003 to 2018.

Voxels and octrees are the two popular solids 3D data representations that exploit the full geometry of the 3D objects. [23] represents the input object as a 30 × 30 × 30 binary tensor which categorized the 3D object, even though this approach is among the first to exploit 3D volumetric data on deep learning directly, they imposed many constraints among which is the additional kernel which leads to difficulty in processing high-resolution data. Octrees which is a varying sized voxel was exploited in [24] for 3D object classification, retrieval and segmentation tasks. The authors proposed octree based convolutional neural network that support a modest octree structure that supports all CNN operations on the GPU.

Some works exploit the high-level structures of 3D shape by expressing the 3D shape in form of high-level shape feature which is informative and contains geometric extract of the 3D objects. 3D descriptors can be divided into Global descriptor [25]–[27] and local descriptor [28]–[30]. The other classification is based on the essence of the information characterization e.g. non-diffusion 3D descriptors e.g. statistical moments [31], light field descriptor [32], and the diffusion-based descriptors [33]–[36] this will be discussed in details in section IID. Other two popular high-level 3D data representations are the graphs and skeleton. [37] Performed a 3D object retrieval based upon a graph-based object representation which is composed of new mesh segmentation along with a graph matching between graph of the query and each of the graph that corresponds to the object of the 3D objects database. In [38], a unified framework was proposed that generalize CNN architectures to graphs and manifolds and learn compositional task- specific features. The relationship between graphs and meshes will be elaborated in section III D. To have an effective deep learning model on 3D data representations, there is a need to consider the fundamental properties of the 3D data representations and its efficiency, simplicity and usability which is the main priority in this work.

3D data representation used in deep learning methods for 3D objects retrieval and classification have continued to grow rapidly in recent time. In [39], Griffiths *et al.* review the current state of the art deep learning architectures but focus on unstructured Euclidean data only, also in [40], Pichao *et al.* presented a detailed overview of recent advances in RGB-D based motion recognition. In this survey, the methods are categorized into four groups based on the modality adopted for recognition which are; RGB-based, depth-based, skeleton-based and RGB+D-based. A special consideration was given to the way in which the spatial, temporal and structural information have been utilized. Some of the key benefits of this survey is the comprehensive discussion of the challenges of RGB-D based motion recognition, analysis of the limitations of available methods and finally discussion of potential research directions. The aim of our paper is to broadly review the performance of deep learning methods on different 3D data representations. We emphasize on the computational difference of each 3D data representations which include: Simplicity (less difficulty of acquisition, hardware speed/timing), Usability (benefit) and Efficiency (effectiveness).

In [41], a survey was presented that classified 3D data representation into Euclidean and non-Euclidean data. However, the recent deep learning methods that explore octree data representation were not presented. Therefore, our current work presented all the different 3D data representations. currently used in deep learning architectures including the most popular 3D objects datasets available in the deep learning community. The main contributions of this paper are as follows:

1) A comprehensive review about the performance of deep learning methods on different 3D data representations with an emphasis on the computational difference of each 3D data representation based on simplicity, usability and efficiency.
2) Thorough Analysis about the future direction of 3D data representations used in deep learning models with literature to support the field where the future direction would be beneficial

II. REVIEW OF DIFFERENT CATEGORIES OF 3D DATA REPRESENTATIONS

3D data representations serve as the basis for computer graphics, robotics, visualization and many others. They serve as the language for defining geometry syntax and semantics. In this section, we reviewed in details the different categories of 3D data representation which include: Raw data, solids, surfaces, multi views and high-level structures. Recent work [42] focus on unstructured Euclidean data only in contrast to ours which extensively study both structured and unstructured data which will serve as a guide for choosing the suitable 3D data representation for future research.

A. RAW DATA

Raw 3D data can be obtained by different divergent scanning devices such as Microsoft Kinect [43], structured lights scanning [44] and many others. Some of the popular 3D data representations that belong to this group are point cloud, RGB-D data, and 3D projections subsubsectionPOINT CLOUD Point clouds have a background in photogrammetry and in recent time lidar. A point cloud is simply a set of 3D data points and each point is represented by three coordinates in a Cartesian or other coordinate systems.it is regarded as a set of unstructured 3D points that symbolize the geometry of 3D objects and are utilized in many computer vision tasks. E.g. classification and segmentation [17], object recognition [45], reconstruction [46] etc.

Even though points clouds can be obtained easily using Kinect [43] and other sensors like devices, processing them can be challenging due to absence of connectivity information in the point clouds and capturing them from the environment in the acquisition setup. The point clouds obtained sometimes are incomplete, with noise and missing data which can be caused due to constrained of the sensors [47] and other environmental factors [48]. Recent methods deal with noise reduction in point cloud [49].

1) RGB-D DATA

Microsoft Kinect [43] can be used to characterize 3D data to RGB-D images. It gives a 2.5D data about the obtained 3D object by giving the depth map (D) together with color information (RGB). Many RGB-D datasets are available like RGB-D object dataset [50], SUN 3D [51] and many more. RGB-D data prove to be effective in pose regression [52], correspondence [53] and character recognition [54].

2) 3D DATA PROJECTIONS

3D projections are a way of mapping 3D points to 2D planes. It is realized using imaginary projections which give the projected data crucial features of the original 3D object. Many of the projection's methods convert the 3D model to a 2D grid with key properties. Spherical domains projections [55] aid the data projected to be invariant to rotations. However, some projections are not optimal in complicated computer vision tasks due to loss of information in projections [56].

B. SOLIDS

Solids representations of 3D models are virtually space control information for a given object. Usually the information is binary which implies that the space can be occupied by the object or none. Octree and voxels are the two major solids representations used in deep learning community.

1) OCTREE

An octree is a simplified data structure for effective storage of 3D data. It is an extension of a 2D quadtree. The individual node in an octree contains eight children [57]. Octree is simply a fluctuating sized voxel and it is considered one of the most scattered voxel representations which was recently used in conjunction with CNN for 3D shape analysis task in [12] and [24]. It has the advantages of efficient memory utilization and can be used for generating high resolution voxels [58]. However, it has a major drawback which is caused by its inability to maintain the geometry of some 3D objects like the smoothness of the surface.

2) VOXELS

Voxels are used to represent 3D data by characterizing the manner the 3D object is allocated through the three dimensions of the scene. The occupied voxels can be classified into seeable block or self-occluded by encoding the view information about the 3D shape. [60] and [61] used a voxel variational auto encoder for shape classification. [62] Create CNN learning as a beam search with the intention of identifying an optimal CNN architecture namely, the number of layers, nodes, and their connectivity in the network, but the major limitations of voxels are its demand for unnecessary storage due to representation of both the occupied and non-occupied scene [63]. The huge demand for memory storage makes it not good enough for high resolution data [57].

C. SURFACES

Surfaces polygons are usually used in boundary representation of 3D objects which surround the inner part of the object. The set of this polygons are usually stored for the description of the object which has the benefit of simplicity and speeding of the rendering of the surface and object display because all surfaces can be characterize with linear equations. There are many methods for surface representations of 3D objects such as the polygon mesh, sub division, parametric and implicit but among these representations' polygon mesh is the most popular surface representations used in the deep learning community.

1) 3D MESH

3D meshes consist of a combination of vertices, edges and faces that are mostly used in computer graphics application for storing 3D objects and for rendering purpose. The vertices contain connectivity list that describes how each vertices are connected to one another. The major challenges of mesh data are, they are irregular and very complex, which makes them not usually used in deep learning methods until recently when [21] propose MeshNet which can to deal with the complexity and irregularity problems of mesh data and successfully performed 3D shape classification and retrieval task on Model 10 dataset. Also [22] used the edges of the mesh to perform pooling and convolution on the mesh edges by taking advantages of their intrinsic geodesic connections.

D. HIGH-LEVEL STRUCTURES

In 3D shape retrieval and classification, there is a need for succinct still, very rich representation of a 3D object that will be used to describe an object as representative of some category. 3D shapes can be represented in the form of high-level 3D shape descriptors which is a simplified representation that contains the geometric characteristics of the 3D object [25]–[30]. Apart from 3D descriptors, the graph can also be expressed in the form of high-level structures.

1) 3D DESCRIPTORS

3D shape descriptors play a significant role in 3D shape retrieval and other shapes analysis tasks. For 3D shape retrieval which can be describe as a way of querying a 3D object from a database of many 3D models in order to discover the closest match, there is a need to change the shapes into feature descriptors in order to keep the enormous volume of 3D data and expeditiously query to find the closest match [25]–[30]. 3D shape descriptors can be divided into Global descriptors [25]–[27] and local descriptors [28]–[30]. The other categories of 3D shape descriptors are the diffusion -based [33]–[35] and non-diffusion based [like the statistical moments [31], light field descriptor [32], and Fourier descriptor [64]. The diffusion-based descriptors are effective in capturing the geometric properties of the 3D shape. For example the Global Point Signature which is term as (GPS) [34] characterizes the shape of a 3D model by using the Laplace-Beltrami Operator that is defined on a 3D surface. Other two popular diffusion based descriptors are the Heat Kernel Signature (HKS) [33] and Wave Kernel Signature (WKS) [35] which both have the advantages of invariance to isometric deformations. Regardless of the capabilities of HKS, GPS and WKS they are still point based descriptors that lacks the global description of the shape. Temperature Descriptor which is a global descriptor was developed to represent the entire shape which is very efficient but represents the shape at only one single scale that leads to an incomplete description.

2) GRAPH

Graph 3D data representation collect the geometric essence from a 3D object by linking different shape parts using a graph. Graph approaches are usually categorized into three based on the category of graph used e.g. model graph [65], reeb graphs [66] and skeleton graph [67]. Meshes are also extended to graph-structured data in which the nodes of the graph are used as the vertices of the mesh and the edges represent the connections between the vertices [68].

E. MULTI VIEW DATA

Another form of 3D data representation is to render a set of images from verities of views and takes the pile image and use as an input to CNN which can be used for shape analysis tasks [69], [70]. The key benefits of these approaches are that they can handle high-resolution inputs as well as utilizing the full image-based CNNs for 3D shape analysis tasks. However, determining the number of views and self-occlusions are major draw-backs of these methods which can lead to huge computational cost if the numbers of views are large.

III. DATASETS

Datasets are very useful for fair comparisons between different deep learning algorithms. Synthetic and real-world datasets are the two major categories of datasets available in the deep learning community. Most of these datasets are produced by universities research groups or large industries. This section presents the benchmark datasets mostly used in testing the deep learning algorithms. We present the datasets in chronological order based on the category.

A. RGB-D DATASETS

The following list outlines the different types of RGB-D datasets that are captured using a Kinect or similar devices:

1) RGB-D OBJECTS DATASETS

This dataset consists of 11,427 RGB-D images which are manually segmented. The dataset was developed in conjunction with the Intel labs Seattle by researchers from Washington University. It is available freely for educational/noncommercial use and consists of 300 common objects that are categorized into 51 classes. The kinect style sensor was used to acquire the images and generate 640 × 480 RGB-D frames.

2) NYU DEPTH DATASETS

The NYU depth dataset was developed by researchers at Yew York University and was obtained using Microsoft Kinect V1 and consist of 1449 RGB-D segmentation labels for images of indoor scans. There are 40 classes of objects split and 407,024 validation images are available.

3) SUN RGB-D DATASETS

This dataset was developed at Princeton University using four sensors to acquire the datasets and consist 10,000 manually segmented images which are split into 63 classes of indoor scenes.

B. OTHER 3D DATASETS

The following list outlines other Synthetic and real-world datasets that are mostly used in the deep learning community.

1) ShapeNet

ShapeNet is a large collection of 3D objects e.g. bicycles, planes, bottles, chairs, tables, etc. that are developed by a team of researchers from Stanford and Princeton universities as well as TTIC institute. Two categories datasets are available that is ShapeNet Core that includes 51,300 3D models that are divided into 55 classes and ShapeNetSem which consist of 270 classes of 12,000 models.

2) SCAPE DATASET

One of the key aims of this dataset is providing human shapes models. The 3D models are obtained using both the shape and pose parameters so that changes due to deformation in both poses and shapes can be captured. Cyberware whole-body scanner is used to capture the surface data then meshes of the full body are created after merging scans of four directions that are captured by the scanner simultaneously. It consists of scanned human figure of 37 people having 70 different poses each.

3) TOSCA DATASET

The Tosca dataset consist of 3D non-rigid shapes which are categorized into Non-rigid world and Tosca high-resolution datasets. Both of the datasets consist of animals and humans in a variety of poses that can be used for shape analysis and correspondence experiments

4) SHREC DATASETS

Network of Excellence AIM @ SHAPE [71] initiated the now famous 3D Shape Retrieval Contest (SHREC) in 2006.

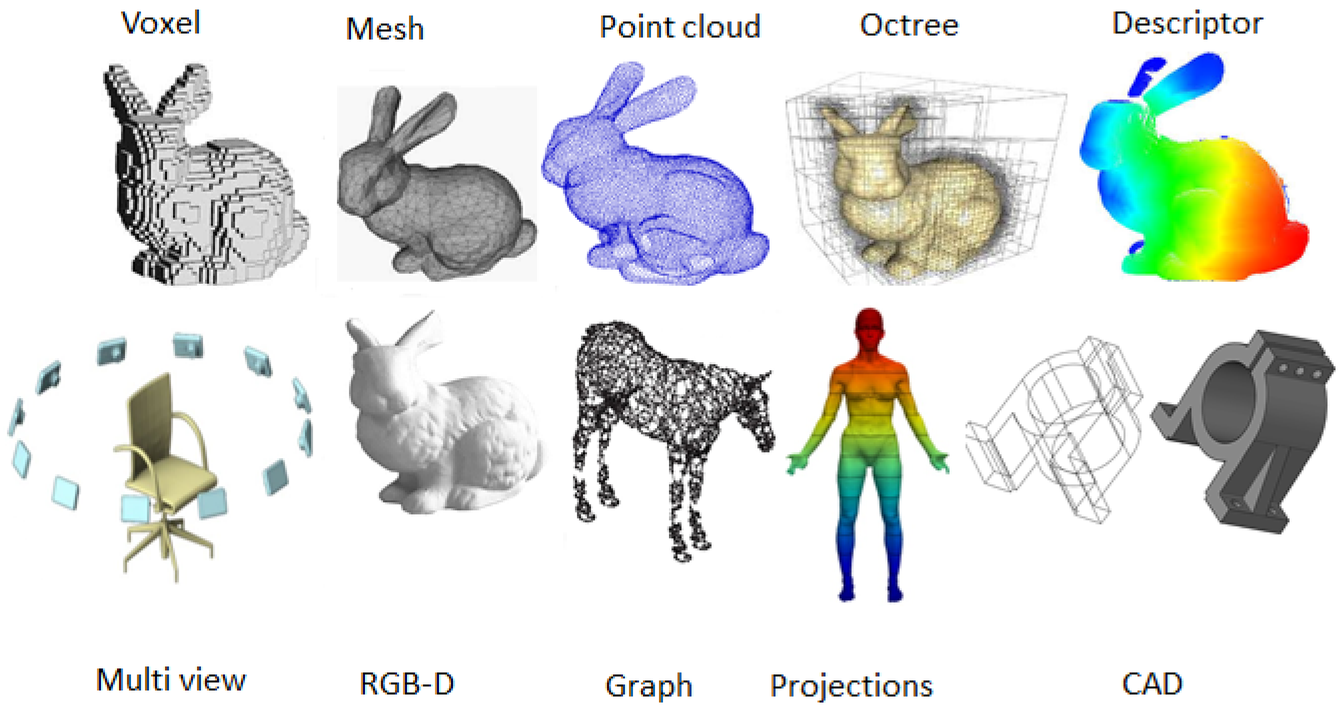

FIGURE 1. Examples of different 3D data representations.

It contains a track of models from the Princeton Shape Benchmark. Now it is organized in collaboration with the Eurographics Workshop on 3D object Retrieval every year including many tracks which consist of CAD models, protein models, water tight models and many more. The main aim is to provide researchers with an opportunity to test their algorithms by using common test collection.

5) NIST SHAPE BENCHMARK

The National Institute of Standard and Technology (NIST) shape benchmark consist of 800 3D models of daily life objects which are sub-divided into 20 models per class.

6) FAUST

The dataset contains 300 high resolutions of human body scans in 10 subjects and 30 poses. The models are obtained through full body high accuracy 3D multi-stereo system with 172,000 average numbers of vertices for each object and the ground truth correspondence are computed automatically.

7) ENGINEERING SHAPE BENCHMARK (ESB)

This dataset was developed by Purdue University researchers. It consists of 867 triangulated meshes mostly from CAD parts in the field of mechanical engineering. This dataset has two levels of order with 3-main classes and 4 sub-classes.

8) McGILL 3D SHAPE BENCHMARK

The models in this repository are mostly adopted from PSB and other internet websites that are mostly created from CAD modeling tools. The dataset consists of 456 models in total and 255 of which are articulated parts shapes. They are divided into 10 classes with 20-30 models in each category.

9) ModelNet

The ModelNet dataset is a comprehensive collection of clean 3D CAD models provided by researchers from department of computer Science of Princeton University. The datasets contain 127,915 CAD models belonging to 662 object categories. It is obtained using online search engines after querying for each object category and then manually annotated the data. ModelNet has two subsets which ModelNet10 and ModelNet40 that are mostly used in object recognition and classification tasks.

10) PERSISTENT HEAT SIGNATURE

This dataset was developed by researchers at Ohio state university for used in Shape Retrieval algorithms for partial and incomplete models with pose variations. The dataset consist of 50 queries models among which 18 are complete and the remaining are incomplete or partial models. In total in consists of 300 shapes from 21 classes of dogs, horses, airplanes etc.

11) ScanNet

This is one of the recent and rich datasets for real world scenes. The dataset was annotated and labeled with semantic segmentation and consist of 2.5 million views that are used

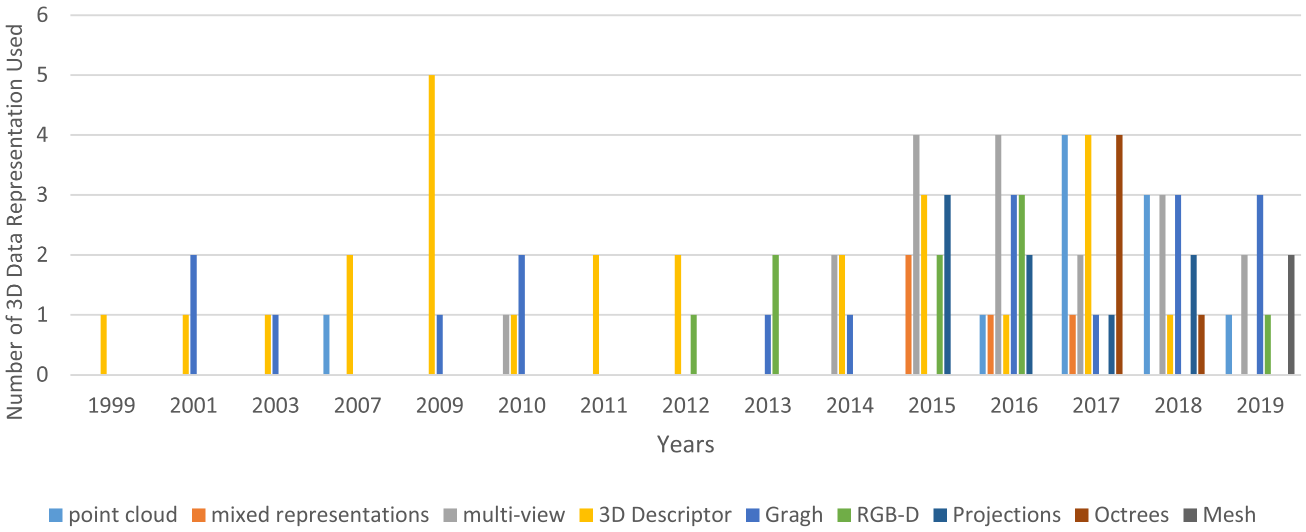

FIGURE 2. **Progress of 3D data representation along with time from 1999-2019.**

for training directly without the need for pre-training on other datasets.

IV. PERFORMANCE OF DEEP LEARNING METHODS ON DIFFERENT 3D DATA REPRESENTATIONS

With the increase number of 3D models, 3D shape retrieval and classification tasks continue to receive attention in the field of computer vision, medical imaging, graphics and many others. 3D shape retrieval deals with the issue of discovering 3D objects from a shape database that are similar to a particular query. Network of Excellence@Shape [71] initiated the famous 3D shape retrieval contest (SHREC) in 2006. It contains a track of models from the Princeton shape benchmark and now it is organized in collaboration with the Eurographics Workshop on 3D object Retrieval every year including many tracks that consist of CAD models, protein models, water tight models and many more. Deep learning has achieved tremendous success in image-based task [72]–[74]. Despite the success of deep networks in image concepts, in sufficient training examples can leads to decayed performance more especially for deep networks with strong representations structure making them vulnerable to overfitting. To address some of these problems, Shu *et al.* in [75] proposed a novel deep network structure that can transfer labeling information across heterogeneous domains particularly from text domain to image domain which has the advantage of reducing the issue of insufficient image training data by utilizing the labels in the text domain. They built multiple weekly shared layers of features which enable them to share the labels from text to image. Experiments on real world datasets show the competitive performance of the propose method. Motivated by their success in [75], the authors in [76] proposed a more Generative Deep Transfer Networks (DTNs) that are equipped with more generated layers and they use several parameter and representative-shared layers that helps to hierarchically learn to transfer the semantic knowledge from web texts to images. They used two SAEs to accept both text and images as inputs then followed by multiple generalized layers. They test their approach on their new datasets that is an extension of NUS-WIDE [77] that can be used for social image retrieval, multilabel image classification and cross-domain processing of image and text. With the high rise of 3D models available examining and understanding them is of great significance more especially in retrieval and classifications tasks.one of the clue for this kind of tasks is to get the features of 3D shape that can rightly describe both the shape and their parts. Many researchers utilized the different 3D data representations available to perform shape analysis task. A recent survey in [78] reviewed the most common architectures of deep neural networks which are: Convolution Neural Network (CNN), Autoencoder, Restricted Boltzmann Machine (RBM) and Long Short-Term Memory (LSTM). CNN which is most widely use deep neural network in computer vision contains many convolutional and subsampling layers which are sometimes follow by fully connected layers. The training example (x,y) with respect to the cost function in hidden layers are expressed as [78]:

$$J(W,b;x,y) = \frac{1}{2}\left\|h_{w,b}(x) - y\right\|^2 \tag{1}$$

For layer l, the error term δ equation as [78]:

$$\delta^{(1)} = ((W^{(l)})^T\delta^{(l+1)}) \cdot f'(z)^{(1)} \tag{2}$$

δ^{l+1} represent the error of the $(l+1)th$ layer of a network with cost function $J(W,b;x,y)$ while $f'(z)^{(1)}$ is the activation function derivative [78].

$$\nabla_{w^j} J(W,b;x,y) = \delta^{(l+1)}(a^{(l+1)})^T \tag{3}$$

$$\nabla_{b^{(j)}} J(W,b;x,y) = \delta^{(l+1)} \tag{4}$$

a serve as the input. while the error for sub-sampling layer is computed also as [78]

$$\delta_k^{(l)} = upsample((W_k^{(l)})^T \delta_k^{(l+1)}) \cdot f'(z_k^{(l)}) \tag{5}$$

Here k is the filter number in the layer. In [4], a voxel-based method was used to characterize a 3D shape and use a 3D CNN to the whole volume. References [79], [80] used the features describe on a manifold to execute CNN operations. But still using deep learning methods on 3D data is very challenging because of the complexity and geometric properties of the 3D models. The advantage of having many verities of 3D data representation has given researchers the opportunity to select the suitable data representations for their tasks. This section reviewed the performance of deep learning methods used in different 3D shape analysis tasks based on the representation of 3D data adopted. Figure 2 below illustrates the taxonomy of different 3D data representations currently in used in deep learning community.

A. PERFORMANCE OF DEEP LEARNING METHODS ON RAW DATA

Depending on the scanning device used to capture object of particular interest, raw 3D data can be obtained in different kinds. Range image can be obtained from UAV scans using different viewpoints and then used a registration process to combine them together to make a correspondence between them. The 3D point cloud can be obtained using lidar scanners to capture a scene, while RGB-D images can also be obtained using Kinect devices. Previously, Handcrafted descriptors signatures are used to extract features from the 3D objects or images. The success achieved by [81] in the image classification task now makes deep learning taking over of majority of computer vision tasks. While neural networks have been in existence since [82], the development of powerful computer hardware's like the GPU and the accessibility of large-scale datasets makes deep learning a success. In this section, we will cover the performance of deep learning methods on raw 3D data namely: Point Cloud, Projections and RGB-D including their strength and weakness.

1) PERFORMANCE OF DEEP LEARNING METHODS ON POINT CLOUDS

Point cloud is a special type of 3D data representation because of its irregular structure. Point cloud is uncomplicated and integrated data structures which are clear of complexities of mesh data. Previous methods for using point cloud are mostly hand crafted for a particular task. They are mostly categorized into intrinsic [83], [84] and extrinsic [85], [86] which are used as point features to encrypt some statistical information of points.

Reference [87] used point clouds as a combination of sets with distinct sizes. In this method, they show that when learning an underlying model, the orders in which input/output data are organized are important. They used read-write network to learn sorting numbers. Reference [88] Introduce a simple permutation equivariant layer for deep learning with set of structures. The layers are obtained by parametric sharing and have a linear time complexity in the size of each set. They successfully use deep permutation invariant networks to perform point cloud classification and MNIST- digit summation. Even though their network has a linear time complexity in the size of each set and it is comparatively simple, the performance of the network on ModelNet dataset was low. Reference [89] Proposed DeepSet that operate on sets and can also be used in different scenarios in both supervised and unsupervised task. The applicability of their method was demonstrated on population estimation, set expansion, outlier detection and point cloud classification.

Qi *et al.* in [17] proposed pointNet which is a network architecture that rightfully utilizes unordered point clouds and provides end-to-end classifications with less memory requirement than voxels grids or possible loss of information from 2D image representations. Reference [17] represents $\{x_1, x_2, \ldots x_n\}$ as unordered pointset and $x_i \in R^d, f : X \to R$ can be define as a set function that maps a vector to a set of points [17]:

$$f(x_1, x_2, \ldots, x_n) = \gamma \left(\underset{i=1,\ldots n}{MAX} \{h(x_i)\} \right) \tag{6}$$

where γ and h are the multi-layer perception (MLP) networks. pointNet does not use convolution operations to extract features as commonly used by other methods, instead it used fully connected layers to represent each point in a Euclidean space. It uses a spatial encoding for every point to combine into a global point signature for classification purpose. They use a Multi-layer Perception (MLPs) to generate features and combined them using max-pooling and single symmetric function. To help with the classification, they use the objects orientation in a canonical form to intensify the invariance to permutation of the input. PointNet demonstrates its robustness against input perturbation and partial data by producing comparable results in classification and segmentation task to the state-of-the-art methods as can be seen in table 2. A major challenge of pointNet is that it fails to capture local structures caused by the points occupied by the metric space. To overcome this limitation, the authors in [18] proposed pointNet++ which is a hierarchical neural network that uses the idea of 2D-CNNs where features are capture on a larger scale progressively through a multi-resolution hierarchy. Despite the fact that this network captures more features, it is very complicated and very low in computational speed. Nevertheless, PointNet++ performance on ModelNet 40 dataset is 90.7% which is a 2.7% higher than PointNet and also out performed [90].

Reference [91] proposed KD-Networks and avoids operating on the point clouds structure directly. Instead KD-Network uses multiplicative transformations based on the sub-divisions of the point clouds foist onto them by Kd-trees. The network is feed forward network and it demonstrate competitive performance in shape retrieval, classification and shape part segmentation tasks. Also, in [92], Roveri and Rahmann *et al.* avoided using point cloud directly

TABLE 1. A summary of 3D data representations.

3D Data Representation	Mode of Acquisition	Properties/Key features	Advantages	Limitations
Point Cloud	Point cloud can be obtained using Kinect [43] or structured light scanners [44]	Compact representation of 3D surface geometry	Ease of acquisition using any of the available sensors	May contains noise, missing parts, incompleteness and irregular sampling density
RGB-D DATA	Can be obtained using Microsoft Kinect [43]	Contains 2.5D information of captured 3D object with color and depth map (D)	Effective representation and very simple to work with	The flat 2D images of depth and color are not enough to learn the full geometry of 3D object.
3D Data Projections	3D data projections are obtained by projecting 3D data into 2D space	The projected data contains some essential features of the original 3D shape	The projected data is invariant to rotations at the main axis of the projection	Not suitable for complicated task like dense correspondence due to missing information from the projection [55].
Voxel	Can be obtained by modeling 3D data to describe the distribution of 3D object through the three dimensions of the scene	Contains encoded view point information of the 3D object	Offer full volumetric representation of 3D shape	Representing both occupied and non-occupied area of the 3D scene leads to huge demand for memory and voxels is not suitable for high resolution 3D object.
Octree	3D objects are models in a form of hierarchical data structure	is a simple varying size voxels	More efficient 3D volumetric representation and be used for high resolution 3D data [59]	The geometry of the 3D objects is not preserve in terms of intrinsic properties of the shapes.
Mesh	Can be created using 3D software packages like Maya, blender etc.	very irregular and complex 3D data representations with combination of vertices, edges and faces.	mostly used in computer graphics application for storing 3D objects and rendering purpose	Very difficult to learn because of the irregularity and complexity of mesh data.
3D data Descriptor	formed by defining features on 3D objects then concatenate and compute them to obtained a factor vector that describe the 3D shape	Describes topological characteristics of a 3D shape which is the signature of the 3D shape	It is used in many shape analysis tasks like shape retrieval, point matching etc.	Handcrafted low-level descriptors do not learn enough discriminative features from the 3D shape.
Graphs	3D meshes can also be extended to graph structured data in which the nodes of the graph are used as the vertices of the mesh and edges represents the connections between the vertices [68]	Graph based methods usually extract geometric meaning from a 3D shape	Models proposed for graphs can also be used for mesh data	Deep learning models proposed for meshes cannot be used for graphs.
Mixed	formed by combining more than one 3D data representations e.g. using a pair of multi view and 2D sketches to represent each 3D object	Carried both properties of the two data representations combined	Exploiting the advantages of both representations	Leads to complex network architecture.

by automatically transforming 3D unordered input data into a set of 2D depth images and successfully classified them by utilizing excellent performing image classification CNNs [93]. Their network consists of 3 key modules, the first module deals with input point cloud after learning k-directional views to create the depth maps, while the last two modules processed the already created k-depth maps for object classification. Previous deep networks that directly deal with points usually used a supervised learning approach. Yang *et al.* in [94] proposed FoldingNet a novel end-to-end deep auto-encoder to deals with unsupervised learning issues on point clouds. Their network used a graph-based enhancement on the encoder to make local structures on top of pointNet and a folding decoder successfully deforms a canonical 2D grid onto the underlying 3D object surface of a point cloud. They also used an SVM together with the FoldingNet which is used for 3D classification. The network performed well on ModelNet40 dataset which achieved high classification accuracy. Recently, Li *et al.* proposed So-Net [95] which used unsupervised model for a permutation invariant architecture with orderless point clouds. The key idea of So-Net is the building of Self Organizing Map (SOM) to models the spatial distribution of point clouds. The network represents the input point cloud by a single feature vector by using the SOM to execute hierarchical feature extraction on each points and SOM nodes. SO-Net archived higher classification accuracy on ModelNet 40 as can be seen in table 2.

Reference [96] propose RS-CNN a relation shape CNN which extends regular grid CNN to the irregular configuration of point cloud. The network learns from geometric topology constraint among individual points. Each local convolutional neighborhood is created by using a sample point x

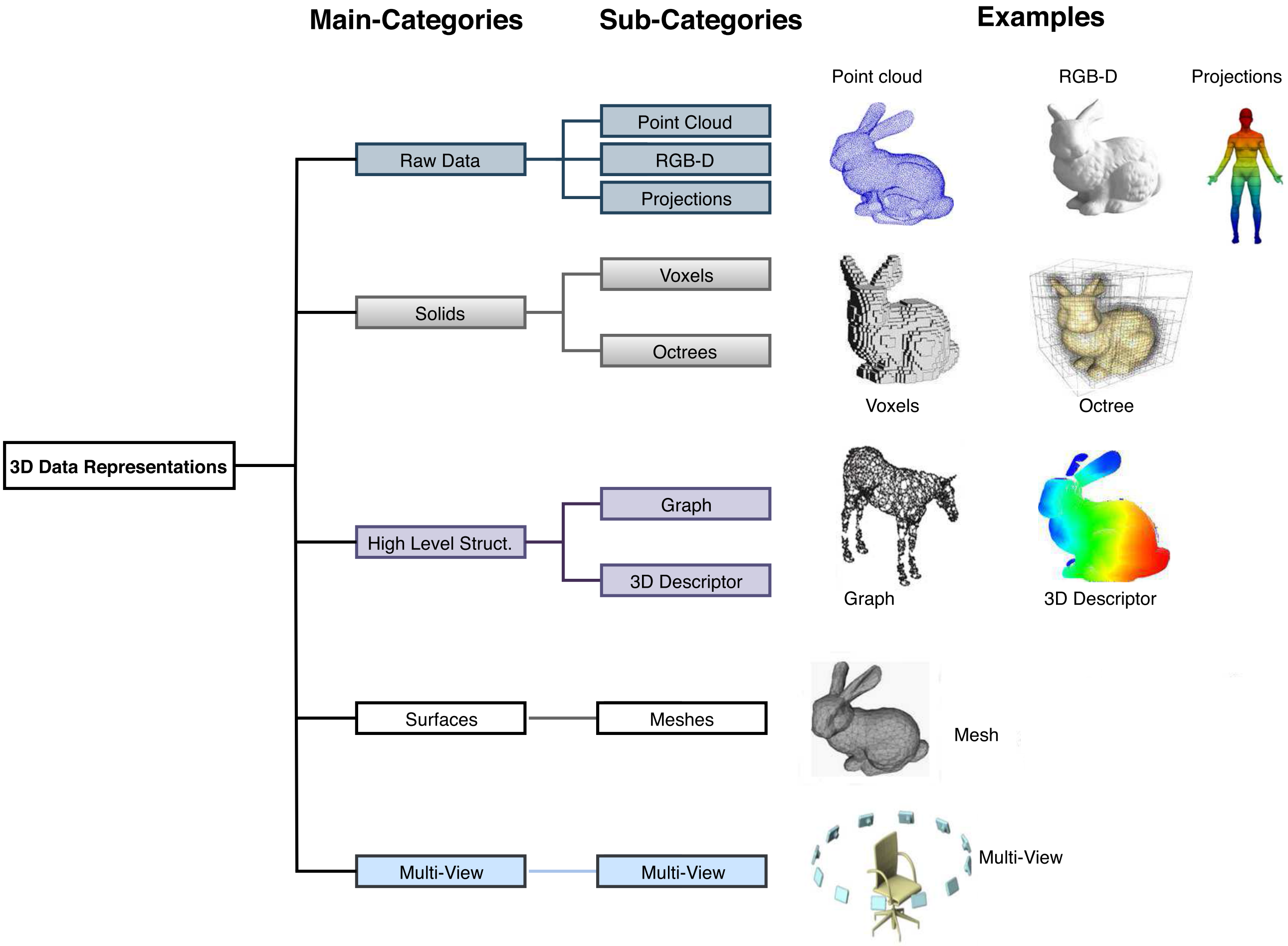

FIGURE 3. The taxonomy of different 3D data representations currently in used in deep learning community.

which the centroid then surrounding points as its neighbors. RS-CNN performed well on ModelNet40 which achieves a state-of-the-art performance as can be seen in table 2. In another work, [97] address the problem of 3D semantic segmentation of unstructured point clouds using a deep learning architecture by introducing grouping techniques that define point neighborhoods in the initial world space and the learned feature space. They use a dedicated loss functions to help structure the learned point feature space by defining the neighborhood in an adaptive manner which is very sensitive to the local geometry by utilizing k-means clustering on the input point cloud and then defining dynamic neighborhoods in the learned feature space using K-nearest neighbor (KNN). The effectiveness of the propose method was demonstrated on the task of semantic segmentation of 3D point clouds on indoor data from the standard 3D indoor scene dataset, ScanNet dataset and outdoor data from the virtual KITTI 3D dataset. Similarly, [98] propose PointSift which is similar to a SIFT. The module tries to encode information of variants of orientations which is adaptive to scale. Instead of using K-nearest neighbor as used in PointNet++, they obtain the information from all the points in the local neighborhood by combining the pointSIFT module on the PointNet++ architecture which demonstrate a high performance on segmentation task on ScanNet and Standard Large-Scale 3D indoor spaces datasets. In similar work, [99] proposed SPLANet that used unordered point cloud and incorporate a spatial convolution operator within the network structure. In this method, sparse bilateral convolutional layers are utilized that use indexing structures to perform convolutions only on the parts of the lattice been occupied. One of the major variations of SPLANet to PointNet++ is the use of flexible specification of the lattice structure to help hierarchical and spatially aware feature learning. SPLATnet have many advantages among which are: it accepts the input point cloud and use it directly without requiring any pre-processing to voxels or images and it allows an easy specification of filter neighborhood as in standard CNN architectures.

Reference [100] proposed to use ConvNets to recognize human actions from depth maps on a dataset based on Depth Motion Maps (DMMs). They employ three strategies to effectively utilized the capability of ConvNets in minimizing discriminative features for recognition. they start by rotating virtual cameras around subjects represented by the 3D

TABLE 2. Classification and mean average precision accuracies of some methods on ModelNet40 datasets.

Method	Representation	ModelNet40(Clssif. Acc.)	ModelNet40(mAP)	DeepModel
Johns et al. [119]	Multi views	90.7%	-	CNN
Bai et al. [69]	Multi views	83.1%	81.9%	GPU-CNN
Zanutigh et al. [121]	Multi views	87.8%	-	CNN
Wang et al. [9]	Multi views	93.8%	-	CNN
Qi et al. [5]	Multi views	91.4%	-	CNN
Asako et al .[14]	Multi views	97.3%	-	CNN
Yifan et al. [10]	Multi views	93.1%	85.7%	GVCNN
Xinwei et al. [133]	Multi views	-	88.0%	- TCL-CNN
Jianwen et al. [134]	Multi views	94.16%	92.8%	MLVCNN
Wu et al. [23]	Voxels	77.0%	49.2%	CDBN
Maturuna et al. [136]	Voxels	83.0%	92.0%	3DCNN
Xu and Todorovic [141]	Voxels	81.2%	-	CNN
Zhi et al. [52]	Voxels	88.9%	93.9%	volumetricCNN
Wang et al. [24]	Octree	89.9%	-	OCNN
Feng et al. [21]	Mesh	91.9%	81.9%	MeshNet
Shi et al. [8]	3D projections	77.6%	76.8%	CDBN
Sinha et al. [112]	3D projections	83.9%	51.3%	CNN
Sfikal et al. [114]	3D projections	95.5%	86.3%	CNN
Sfikal et al. [115]	3D projections	90.7%	83.5%	EnsembleCNN
Mengwei et al. [157]	Descriptor	90.5%	80.1%	3D-DDSD
Zhang et al. [178]	Graph	92.9%	-	LDG-CNN
Yifang et al. [80]	Graph	96.6%	-	HGCNN
Ben-Shabat et al. [187]	Mixed Data	91.4%	-	CDBN
Hedge and Zadeh.[189]	Mixed Data	90.8%	-	3DCNN
Ravanbakhsh et al. [88]	Point clouds	90%	85.7%	DBN
Zaheer et al. [89]	Point clouds	90.3%	88.0%	DBN
Qi et al. [17]	Point clouds	89.2%	92.8%	DBN
Klokov et al. [91]	Point clouds	91.8%	49.2%	KD-Network
Yang et al. [94]	Point clouds	88.4%	-	AE
Yongcheng et al. [96]	Point clouds	93.6%	-	RS-CNN
Feng et al. [104]	RGB-D	-	24%	CAEs-Ensemble

points of the captured depth maps in order to mimicked different viewpoints, then DMMs are encoded into Pseudo-RGB images which helps in turning the spatial-temporal motion patterns into textures and edges and finally through the use of transfer learning, the three ConvNets are trained independently on the color coded DMMS which are constructed in three Orthogonal planes. There proposed method was evaluated on MSR Action 3D, MSR action 3D Ext and UT-Kinect-Action datasets and achieved the state-of-the-art results on these datasets. As can be observed from the literatures reviewed in this section, the major challenges in raw point cloud are its non-uniform nature of the data which was usually caused by occlusions, sensor noise and distance from the sensors. Since point cloud has unordered structure researchers usually use a learning approach that is invariant to the order of the point cloud.

2) PERFORMANCE OF DEEP LEARNING METHODS ON RGB-D DATA

The availability of RGB-D datasets from RGB-D sensors e.g. Microsoft Kinect has motivated researchers to exploit RGB-D data representation due to the added advantage of color information and depth representation provided by the sensors. Socher *et al.* [101] were the first to use RGB-D data for 3D object classification. The authors used the integration of convolution and recursive neural networks to process depth channel and color independently. In the beginning, two single-layers CNN are used to extract low level descriptors from the RGB-D data. The output descriptor was forwarded to a set of Recursive Neural Networks (RNNs) that are initialized with random weights. The resultant descriptors from the RNN were merged to serve as input to a SoftMax classifier. This approach demonstrates good performance for house-hold object classifiers. Reference [102] also used a multi-scale CNN for semantic segmentation on indoor RGB-D scenes. The network use three different scales to process the input depth and RGB images then used the combined upsampled results which are forwarded to a classifier to obtain object class labels. The classifier predictions were merged with a super pixel's segmentation of the scene which is performed in parallel to get the final labeling of the scene. This method showed efficient and fast performance than the previous methods. However, its major limitation is the failure of the CNN to learn the geometry of the shape by focusing on learning only the class objects.

Motivated by the performance of [100], Eitel *et al.* in [103] proposed RGB-D architecture for object recognition that consist of two separate CNN processing for each representation and finally merging them with a late fusion network as can

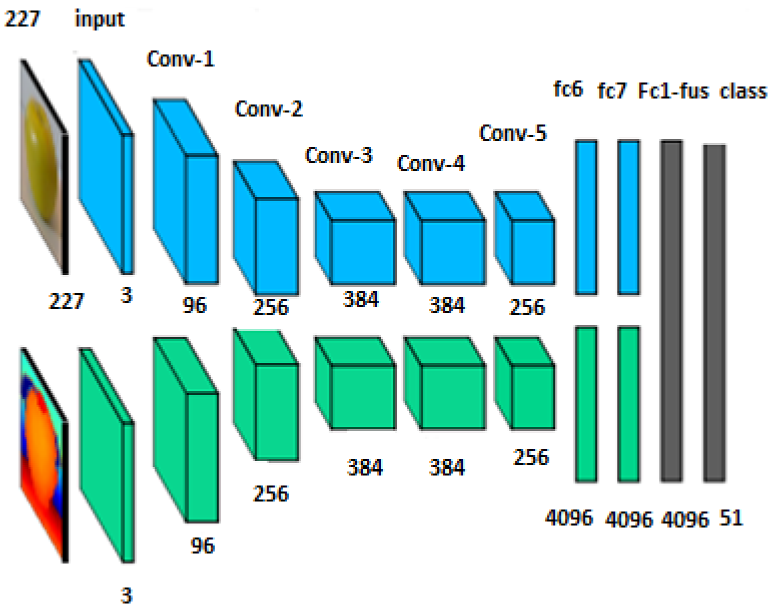

FIGURE 4. Object recognition Architecture for RGB-D data were two streams CNN are utilized [100].

be seen in figure 4. They avoided using the same network for more than one different learning task like segmentation and classification instead they used one CNN stream for processing RGB color information and the other stream for the depth. They trained each network separately and then fused in the results to the Fully Connected (FC) layers and the SoftMax for object classification which leads to promising results for object recognition task that outperformed existing methods. In [104], Feng *et al.* used Ensemble of auto-encoders for 3D shape retrieval. Each auto encoder was trained to learn a compressive representation of depth views using the Stochastic Gradient Descent algorithm with a large-scale CAD dataset. A likelihood score is derived as a similarity measure by viewing each auto-encoder as a probability model. They also used Domain Adaptive layer (DAL) to receive the output score of the auto-encoders to rank the retrieved scenes. This method demonstrates increased performance when compared with other similar methods.

Reference [105] proposed an approach that utilize multi-scale CNN that is trained from raw pixels and use it to extract dense feature vectors which encode regions of multiple sizes centered on each pixel. They also propose a technique to automatically obtains from a pool of segmented components, a best set of components that thoroughly explain the scenes. There method achieved a record accuracy on the SIFT flow dataset, Barcelona dataset and on standard background dataset. The major limitation of this approach is that by stacking a depth channel onto an existing CNN architecture leads to less exploitation of the full geometry information encoded in the depth channel. Gupta *et al.* exploits stacking a depth channel on existing CNN architecture by representing each pixel in terms of horizontal disparity pixel height above ground and angle between normal and gravity. They stacked these three computations into a three-channel image. In the pipeline, a structured forest approach was first implemented to get the contours from the RGB-D image which is used to generate region proposals in the depth channel of the image using a RF regressor at the same time, the RGB channels of the region proposal are handle by a CNN for feature extraction then SVM classifier is used to processed both the RGB and depth channels features. In [106], a long-Short term memory (LTSM) fusion on top of multiple convolutional layers to fuse RGB and depth data are proposed. In this method, many convolutional layers and a long short-term memory layer are stacked together to obtain depth channels and photometric features. Both the long-range spatial dependencies and short-range are encoded by the memory layer in an image along the vertical direction. The proposed method achieved a state-of-the-art performance on the large-scale SUN RGBD dataset and the NYUDV2 dataset.

The concept of transfer learning and CNNs are combined to train 4 CNN individually by Alexandre [107]. The four individual CNN are train in sequence against training them in parallel and then using the weights of an already trained CNN as a beginning point to train other CNNs that will process the remaining channels. This approach saves training time and also further improves the recognition accuracy based on the experiments carried out. Schwarz *et al.* [108] also delve into the transfer learning idea for object classification. In this model, they use a canonical perspective to rendered RGB-D data to obtain depth color because of the distance from the object center. The CNN used in this method are pre-trained CNNs for object categorization and the features obtained by the CNN are forwarded to SVM to decide object class and pose. Inspired by the success of [104], Zhu *et al.* [109] proposed to learn a robust domain invariant representation between 3D shape and depth image domain by creating a pair of distance neural network for every domain. The networks are connected via a loss function with a restriction on the inter-class and intra-class margin that helps to reduce the intra-class variance and still maximizing the inter-lass margin between the data from the two domains i.e. depth image and 3D shape. This method produces superior results over existing state of the art methods based on depth image 3D shape retrieval.

Learning RGB-D data using deep learning approach have demonstrated effective performance. Furthermore, the extra depth representation been provided by the RGB-D sensors on top of the standard RGB-channels allows researchers to treat the depth channels and the colors individually even though others utilized only the depth information for their systems. The major disadvantages for this kind of data is that sometimes the data might be noisy and incomplete capture data which makes them difficult to use in complex situations. There is also the issue of lack of learning the full geometry of the 3D object this motivates many researchers to exploit the volumetric representations of the 3D shape as discussed in section C.

3) PERFORMANCE OF DEEP LEARNING METHODS ON 3D DATA PROJECTIONS

Representing a 3D shape by using a number of 2D projections rendered from many directions is also used in many 3D shape analysis tasks. Stereographic projection can be described as a special mapping that directly projects a sphere on to a plane. Early works on this direction is the approach of Zhu *et al.* [110] which learn the features of a 3D shape by projecting them into 2D planes. A global deep representation of a 3D object was obtained by using an Auto encoder for the task of 3D object retrieval. Initially each 3D shape undergoes a series of scaling, translation and pose normalization then a set of 2D projections were applied on already processed 3D object and then forward to a stack of RBMs to obtain the features for all projections. [110] also introduce the pre-training procedure for binary units and generalize to real valued by showing that the pixels correspond to the visible units because their states can be observed while the feature detectors correspond to the hidden units. they then defined a joint configuration (v, h) for both the visible and hidden units [110] as

$$E(v, h) = -\sum_{i \epsilon visible} a_i v_i - \sum_{j \epsilon hidden} b_j h_j - \sum_{i,j} w_{i,j} v_i h_j, \quad (7)$$

v_i, h_j represents the binary states of both the hidden units and visible units of i and j while a_i, b_j denote their biases and finally wij the weight between them. Experiments showed that the proposed method performed better compared to global descriptors-based approaches. This is due to the combination of local descriptor with the global representation which results in a good performance.

Leng *et al.* [111] also used an AE for 3D object retrieval. In this method, an extension of the normal AE i.e. Stacked Local Convolutional Auto-Encoder (SLCAE) was used. They exploit multiple depths images of different views of 3D shape as input to the AE then trained each layer of the architecture with gradient decent method. The method was tested on many standard datasets with promising results. In [8], Shi *et al.* proposed DeepPano which extract 2D panoramic views of 3D shape using a cylindrical projector in the principal axis of the 3D shape. To make the learned features Invariant to rotations they train the model architecture with 2D classical CNN and use a row-wise max pooling layer in between the convolution layer and the fully connected layers. This method was tested on 3D object retrieval and recognition task which demonstrate effective performance in comparisons with previous methods. In [112], Sinha *et al.* convert a 3D object into a geometry image and use CNNs to learn 3D shapes. Rotation, scaling, data augmentation is carried out as pre-processing step to have more training data. This method uses authelic parameterization to create planar parameterization on a spherical domain which helps to learn the 3D object surfaces. This framework was tested on standard 3D shape datasets like ModelNet 10, ModelNet 40, McGill11 and SHREC1 which achieved a higher performance in comparisons to state of the art.

Motivated by the success of projections methods, in [113], Cao *et al.* projects a 3D object onto a spherical domain centered on its barycenter then classify the spherical projections using a neural network. To successfully captures the 3D features, they used two complementary projections with the first capturing the depth variations of the 3D object and the second one dealing with the contour information fix in different projections from different angles. This method produces comparable results on 3D object classification tasks on different standard datasets. In [114], Sfikal *et al.* extracted 3D objects normalized pose and then represent them as panoramic views. They used SYMPAN method to pose normalized the 3D model after the panorama representation is extracted then used it to train CNN network by utilizing the augmented view of the extracted panoramic representation views. This method performed well in standard large-scale dataset on classification and retrieval task. The method in [115] is extensions of their previous method where they used an ensemble of CNNs for the learning task which leads to impressive performance in ModelNet10 and ModelNet40. Projections based approaches proved to be effective for learning 3D shapes more especially by exploiting the deep learning methods. To overcome the loss of geometric properties of the shapes during projections, [112] used many projections representations to recover the loss data. There is a lot of benefits of using 2D deep learning models directly on Projections data but it still requires much fine-tuning.

B. PERFORMANCE OF DEEP LEARNING METHODS ON MULTI-VIEW DATA

Exploiting multi-view data of 3D objects have shown that building 3D objects descriptors directly on 3D data might not be the best approach to solving 3D shape retrieval and

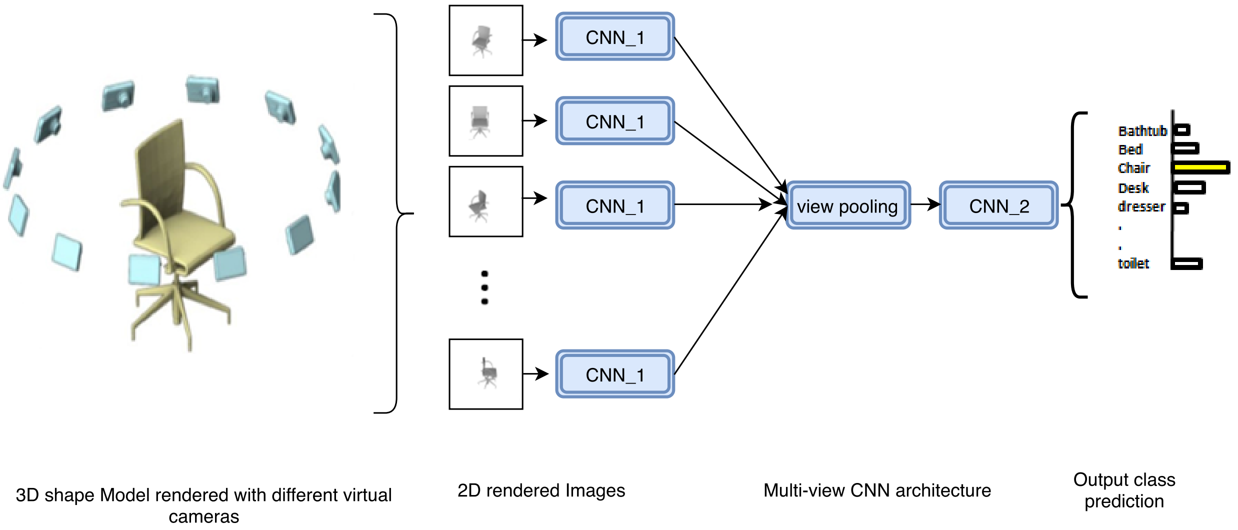

FIGURE 5. **Multi-view CNN a novel architecture that combines information from multiple views of a 3D shape into a single shape descriptor [70].**

classification tasks. Despite the efficiency of the volumetric deep learning methods, most of this approach requires huge computational resources due to the used of convolutional filters to obtain the features. These motivate researchers to exploit the multi-view data representation which has the benefit of utilizing the standard 2D deep learning models. Multi-view CNNs consist of several of 2D rendered views of a 3D shape that are used to obtained the viewpoints by utilizing the normal rendering devices. Some of the key reasons why multi-view CNN performed better than the volumetric CNNs are: High resolution 3D data can be used because of low memory requirement, availability of large training dataset and the use of standard 2D deep learning models. Leng *et al.* [116] were among the first researchers to exploit 2D deep learning models to learn multi-view 3D data. In this method, high level features of 3D object were extracted on a number of view-based depth images using deep belief network. They adopted a contractive divergence method to train the network in a layer-wise approach. This method performed better than [90] that use composite descriptors. In [117], Xie *et al.* adopted the multi-view depth image representation and proposed Multi-View Deep Extreme Learning Machine (MVD-ELM) that achieved rapid and qualitative projective feature learning for 3D models. This method uses 20 multi-view depth images that are captured uniformly at the center of each 3D shape using a sphere. in ELM feature mapping, an input data $x \epsilon R^D$, the generalized output function of a single hidden layer feedforward neural network for ELM is given by as in [117]

$$f(x) = \sum_{k}^{i=1} \beta_i h_i(x) = h(x)\beta, \qquad (8)$$

$h(x) = [h_1(x), \ldots h_k(x)]$ represent the output vector of the hidden layer while $\beta = [\beta, \ldots \beta_k]^T \epsilon R^{KXM}$ is the output weights vector [117], while in ELM learning $T \epsilon R^{NXM}$ is denoted as the target matrix supplied by the training data N. The N random feature maps contains in $[h(x_1), \ldots, h(x_N)]^T \epsilon R^{NXK}$ are obtained in the initial stage and the weighted sum of the training error and the norm of output weights is minimize as the objective function [117]

$$w \, \|H\beta - T\|_2^2 + \|\beta\|_2^2 \qquad (9)$$

using the closed form solution, β can be obtained [117] as:

$$\beta = \begin{cases} (H^T H + \frac{1}{w} I)^1 H^T T, & L \leq k, \\ H^T (H H^T + \frac{1}{w} I)^{-1} T, & L \geq k, \end{cases} \qquad (10)$$

I represent the identity matrix. in the case of multi view feature mapping, they represent the input data as $N = MXD$ depth images that are denoted by a matrix of $(dXdXD)$ which is an array of N depth images of dXd resolution [117]. Since the MVD-ELM consist of layers of convolution and pooling operations, for a given layer L, the random convolution for the normalized kernels is given by [117]

$$W_l = [w_{l,k}]_{k=1}^{k_l} \subset R^{c_l X c_l X K_l}, \quad l = 1, \ldots, L \qquad (11)$$

which include K_l convolution kernels $w_{l,k}$ of size $c_l X c_l$ and the $k - th$ normalized random convolution kernel is obtained as [117]: $W_{l,k}(i,j) = rand(0,1), i,j = 1, \ldots, c_l$

$$W_{l,k}(i,j) = W_{l,k}(i,j) / \sum_{i,j} (W_l, k(i,j)) \qquad (12)$$

$rand(0, 1)$ generates a random number in [0,1]. Also, the $k - th$ feature map for any view n at layer l can be obtained [117] as:

$$F_{l,k,n} = (F_{l-1,k,n} * W_{l,k}) \bigotimes m_{l,n}, \quad n = 1, \ldots, D \qquad (13)$$

The $*$ is convolution operation while $\bigotimes$ is the multiplication of the element-wise that put in the foreground mask $m_{l,n}$ that remove the background. while for multi-view pooling, they use the average pooling. For the 1-th layer, the size of the pooling is taking as s_l that cause the pooling maps of size $d_l/s_lXd_l/s_l$ [117]. The average pooling is then applied to obtained the $k-th$ pooling map for layer 1 and view n as [117]

$$P_{l,k,n}(p,q) = \frac{1}{s_l^2}\sum_{i=(p-1)*sl+1}^{p*sl}\sum_{j=(p-1)*sl+1}^{p*sl} F_{l,k,n}(i,j). \tag{14}$$

$p, q = 1, \ldots, s_l$ where $P_{l,k,n}$ is the next layer input feature map. This method was extended to fully convolutional (FC-MVD-ELM) which was used for 3D segmentation by training the multi-view depth images and then projected the predicted labels obtain during training back to the 3D object then smoothed the final result using the graph cut optimization method. Both the two methods are tested on 3D shape segmentation and classification tasks and the results significantly outperformed [23] with less processing time.

Leng *et al.* carried out more research in an effort to exploit deep learning models on multi-view 3D data and came-up with the extension of classical auto-encoders in [111]. In this work, 3D object retrieval approach was proposed based on Stacked Local Convolutional Auto-Encoder (SLCAE). The SLCAE was trained using greedy layer-wise strategy and then use gradient descent to train each layer to obtain the representative of input data which is the feature of 3D object. The results of the experiments conducted demonstrate that the approach significantly improved on 3D shape retrieval compared with several state-of-the-art methods on PSB, SHREC 09 and NTU datasets. Reference [118] proposed a 3D CNN to deals with a number of 2D views of the 3D object where a Stochastic Gradient Descent (SGD) was used to pre-train the convolutional layer and then employ a back propagation to fine-tune the whole network and finally used the results of the two phases for 3D shape retrieval. This method also outperformed the state-of-the-art methods based on the experiments carried out on public available 3D datasets.

In [70], the novel multi-view CNN (MV-CNN) was proposed by Su *et al.* for 3D shape retrieval and classification tasks. In this method, there is no specific order use in view pooling layer to process multiple views of 3D objects which is not the case in [103]. Multi-view CNN rendered 12 and 80 virtual views for the object and used the image stacks to serve as the input of CNN which was pre-trained on ImageNet 1k dataset and fine-tune on ModelNet40 [4]. For the shape retrieval part, the fully connected layer of the network which is the seventh layer serves as shape descriptor. They defined a distance between shape x with n_x image descriptors and shape y with n_y image descriptors as [70]

$$d(x,y) = \frac{\sum_j min_i \left\|x_i - y_j\right\|_2}{2n_y} + \frac{\sum_i min_j \left\|x_i - y_j\right\|_2}{2n_x} \tag{15}$$

they represent the distance between a 2D image x_i and a 3D shape y as $(x,y) = min_j \left\|x_i - y_j\right\|_2$ then they compute the average of all n_x distances between $x's$ 2D projections and y. Experiments evaluation on shape retrieval and classification showed that multi-view CNN outperformed all other previous methods tested and also the shape descriptor outperformed the state of the art 3D ShapeNet [4] with a wide margin in the retrieval task on ModelNet40 dataset figure 5 shows the architecture of multi view CNN. Johns *et al.* [119] use a different approach to utilize the views of a 3D object using camera trajectories. The views are arranged in pairs together with their relative pose and then supply to a CNN. Each pair are classified separately and the final result is obtained using the weight contribution of each pair. The VGG-M architecture Chatfield *et al.* [120] was employed which consists of three fully connected layers and five convolutional layers. It can accept depth image, gray-scale or both as input. This method performed better than ShapeNet [23] and [70].

The impressive performance of multi-view deep learning architectures pushes researchers to carry out more work on GPU based approaches to learn multi-view 3D data. Bai *et al.* in [69] propose a multi-2D views real time GPU CNN search engine. The method is called GIFT and use two inverted files in which one is use for matching and the other for ranking the initials results. This method was tested on many standard 3D datasets like ModelNet, PSB, SHREC 14, McGill and many others and produced more qualitative results than the state-of-the-art methods. More research work to exploit multi-view 3D data was carried out. Zanuttigh and Minto in [121] used a multi-branch CNN to classify different 3D objects. In this work, the input consists of a rendered depth maps from different point of views of the 3D object and five convolutional layers for each CNN branch to process each depth maps to produce a class file vector. These vectors are then supply to a linear classifier to indicate the class of the 3D object. In [9], Wang *et al.* proposed the view clustering and employ pooling layers on the dormant sets. The main idea here is to pool views that are similar and cluster them recursively to build a pooled feature vector and then forward this feature vectors which serve as input to the same layer in a recurrent training approach. This network performed effectively on 3D shape recognition task with a higher performance than the state of the art [23], [70]. Based on the perception that multi-view deep neural networks perform better than the ones utilizing the full 3D information of 3D shapes, Qi *et al.* [5] carry out extensive study and compare volumetric and multi-view CNNs for object classification. In this study, sphere rendering based on multi-resolution 3D filtering is used to get information from multiple scales for the multi-view CNN were proposed and using combination with training data augmentation achieved enhancement of multi-view CNN on ModelNet 40. The multi-view approaches proved to perform better compared to the volumetric counterparts with less computational requirement need. Nevertheless, this kind of representation still has some challenges which include lack of preserving the intrinsic geometric features of a 3D object and the needs to have sufficient number of views.

More research work to improve on the performance of multi-view-based data representation was carried out and in [14], Kanezaki *et al.* proposed RotationNet which utilizes the multi-view images of an object as input and together estimate both the pose and object category. Different from previous methods that use known view point labels for training, this method deals with the view point labels as latent variables. The network use only a partial set of multi-view images for inference. The performance of this method is demonstrated by its higher performance on ModelNet datasets. Against the view to shape setting normally adopted by many methods, Feng *et al.* [10] propose group view CNN (GVCNN) in order to exploit the intrinsic hierarchical correlation and discrimination among views. This model consists of a hierarchical view group shape architecture which is organized into view level, group level and shape level that are re-arranged using a grouping strategy. Initially, view level descriptor was first extracted using an expanded CNN, then a group module was used to estimate the content discrimination of each view in order to split all views into different categories based on their discriminative level and pooling from view descriptors was done to further generate a group level descriptor and finally combined all group level descriptors into the shape level descriptor based on their discriminative weights. This approach achieved a robust performance on 3D shape retrieval and classification tasks on ModelNet40 dataset.

References [122], [123] extensively discusses 3D model retrieval methods which they classified into model based and view based methods. At the beginning, majority of the approaches used model-based methods and deals directly with a 3D model data for retrieval. Examples of this methods are (geometric moments [124], surface distribution [125], volumetric distribution [126], surface geometries [127] among others. Similarly, Osadi *et al.* [127] utilized the shape feature of 3D models by constructing the distribution of the shape that is sampled from the 3D model as the digital signature of an object which was used to calculate the similarity between different models. Most of the model-based methods require 3D model reconstruction which requires high computational costs. Recently, view based methods have been used in 3D model retrieval in [128]–[130] and have the advantages of not requiring 3D model reconstruction [122]. In view-based methods, many visual features are extracted from multiple views of a single 3D object that are represented by a set of 2D images captured from different views such as HOG descriptors [131], Zernike moments [132] among others. In [122] a Multi-modal clique graph (MCG) matching method was propose that systematically generates MCG that used cliques and contains neighborhood nodes in multi-modal feature space together with hyper-edges that link pairwise cliques. In this method, the issue of set-to-set distance measure was addressed using an image set-based clique/edgewise similarity measure which is central difficulty in MCG matching. By representing an individual 3D model with multi-view and multi-modal information their MCG preserve both the local and global structure of a graph. The proposed MCG matching method was applied to view-based 3D model retrieval which is evaluated extensively using three popular single-modal datasets; NTU, PSB, ETH datasets and a novel multi-view RGB-D object dataset (MV-RED) which was contributed by the authors.

Reference [122] list three main steps of view-based methods which include selection of representative view through clustering, the measurement of similarity based on distance metrics or probabilistic models and the generation of a ranking list based on the computed similarity measure. The authors proposed a discriminative multi-view latent variable model (MVLVM) for 3D object retrieval where they regarded each individual 3D model as a square of ordered 2D images capture from multiple views then they used an undirected graph model with latent variables to automatically find the context among multi-view images in both the spatial and feature domains. Each node in the graph denotes a latent representative view and each edge represents the latent spatial context between the corresponding latent representative views which helps to get a joint distribution over similarity labels and latent variables given the observed features obtained from multiple views by using an undirected graph model to learn the dependencies among the latent variables. There proposed method was evaluated on single modal datasets (NTU and ITI datasets) and a multi-modal dataset (MVRED-RGB and MVRED-DEPTH) which shows the superiority of the proposed method.

In an attempt to study variants of deep metric learning losses for 3D object retrieval, [133] proposed two kinds of representative losses which are center loss and triplet loss to learn more discriminative features than the normal traditional classification loss for 3D shape retrieval. The authors also propose the normal loss to the triplet center loss, this loss learns a center for each class and minimizes the distance between samples and centers from the same class. This method was tested on ModelNet40 dataset and outperformed [68]. Recently, Jiang *et al.* [134] proposed a multi-loop view CNN (MLVCNN) for 3D object retrieval which used different loop directions to extract a number of views and introduce a hierarchical view loop shape architecture that consist of view level, loop level and shape level in order to carry out 3D shape representation for different scales. A CNN is trained to extract view features in the view level, and then the loop level features are generated using the proposed loop normalization and LSTM for each loop of view considering the intrinsic associations of different loops in the same loop. Finally, all the loop level descriptors are combined into a shape level descriptor for 3D shape representation which is used for 3D shape retrieval. This method was evaluated on ModelNet40 dataset and outperformed the state-of-the-art methods in 3D shape retrieval task as can be seen in table 2.

C. PERFORMANCE OF DEEP LEARNING METHODS ON SOLIDS

Two major representations of volumetric 3D shapes are Octree and Voxels i.e. representation of 3D in three-dimensional space as a regular grid. Volumetric 3D data representation usually encodes a 3D object as a 3D tensor

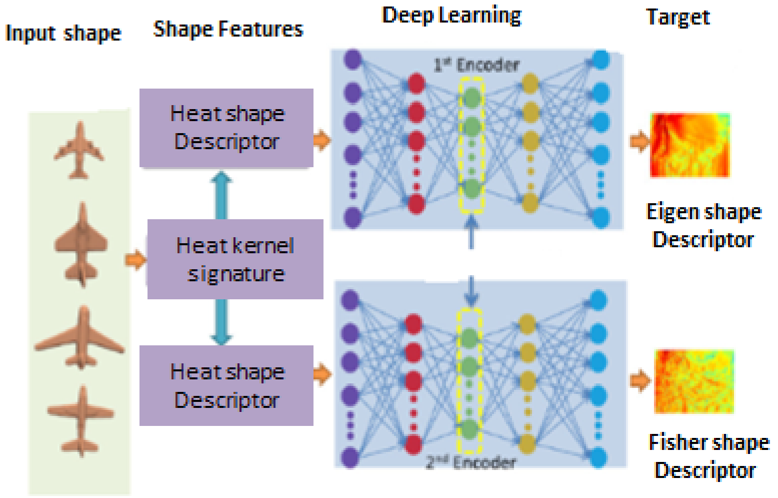

FIGURE 6. ShapeNet Architecture were the input is represented as a 30 × 30 × 30 binary tensor [23].

of binary or real values. Voxel representation characterize 3D object by describing its distribution through the three dimensions of the scene. A 3D convolution accepts a shape w,h,l as input, then use a kernel k, k, d and delivers a shape w, h, m. The depth is increase by a set of strides after the convolution of each 2D plane. Given $C(n, d, f)$ as an operator for 3D convolution for which $nxnxn$ is the input and d as the size of the feature maps f × f × f. At position x, y, z the output on $m - th$ feature map of layer l as in [39] is:

$$v_{lm}^{x,y,z} = b_{lm} + \sum_{q} \sum_{i=0}^{f-1} \sum_{j=0}^{f-1} \sum_{k=0}^{f-1} w_{ijk}^{lmq} v_{(l-1)q}^{(x+1)(y+j)(z+k)} \tag{16}$$

b_{lm} represents the layer bias and w_{ijk}^{lmq} are the weights at position i, j, k of the kernel at the $5 - th$ feature map while q pass through the feature maps in the $l - 1th$ layer [39]. The major limitation of voxels based method is their unnecessary demand of computer storage because of its ability to represents both non-occupied and occupied parts of the scene which makes it not suitable for high resolution 3D objects [57], [63].

Octree which is also a volumetric representation of the 3D object is more efficient and it is a varying size voxel. Octree data representation was recently utilized in conjunction with CNN for 3D shape analysis tasks [24], [58] and prove to be effective for high resolution input [59]. In this section, we will explore the performance of deep learning methods on these two volumetric 3D data representations.

1) PERFORMANCE OF DEEP LEARNING METHODS ON VOXELS DATA

Wu *et al.* in [23] are one of the first deep learning methods to utilize the voxels 3D data representation. In this method, a 30 × 30 × 30 binary tensor represent the object which is used as the input to indicate if the voxel belongs to the 3D object or not as shown in figure 6. Also [135] adopted the concept of a Convolutional Deep Belief Network (CDBN) used in 2D deep learning to characterize the 3D model. The CDBN framework also use convolution to minimize the number of parameters because of its ability to share weights. This property gives CDBNs the ability to learn the joint probability distribution of voxels by representing varieties of object categories with a few parameters. The ShapeNet network consist of one input layer, one output layer and three convolution layers which in total makes five layers which were initially pre-trained in a layer wise manner and Contrastive Divergence method was used to train the first four layers while the last layer was trained using Fast Persistent Contrastive Divergence. A single depth map was used as the input during testing stage which represents the 3D object and then converted to a voxel grid representation. The network was tested on retrieval and classification, view-based view prediction. The major constrained in this method include the additional dimension in the convolutional kernel which leads to computationally unmanageable model that is difficult to process high resolution input and there is the issue of very hard learning process because the network was trained on isolated view of fixed sized with minimum information. Nevertheless, the network performs better despite the fact that it is operating on low resolution voxels.

VoxNet was proposed by Maturana and Scherer in [136]. In this method, the concept of 3D convolution was employed on 3D object recognition task on different of 3D data representation which include RGB-D, 3D CAD models, and Lidar point clouds. The key difference of convolution used in VoxNet with the 2D convolution is the filter. A 3D filter replaces the 2D filter and the network consist of two FC layers, one pooling layer, two Convolution layers and one input layer. A volumetric occupancy grid of 32 × 32 × 32 voxels was used as the input data and then supply to the network which they used Stochastic Gradient Descent (SGD) to train with a momentum. Extensive experiments demonstrate that VoxNet outperformed [23] in NYUv2, ModelNet datasets when the network was trained from scratch for classification task but [23] outperformed the VoxNet on NYUv2 if the pre-trained model of ModelNet10 is employed. In [137], inspired by the impressive performance of [136], Seaghat *et al.* improved the architecture of VoxNet to accommodate orientation of 3D object during the learning process which helps improve the classification performance on ModelNet 10 dataset.

Unsupervised learning approach was used in [138]. In this method, 3D GAN was proposed in which probabilistic latent space was used to tactically learn the features of 3D object by employing the adversarial discriminator which was used to model 3D objects then create synthetic data. Voxception-ResNet was proposed by Brock *et al.* [139]. In this method, a 2D very deep model was adopted on ModelNet10 and ModelNet40 datasets for classification task. The authors very deep model (VRN) that depends on the architecture of [93] and [140] and also used the same batch normalization of [93]. This network required data augmentation for training and consist of 45 deep layers. This network is also similar to VoxNet because of the presence of 3D filters with ConvNet. The VRN model demonstrates effective performance on ModelNet datasets in classification task which is a state-of-the-art result. In spite of the efficiency of this method, there is a likelihood of over fitting problem if a large amount of data augmentation is not provided which usually leads to small dataset of a deep architecture.

The Beam search model for learning the optimal 3D CNN architecture was proposed by Xu and Todorovic in [141]. This model performs 3D classification on ModelNet 40 dataset and successfully indicates number of layers of 3D CNN, number of nodes, training parameters and connectivity. It consists of one FC layer and two Conv. layers which are the starting points of this architecture and then increase to building 3D CNN optimal model through adding a new Conv. Filter. The training of the network was similar to [135] where they also adopted the Contrastive Divergence method to train Conv. Layers and then used Fast Persistent Contrastive Divergence to train the FC layers. After one layer was successfully learned, its weights are fixed and parameters of the activation are sent to the next layer. The network produced effective results on ModelNet 40 dataset for classification task. Deep Sliding shapes was proposed by Song and Xiao in [142] in order to learn the features of 3D objects at different scales by utilizing 3D CNNs for object classification and recognition task on ModelNet dataset. In this method, depth maps of RGB-D scenes are converted to 3D voxels by exploiting a directional Truncated Signed Distance Function (TSDF). The TSDF representation gave this model the advantage of learning the geometry of the 3D object which is an alternative to using depth map directly. This model demonstrates effective performance on NYUv2 dataset on various object classes for object detection task.

Even though volumetric 3D models prove to be effective, most of the existing architectures needs large amount of computational resources because of convolution operation and the huge number of parameters. Due to this major constrains, Zhi *et al.* [62] proposed a real time volumetric CNN for 3D object recognition task which is termed LightNet. The network architecture has two major capabilities which are: the use of multi-tasking to learn a lot of features at the same time and its ability for fast convergence with fewer parameters by adding the batch normalization operation between both the activation and convolution operations. The network was tested on ModelNet datasets and it outperformed [136] by 24% in both ModelNet10 and ModeleNe40 datasets for classifications tasks. Recently, Wang *et al.* [143] propose NormalNet which is a voxel-based CNN for 3D shape retrieval and classification task. In this method, normal vectors of the object surfaces are used as input instead of binary voxels. The authors propose a Reflection Convolution Concatenation (RCC) module for extracting clear features for 3D vision tasks and at the same time minimizing the number of parameters. The performance of NormalNet was significantly improved by combining two networks that accept normal vectors and voxels as input respectively. The network achieves a competitive performance on 3D shape retrieval and classification task on ModelNet10 and ModelNet40 datasets.

2) PERFORMANCE OF DEEP LEARNING METHODS ON OCTREE DATA

Octree data representation serve as a central method used in many computer graphics applications such as shape reconstruction, rendering, collision detection and 3D shape analysis tasks. Recently due to its GPU implementation friendly approach, some methods start to use it for many shape analysis tasks. In [144], Häne *et al.* proposed a hierarchical surface prediction (HSP) which aid high resolution voxel grid prediction. The interior and exterior of the object are represented by a course resolution voxel. Tatarchenko and Dosovitskiy [57] used octree representation to generate volumetric 3D output in a memory efficient manner by utilizing deep convolutional decoder architecture. This method can predict the occupancy values of individual cells and the structure of the octree which makes it possible for higher resolution output with minimum amount of memory.

Reference [58] proposed OctNet which is a representation for deep learning with sparse 3D data. In this method, the sparsely of the input data was exploited to hierarchically partition the space by utilizing a set of unbalanced octrees that used each leaf node to stores pooled features representation. OctNet was build based on the observation that 3D data is usually sparse in nature like point cloud, meshes etc. which leads to an unwanted computation when using 3D convolutions this motivated the authors to exploit the sparsely property of this data representation. OctNet was trained on [23] for 3D shape classification task. OctNet showed the advantages of octree-based decoder on depth fusion, 1283 2563 and even higher resolutions output could be achieved by octree-based decoders.

Motivated by the success of OctNet, Wang *et al.* [24] proposed O-CNN which is an Octree-based Convolutional Neural Network (O-CNN) for object classification, retrieval and segmentation tasks. The O-CNN was designed using a modest octree structure that supports all CNN operations on the GPU. The average normal vector of a 3D shape was used as input and the features of the finest level octants are then computed. A pooling operation was performed to downsample to the parent octants at next coarser level which is then supplied to the next CNN layer, and this operation repeated continuously until all the layers are evaluated. Repeated convolution and pooling operations are performed on the octree data structure from bottom to top and a Relu function was used for output activation and finally batch normalization was used to reduce internal covariance shift. 3D convolution is applied efficiently to an octant by picking its neighboring octants at the same depth and the convolution operator Φ_c is represented in the unrolled form as in [24]

$$\Phi(O) = \sum_n \sum_i \sum_j \sum_k W^n_{i,j,k} \cdot T^{(n)}(O_{ijk}). \tag{17}$$

where O_ijk represents a neighboring octant of O and and the feature vector represented by $T(.)$.O-CNN was tested on SHREC16 dataset for shape retrieval task and achieved comparable results to state of the art. Moved by the success of O-CNN, the authors in [12] proposed Adaptive O-CNN (AOCNN) which represents 3D shapes adaptively with octants at different levels, and the 3D shape was models within each octant with a planar patch. In this method, an adaptive encoder and decoder for encoding and decoding 3D shapes were proposed. The encoder is used to receive the planar patch normal with displacement as it is input and then execute 3D convolutions at the octants at each level only while the decoder read the shape occupancy and the status of the sub-division at each level to estimate the best plane normal and displacement for each leaf octant. AOCNN was validated on effectiveness on 3D auto-encoding, shape prediction, shape classification and shape completion for noisy and incomplete dataset with a classification accuracy better than [17], [58] and [24]. AOCNN have two major limitations which are: one, lack of seamless in the adjacent patches at the adaptive octrees and secondly, the planar patch used does not approximate curved features very well.

D. PERFORMANCE OF DEEP LEARNING METHODS ON SURFACES

Surface mesh is one of the most significant and powerful 3D shape representation and it is widely used in computer graphics. However, due to the irregularity and complexity of mesh data not much deep learning work has been done using this data representation for 3D shape analysis tasks as can be seen in figure 1 which shows the progress of 3D data representations along with time used for 3D shape analysis tasks from 1999 to 2019 based on the literature's discussed in this review.

Reference [21] recently proposed MeshNet which directly learns 3D shape representation from mesh data. The authors proposed face unit and feature splitting with a general architecture with effective blocks to deal with the key challenges of mesh data. The faces are treated as the unit and there exists a defined connection between faces sharing common edges which help to solve the irregularity and complexity problem with a symmetry function and per-face processes. Furthermore, faces features are split into structural feature and spatial descriptors that are used for learning the initial features and then a mesh convolution block for aggregating neighboring features. MeshNet was able to solve both the irregularity and complexity problem of mesh. MeshNet was tested on ModelNet 40 dataset for 3D shape retrieval and classification task. Experimental results and comparison with the state-of-the-art method shows MeshNet achieved satisfying result on ModelNet datasets. Reference [22] proposed MeshCNN that utilize the distinctive features of mesh data and design a convolutional neural network that specifically deals with triangular meshes. In this method, a well specialized convolution and pooling layers that work on the edges of the mesh are designed by taking advantage of their intrinsic connections. The convolution operations are applied on edges and the four edges of their incidental triangles and an edge collapse operation is used for pooling operation that maintains surface topology which generates new mesh connectivity for further convolutions. Using this approach, a task driven process was established were the network get rid of redundant features and then exposes and expands the significant ones. In MeshCNN, edges of a mesh are treated similar to pixels in an image because they are the building block which usually all operations are performed on. Since every edge is incident to exactly two faces (triangles) that normally defines a natural fixed sized convolution neighborhood of four edges, they exploit the consistent face normal order and apply a symmetric convolution operation which learns edges features that are invariant to transformations in translations, scale and rotation. Another important feature of MeshCNN is its pooling operations which adopted the popular mesh simplication technique [132] to downsample the features. But different from the original edge collapse which directly removes edges that have a minimal geometric distortion, the mesh pooling assigns the option of which edges to collapse to the network.

E. PERFORMANCE OF DEEP LEARNING METHODS ON HIGH-LEVEL STRUCTURES DATA

3D objects can also be represented in the form of high-level structure for 3D shape analysis tasks. 3D shape descriptors ease the operations of many shape analysis tasks e.g. for 3D shape retrieval which consist of querying a 3D object against a database of many 3D models in order to find the closest match, it is very important to transform the shapes into feature descriptors to keep the huge amount of 3D data models and efficiently query and find the nearest match. This section covers the up to date innovations in using deep learning methods on 3D descriptor and graphs.

1) PERFORMANCE OF DEEP LEARNING METHODS ON 3D DATA DESCRIPTOR

Zhang *et al.* [146] published the first survey on 3D shape descriptors in 2007 where a categorization of 3D shape descriptors was done. In 2013, Kazim *et al.* in [147] performed extensive reviewed on 2D and 3D descriptors. However, most of the previous reviews cover traditional approaches of constructing 3D shape descriptors. Recently, Rostami *et al.* [148] carryout a comprehensive study on Data-Driven 3D shape descriptors. In this study, the 3D descriptors are divided into two main categories which are shallow descriptors and deep shape descriptors. The shallow descriptors are further sub-divided into optimization based which are mostly implemented in a supervised manner [149] and clustering based descriptors that are mostly unsupervised and are built using Bag of Features technique (BoF) [150]. The deep shape descriptors are sub-divided into probabilistic models [151], auto-encoding [152], or CNN [101]. The probabilistic groups are again sub-divided into DBN-based and GAN-based. Deep learning models offered the advantage of learning hierarchical discriminative features effectively.

In [153], Liu *et al.* encoded low-level features in the visual Bag of Words (BoW) in order to learn high level features from DBNs for retrieval and classification tasks. An experiential evaluation shows that this approach achieved superior performance than the normal BoW low level features. Bu *et al.* [154] used a three-staged approach to learn the geometric essence of 3D objects. In this work, middle level geometric features are built from low-level features extracted from the 3D shapes and then a deep learning model was then utilized to learn the hierarchical high-level features of the 3D shapes. The Authors used scale invariant heat kernel signature [84] and Averaged Geodesic Distance (AGD) as local low-level descriptors and then employ the Spatially Sensitive Bag of Words (SS-BoW) to establish the connection between spatially close words from the extracted low-level features and finally using DBN to learn the high-level features from the SS-BoW). Experiments demonstrate the effective performance of this approach in comparisons to using low-level descriptors alone.

GPU implementation was used in the extension of [154] by Bu *et al.* in [155] which adopted a GPU based implementation for symmetry detection and correspondence tasks in which the proposed method showed improved performance. Inspired by the success of Heat kernel Signature (HKS) in obtaining low-level descriptors, Xie *et al.* [155] utilized the HKS as a low-level descriptor at different scales and used auto-encoder to discriminate features from the HKS for 3D shape retrieval task. In [156], Han *et al.* learn the discriminative features of 3D shapes from a Mesh Convolutional Restricted Boltzmann Machines(MCRBMs) in which Local Function Energy Distribution (LFED) was used to preserved the structure of the local features which leads to successful learning of the local and global features of 3D shapes. The use of MCRBMs which is a deeper model showed effective performance for shape retrieval and correspondence task which outperformed [4] and [84]. Ren *et al.* [157] proposed 3D A-Nets which is a 3D deep shape descriptor in which a new definition of 2D multi-layer dense representation (MDR) of 3D volumetric data was developed in order to capture geometric informative shape descriptor using adversarial neural networks that train a combination of convolutional neural network, adversarial discriminative and recurrent neural network. 3D shape features that stimulate clustering of samples of the same category with correct class label are produced by the generator network while the discriminator network prevent the clustering by allocating them with incorrect adversarial class labels which helps in addressing the challenges caused by computational cost of directly using CNN to 3D volumetric data. The proposed method was tested on ModelNet40 dataset which showed superior performance on 3D shape retrieval and classification tasks over the state of the art.

Motivated by the achievement of 3D deep shape descriptors, Xie *et al.* [158] proposed 3D DescriptorNet which combine the volumetric ConvNets in [136] and the generative ConvNet [159] to model 3D shape patterns. In this method, a probability density function is defined on voxilized shape signal and the model which is a deep convolutional energy-based model can synthesize 3D shape patterns by sampling from the probability distribution via MCMC like Langevin dynamics. In the training stage, analysis by synthesis [15] is used unlike the adversarial learning on the variational inference. The same set of parameters of a single model is used for both the learning and sampling process which leads to statistically rigorous framework. The model produced very high results when tested on ModelNet10 dataset.

[160], Bu *et al.* proposed 3D feature learning framework which fuse different representation data effectively using deep learning approach by promoting the discriminability of uni-modal feature. Firstly, CNNs and CDBNs are used to extract both the geometric and visual information of 3D data then two independent CDBNs are used to learn high-level features from geometric and visual features and finally a Restricted Boltzman Machine (RBM) was trained for mining the deep correlations between different representations. Experiments conducted for 3D shape retrieval and recognition tasks shows that the proposed framework achieved promising results. Similarly, Zhao *et al.* in [161] used a feature fusion method via multi-modal graph learning for view based 3D object retrieval. In this method, a variety of visual features including 2D fourier descriptor, 2D Zernike moments and 2D krawtchouk moments are extracted to describe each view of a 3D object. Then a similarity measure between two 3D objects with multiple views are computed using Haudorff distance and finally using different features, a multiple graph was constructed and the optimized weight of each graph was learned automatically for feature fusion task. Experiments performed on the ETH-80 datasets and National Taiwan university 3D model dataset demonstrate the

superior performance of the proposed method.The increasing availability of 3D models from construct and capture 3D data from low-cost acquisition devices and other modeling tools requires effective 3D shape descriptors in order to analyze and retrieve them, However, there is need for enough training data for the learning algorithm to extract 3D shape descriptors accurately from the examples which help reduce over fitting as in [162]. Despite some of the limitations mention above, the use of 3D descriptors have proved to be effective in 3D shape analysis task.

Recently, Vishwanath *et al.* [163], proposed two simple yet effective early fusion approaches to combine the RGB and point cloud representations which are term point-fusion and voxel-fusion. In this method, a multi-modal voxel-Net (MVX-Net) is presented which augment LIDAR points with semantic image features and learn to fuse image and LiDar features at early stages. In the point fusion, points from the LiDar sensor are projected onto the image plane, preceded by Image feature extraction from a pre-trained 2D detector and a voxelNet architecture is used to jointly processed the concatenation of image features and the corresponding points. While in the voxel fusion part, voxelNet is use to create a non-empty 3D voxel which are then followed by extracting image features for every projected voxel using a pre-trained CNN. The features are then pooled and appended to the VFE feature encoding for every voxel and further used by the 3D Recognition Proposal Network (RPN) to produce 3D bounding boxes. Experimental results on the KITTI datasets demonstrates significant improvements over approaches using a single representation.

2) PERFORMANCE OF DEEP LEARNING METHODS ON GRAPHS

The ability of graph convolution to encode the structure of graph on variety of data input using neural networks has attracted lots of attention recently [37], [164] and it can also be utilized in the semi supervised learning process. Previous approaches for Graph Convolution Neural Networks (GCNN) are usually divided into spatial filtering and Spectral filtering methods. The major difference between these two methods is on the way the locally processed information is merge and how the filtering is used. In this section, we will overview the performance of deep learning methods on both the spatial and spectral filtering methods.

Bruna *et al.* was the first to introduce the concept of spectral convolution on graph data structured in [165]. In this method, spectral CNN (SCNN) is proposed using spectral Eigen decomposition of the graph Laplacian to define a convolution operation. The authors successfully propose two constructions, the first based on the spectrum of the graph Laplacian and the second based on the hierarchical clustering of the domain. Experiments evaluation showed that it is feasible to learn convolutional layers on low-dimensional graphs with a number of parameters in respective of the size of the input which leads to efficient deep architectures. However, this method has some major limitations of being computationally expensive and inconsistent results are produced using the learned spectral filters coefficients on different domain with different basis as in [166]. Kovnatsky *et al.* in [167] overcome this limitation using orthogonal compatible basis on many domains by utilizing a joint diagonalization. But, this needs prior information of correspondence across domains. Some recent approaches used the concept of approximation to construct local spectral filters so that all graph information will be included in the processing [168], [169]. In these methods, the filters are represented through a polynomial expansion to avoid operating on the spectral domain directly. In [168], Defferred et al used Chebysher polynomials on graphs to performed local spectral filtering to estimate graph spectral filters. A more simplified polynomial approximation process introduced in [168] was proposed in [169] by Kipf and Welling. In this method local spectral filters are obtained by graph spectral filters first order linear approximation which is utilized in a two-layer graphical convolution neural network (GCNN). The local spectral filters are used in both the two layers and the information from the closest neighborhood of each vertex is then clustered together.

Inspired by the performance of the local spectral filtering models, Wang *et al.* [170] used the concept of PointNet++ [18] in GCNNs to deals with ordered point clouds. In this method, the framework of PointNet++ was combined with local spectral filtering in order to address two major limitations of these models separately. Spectral filtering was adopted as a learning approach to include the structural details in the neighborhood of all points against processing each point separately in the point clouds [13]. Furthermore, clustering strategy and recursive pooling are used against the graph max pooling operation which avoid the winner takes all approach. This method can be trained in an end to end form which has the advantages of dynamically building the graph and ease of computation of the pooling hierarchy and the graph Laplacian different from [165], [168] and [169]. This method was tested on numerous datasets and achieved a state-of-the-art performance.

The graph spatial filtering idea was first used in [171]. In this method, GNNs are introduced in an effort to used deep learning models on graphs. The GNNs consist of a number of layers in which the graph low-pass and high pass operators serve as the linear combination of each layer. This shows that the graph features learning is based on every vertex neighborhood. Highly sophisticated Architectures are obtained by fluctuating nonlinear function of the vertex [172]–[174]. Similar to CNNs, graph structured data can undergo pooling operation using graph coarsening. In spatial filtering, due to the spatial structure of the input graph, the graph topology is used to aggregate the neighborhood nodes with the feature vectors directly. The approaches used in [171] and [175] typically embed all vertex of the graph using the recursive connections in the RNN. A diffusion function was adopted for the transition function and the node representation are continuously propagated up to the time it is fixed and stable. The nodes representation obtained are then used as features for regression and classification problems. However, the major limitation here is the continues propagation of the node features which leads to a computational cost but addressed by [172]. Li *et al.* proposed a different version of the previous model that utilized the gated recurrent units to execute the update states in order to learn the optimal graph representation. The work of Bruna *et al.* [165] obtained the local spatial formulation of GNN by forcing the spatial local receptive field on the GNN to minimize the number of learned parameters by using similarity measure [176], [177] to group similar features. In Bruna *et al.* [165], the same idea of local receptive field was also used in the graph to calculate a multi-scale clustering to be supplied to the pooling layer which the model successfully decreases the number of processed parameters by forcing the locality on the already processed features.

Linked Dynamic Graph CNN (LDGCNN) for classifying and segmenting point clouds directly was proposed by Zhang *et al.* [178]. The LDGCNN consist of two parts; convolutional layers that serve as the extractors and the fully connected layers which are the classifiers. LDGCNN achieved a state-of-the-art performance on ModelNet40 and ShapeNet datasets. A comprehensive study about the spectral methods can be found in [179]. In this study, the authors showed that mathematically, both the spatial and spectral methods are equivalent more especially with the capabilities of their representations. However, the key distinctions are on the aggregation of the learned features and how the convolution operations are performed which depends on the task.

Recently, in [180], Feng *et al.* proposed a Hypergraph Neural Network (HGNN) for data representation learning. In this method, a hypergraph structure was used to encode high order data correlation in a hypergraph structure which is different from the normal graph that uses mandatory 2 for the degree for all edges. The framework uses degree free hypergraph which gives it's the ability to encode high order data correlation using a simple hyper-edge convolution operation. HGNN was tested on four standard 3D dataset which shows effective performance on learning data representation using high-order and more complex correlations.

3D meshes can also benefit from models design for graphs to be used on mesh structured data. In this setting, the nodes of the graph correlate to the vertices of the mesh while the edges are the connectivity between these vertices. Using this concept, Masci *et al.* [181] proposed Geodesic CNN which generalizes classical CNNs to triangular meshes. In this method, local patches are constructed in a local polar coordinate and a mapping is established between local polar coordinates and the values of the functions around each vertex through the patch operator that helps to define the patches in which the geodesic convolution can be used. Using the idea of multiplication by a template in which the geodesic convolution is built on while here arbitrary rotation is required for the convolution filters due to angular coordinate ambiguity. Despite the advantages of this method which results in using CNN framework to triangular meshes, it has a number of limitations that include computational cost due to rotations on the convolution filters. Boscaini *et al.* [182] try to overcome the limitations of [181] by proposing Anisotropic CNN (ACNN) that is not limited to triangle meshes alone and can easily be used on graphs. ACNN framework adopted a simpler construction of local patches which does not depend on the injectivity of the radius of the mesh. The idea of spectral filtering was used in which a weighting function was employ to in cooperate the spatial information to get a local function which is defined on the meshes. The Eigen values of the Anisotropic Laplacian Beltrami Operator (LBO) received the learnt spectral filters while anisotropic heat kernels serve as spatial weighting functions for the convolution filters. This approach shows remarkable performance for local correspondence tasks.

Monti *et al.* [38] proposed MoNet which void the used of fixed kernel construction. In this method, local systems of coordinates of pseudo-coordinates are defined around each vertex with weight functions. Fey *et al.* [183] proposed Spline CNN which discards the requiring of defining local patches on graph or meshes explicitly. Spline CNN can be utilized on meshes and graphs irrespective of dimensionality. The 1-hop neighborhood ring features of the graph as the patch is used in place of using the charting-based method to define the local patches. The method proved to be computationally efficient and state of the art results are produced on correspondence tasks which is attributed to the used of local support of the B-spline basis which helps to make the kernel size and the computational time independent.

F. PERFORMANCE OF DEEP LEARNING METHODS ON MIXED DATA REPRESENTATIONS

In order to take the full advantages of more than one 3D data representations, some efforts to combined more than one 3D data representation were exploited by Wang *et al.* [184] which uses a pair of multi views and 2D sketch to represent each 3D object. In this method, a Siamese CNNs is used which consist of two similar sub-convolutional networks in which one treats the 2D sketches and the other the multi views input. Both the networks are composed of three convolution layers followed by max pooling layer and a fully connected layer at the top. The stochastic Gradient Descent method was used to train the networks. The method was tested on three benchmark datasets including SHREC13, which achieved competitive results when compare to other methods in the literature on 3D shape retrieval task. Motivated by the performance of [184], the authors in [185] combine the learning pair of AEs, ConvNets and Extreme Learning Machine (ELM) to proposed Convolutional Auto -Encoder Extreme Learning Machine (CAE-ELM) 3D descriptor. they use the extreme learning machine as an Auto-Encoder to represent the input into three main representations which are: compressed representations, sparse representation and equal dimension representation. The ELM-AE hidden nodes biases and the random weights extend the input data to a different

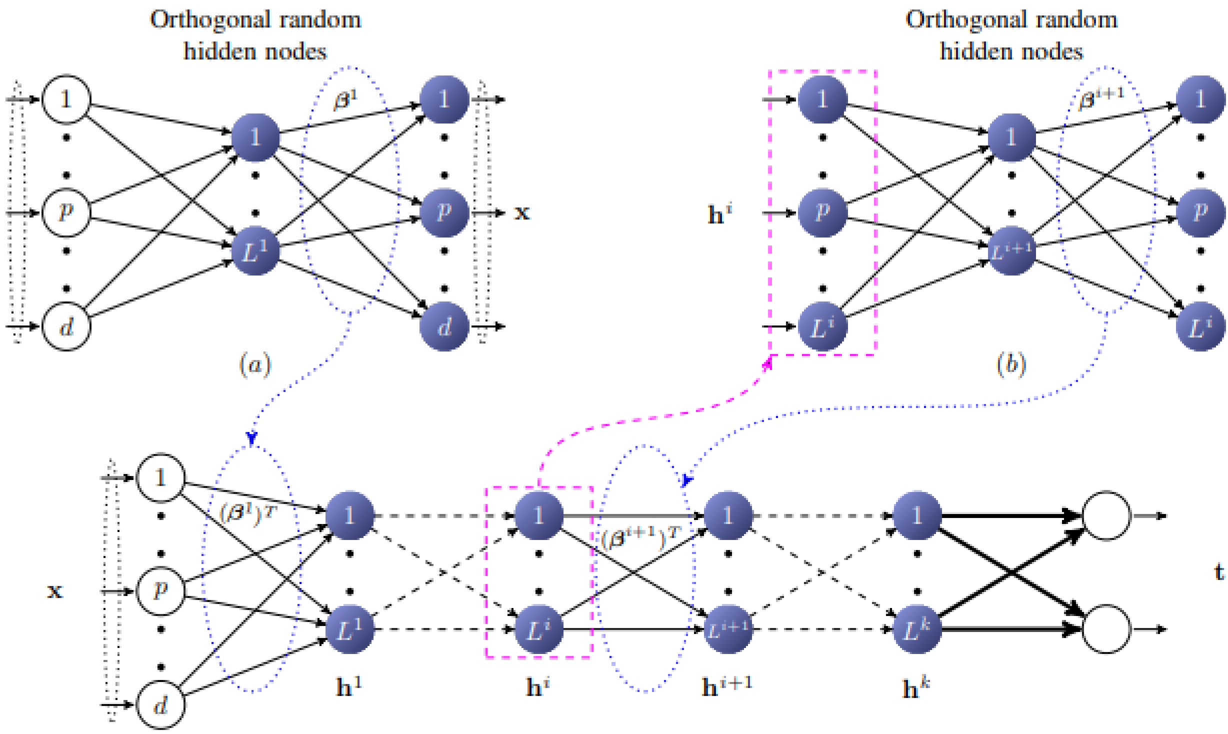

FIGURE 7. ML-ELM. (a) Output weights β^1 with respect to input data x (b) Output weights $\beta i + 1$ with respect to *ith* hidden layer (c) Final weights computed [186].

dimension space using the equation as in [186]:

$$h = g(ax + b) \tag{18}$$

$$a^T a = I, \quad b^T b = 1 \tag{19}$$

where $a = [a_1, \ldots, a_L]$ are the orthogonal random weight while $b = [b_1, \ldots, b_l]$ represent the random bias between the input nodes and hidden nodes [186]. In the case of compressed and and sparse ELM-AE representations, their output weights β are computed as in [186]:

$$\beta = (\frac{I}{C} + H^T H)^{-1} H^T X \tag{20}$$

where $H = [h_1, \ldots, h_N]$ serve as the outputs of the hidden layer of ELM-AE while $X = [x_1, \ldots, x_N]$ represent both the input data and output data of ELM-AE [186]. And the equal dimension ELM-AE representations, their weights output β are computed as in [186]:

$$\beta = H^{-1} X \tag{21}$$

$$\beta = {}^T \beta = I \tag{22}$$

The major advantages of the ELM are its ability to learns high level discriminative features of input data in unsupervised learning way which prove to be effective than many deep learning models [186] and efficient with large scale 3D datasets. The CAE-ELM accepts two different data representations which consist of Signed Distance Field (SDF) and voxels data while global and local features of the 3D models are obtained by the SDF. The CAE-ELM approach is a mixed method that utilized the 3D descriptor with the structured of the 3D objects and achieved a superior performance on classification task on ModelNet datasets. Figure 7. below show the Adding layers in ML-ELM.

The mixed methods continue to attract more interest and in [187], Ben-Shabat *et al.* proposed 3D modified Fisher vectors (3DMFV). The 3DMFV is a 3D point cloud representation that utilized the mixed data representations of continuous generalization of fisher vectors with discrete grid structure to represent the 3D data. In this method, the input point cloud is converted to 3D modified fisher vector by a module then another module which is the deep learning module is represented in the CNN. The network consists of fully connected layers on top, max-pooling layers, and an inception module [188]. The method achieved competitive performance when to the state of the art. Reference [189] also combine voxels and 2D views for object classification task by fushioning both representations. In this method, AlexNet was used for processing the 2D views while the 3D voxels are treated using two 3D CNNs. The advantages of this approach are that it does not require much computational requirements and there is no need for data augmentation.

V. CONCLUSION, DISCUSSION AND TRENDS FOR FUTURE RESEARCH

In this paper, we surveyed the performance of the deep learning methods based on the taxonomy proposed in figure 3 which is discussed here.

In figure 3, the 3D data representations are categorized based on the representations of the 3D object. We divided the 3D object representations into four major categories (Raw data, Surfaces, Solids, High level structures and Multi-views) accordingly.

Understanding the proper categorization of each deep learning method reviewed in the literature is a bit difficult for some methods due to the fact some approaches adopted the mixed data representations [184]. In assigning 3D data representations to most categories, we consider the mode of acquisition of the 3D data and the data structure of each representation.

Also, in the case of graphs, 3D meshes can also benefits from models design for graphs to be used on mesh structured data. But models design for meshes alone cannot be used for graphs. For example, Boscaini *et al.* [182] proposed Anisotropic CNN (ACNN) that can be used in both graph and meshes while in [21], Feng *et al.* proposed MeshNet which learns 3D shape representation from mesh data that can only be used on mesh data alone. We listed the method of [182] under graphs-based categorization not meshes because of the used of local patches construction which does not depends on the injectivity radius of the mesh. Some of the examples stated above shows that there is not rigid border among different 3D data representations proposed taxonomy.

In table 1, we summarize the reviewed 3D data representations based on divergent standard. The following list below analyzes each column of the table.

1) MODE OF ACQUISITION

This column describes how the 3D data is obtained. for example, RGB-D data are obtain from mostly kinect style sensors which are categorized as raw data

2) PROPERTY/KEY FEATURES

This column describes the key characteristics of each 3D data representations discussed in the literature

3) ADVANTAGES

This column describes the key benefits of each 3D data representations discussed in the literature

4) LIMITATIONS

This column describes the restriction or constraint of each 3D data representation.

Previously, deep learning methods have been used extensively in 1D and 2D data. However, utilizing them in 3D field is challenging due to the fact that most of the deep learning architectures previously designed used 1D or 2D as input data. To deal with this limitation, several deep learning models for 3D data have been proposed. Some researchers exploit the local or global descriptors for 3D data to extract low-level features but because the low-level descriptors are not strong enough to describe the high-level semantics of 3D shape other works used them together with deep learning models to obtained high-level descriptors. However, due to the complex nature of 3D data, this approach has insufficient discriminative power due to missing information from the 3D representations because of the shallow nature of the representations.

The availability of RGB-D datasets from RGB-D sensors e.g. Microsoft Kinect has motivated many researchers to exploit this data representation due to the presence of color and depth representation provided by the sensors. However, sometimes the data might be noisy and incomplete capture data which makes them difficult to use in complex situations. There is also the problem of not learning the full geometry of the 3D object which motivated some researches to exploit the full volumetric representations of the 3D shape. 3D volumetric are powerful and rich 3D shape representations that attracted many researches works but their major limitations is the huge demand for computation time and memory which makes them not suitable for high resolution data. Other works exploit the multi view 2D images which have the benefits of learning many feature sets to minimize noise, occlusion and incompleteness issues. However, selecting the number of views is still an open question with many views causing computational overhead.

In order to ease the comparisons between different methods, some of the works discussed in the literature addressing 3D shape classification and retrieval that used ModelNet40 datasets are presented in tables 2 respectively. The table give a summary of state-of-the-art works for 3D retrieval and classification tasks as well as the data representation used, the deep learning models adopted, the dataset and some other key experimental details.

For 3D shape classification, multi views performed very well as can be seen in table 2, Asako *et al.* [14] reported 97.3% classification accuracy on ModelNet40 datasets exceeding other 3D data representations by a relatively great margin. Sfikal *et al.* [114] using 3D data projections achieved mean Average Precision (mAP) of 93.2% exceeding by 2.1% the previous performance of 91.1% by Bai *et al.* [69] on the same dataset. Also [114] achieved 95.5% classification accuracy on ModelNet40 respectively. Recently, Feng *et al.* [180] using graph 3D data achieved 96.6% classification accuracy on ModelNet40 dataset surpassing [178] by 3.7% which also adopt the same graph 3D data respectively. Point clouds 3D data representations also demonstrate high performance with most methods reporting classification accuracy above 90% on both ModelNet40 datasets. Towards using the mixed 3D data representations, [184] reported 91.4% classification accuracy on ModelNet40 outperforming other methods that used single 3D data representation (Chen *et al.* [4], Shi *et al.* [8]) with more than 15% difference on the same ModelNet datasets.

In general, regarding the findings of this survey, the 3D data representation adopted plays a crucial role in determining the performance of a particular method and it can be concluded that deep learning together with a suitable 3D data representation presents effective approach for improving the performance of 3D shape analysis tasks.

All the 3D data representations discussed in this review are very active areas of research. Even though we have highlight where each 3D data has advantages over the other in conclusion no direct winner among the 3D data representations.

In spite of the fact that 3D deep learning is not as matured as 2D deep learning, the works reviewed showed a rapid developing community of researchers that are highly effective in solving 3D computer vision tasks. The papers reviewed present state of the art results using almost all the 3D data representations which clearly showed no winner takes all approach.

We present a summary table which addressed the key benefits and limitations of each 3D data. Furthermore, we looked at the major 3D benchmark datasets by discussing the origin and content of each dataset.

Recently, Wang *et al.* [24] performed 3D shape analysis tasks using octrees where they perform CNN computations in the octree data structure. Utilizing similar lattices hierarchical structures like the tetrahedral lattices or permutohedral lattices could be a great future research.

There have been many techniques in the literature that used 3D projections for 3D shape analysis tasks. Majority of these methods used traditional images, geometry images together with deep learning could be of great interest.

Another future direction is to continue to explore the mixed 3D data representations for feature extraction like the low-level and mid-level features and then used AEs for 3D shape retrieval.

REFERENCES

[1] A. Mademlis, P. Daras, D. Tzovaras, and M. G. Strintzis, "3D object retrieval using the 3D shape impact descriptor," *Pattern Recognit.*, vol. 42, no. 11, pp. 2447–2459, Nov. 2009.

[2] M. Savva *et al.*, "Large-scale 3D shape retrieval from ShapeNet Core55: SHREC'17 track," in *Proc. Workshop 3D Object Retr.*, Lisbon, Portugal, 2016, pp. 39–50.

[3] M. Savva *et al.*, "SHREC 17 track large-scale 3D shape retrieval from ShapeNet Core55," in *Proc. Eurograph. Workshop 3D Object Retr.*, Lyon, France, 2017, pp. 39–50.

[4] F. Chen, R. Ji, and L. Cao, "Multimodal learning for view-based 3D object classification," *Neurocomputing*, vol. 195, pp. 23–29, Jun. 2016.

[5] C. R. Qi, H. Su, M. Nießner, 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. (CVPR)*, Dec. 2016, pp. 5648–5656.

[6] 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.

[7] K. Sfikas, T. Theoharis, and I. Pratikakis, "Exploiting the PANORAMA representation for convolutional neural network classification and retrieval," in *Proc. Eurograph. Workshop 3D Object Retr. EG (3DOR)*, Apr. 2017, pp. 1–7.

[8] B. Shi, S. Bai, Z. Zhou, and X. Bai, "DeepPano: Deep panoramic representation for 3-D shape recognition," *IEEE Signal Process. Lett.*, vol. 22, no. 12, pp. 2339–2343, Dec. 2015.

[9] C. Wang, M. Pelillo, and K. Siddiqi, "Dominant set clustering and pooling for multi-view 3D object recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–12.

[10] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao, "GVCNN: Group-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 264–272.

[11] I. Atmosukarto and L. G. Shapiro, "3D object retrieval using salient views," *Int. J. Multimedia Inf. Retr.*, vol. 2, no. 2, pp. 103–115, Jun. 2013.

[12] P.-S. Wang, C.-Y. Sun, Y. Liu, and X. Tong, "Adaptive O-CNN: A patch-based deep representation of 3D shapes," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–11, Dec. 2018.

[13] A. Godil, "Applications of 3D shape analysis and retrieval," in *Proc. IEEE Appl. Imag. Pattern Recognit. Workshop (AIPR)*, Oct. 2009, pp. 1–4.

[14] A. Kanezaki, Y. Matsushita, and Y. Nishida, "RotationNet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5010–5019.

[15] Y. Zhao, Y. Liu, Y. Wang, B. Wei, J. Yang, Y. Zhao, and Y. Wang, "Region-based saliency estimation for 3D shape analysis and understanding," *Neurocomputing*, vol. 197, pp. 1–13, Jul. 2016.

[16] *3D Shape Retrieval Contest*. Accessed: Aug. 3, 2019. [Online]. Available: http://www.shrec.net/

[17] 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. (CVPR)*, Jan. 2017, pp. 77–85.

[18] C. Qi, L. Yi, H. Su, and L. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 5100–5109.

[19] 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, Sep. 2007.

[20] Q. Huang, Y. Wang, and Z. Yin, "View-based weight network for 3D object recognition," *Image Vis. Comput.*, vol. 93, Jan. 2020, Art. no. 103828.

[21] 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, Jul. 2019, pp. 8279–8286.

[22] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or, "MeshCNN," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–12, Jul. 2019.

[23] 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. (CVPR)*, Jun. 2015, pp. 1912–1920.

[24] 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, pp. 1–11, Jul. 2017.

[25] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3D object classification," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Dec. 2011, pp. 2987–2992.

[26] A. Aldoma, F. Tombari, R. B. Rusu, and M. Vincze, "OUR-CVFH—Oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6DOF pose estimation," in *Pattern Recognition DAGM/OAGM* (Lecture Notes in Computer Science), vol. 7476, A. Pinz, T. Pock, H. Bischof, and F. Leberl, Eds. Berlin, Germany: Springer, 2012, pp. 113–122. doi: 10.1007/978-3-642-32717-9_12.

[27] Z.-C. Marton, D. Pangercic, N. Blodow, and M. Beetz, "Combined 2D–3D categorization and classification for multimodal perception systems," *Int. J. Robot. Res.*, vol. 30, no. 11, pp. 1378–1402, Sep. 2011.

[28] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3212–3217.

[29] L. Yi, L. Guibas, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, and A. Sheffer, "A scalable active framework for region annotation in 3D shape collections," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–12, Nov. 2016.

[30] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.

[31] M. Elad, A. Tal, and S. Ar, "Content based retrieval of VRML objects—An iterative and interactive approach," in *Proc. 6th Eurograph. Workshop Multimedia*, 2001, pp. 97–108.

[32] 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.

[33] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Comput. Graph. Forum*, vol. 28, no. 5, pp. 1383–1392, Jul. 2009.

[34] R. M. Rustamov, "Laplace–Beltrami eigenfunctions for deformation invariant shape representation," in *Proc. 5th Eurograph. Symp. Geometry Process.*, 2007, pp. 225–233.

[35] M. Ovsjanikov, A. M. Bronstein, M. M. Bronstein, and L. J. Guibas, "Shape Google: A computer vision approach to isometry invariant shape retrieval," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops, ICCV Workshops*, Sep. 2009, pp. 320–327.

[36] A. M. Bronstein, M. M. Bronstein, R. Kimmel, M. Mahmoudi, and G. Sapiro, "A Gromov–Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching," *Int. J. Comput. Vis.*, vol. 89, nos. 2–3, pp. 266–286, Sep. 2010.

[37] A. Agathos, I. Pratikakis, P. Papadakis, S. Perantonis, P. Azariadis, and N. S. Sapidis, "3D articulated object retrieval using a graph-based representation," *Vis. Comput.*, vol. 26, no. 10, pp. 1301–1319, Oct. 2010.

[38] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jan. 2017, pp. 5425–5434.

[39] D. Griffiths and J. Boehm, "A review on deep learning techniques for 3D sensed data classification," *Remote Sens.*, vol. 11, no. 12, p. 1499, 2019.

[40] P. Wang, W. Li, P. Ogunbona, J. Wan, and S. Escalera, "RGB-D-based human motion recognition with deep learning: A survey," *Comput. Vis. Image Understand.*, vol. 171, pp. 118–139, Jun. 2018.

[41] A. Ioannidou, E. Chatzilari, S. Nikolopoulos, and I. Kompatsiaris, "Deep learning advances in computer vision with 3D data: A survey," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 1–38, Apr. 2017.

[42] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.

[43] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with Microsoft Kinect sensor: A review," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1318–1334, Oct. 2013.

[44] B.-Q. Shi, J. Liang, and Q. Liu, "Adaptive simplification of point cloud using-means clustering," *Comput.-Aided Des.*, vol. 43, no. 8, pp. 910–922, Aug. 2011.

[45] J. C. Rangel, V. Morell, M. Cazorla, S. Orts-Escolano, and J. García-Rodríguez, "Object recognition in noisy RGB-D data using GNG," *Pattern Anal. Appl.*, vol. 20, no. 4, pp. 1061–1076, Nov. 2017.

[46] J. Park, H. Kim, Y.-W. Tai, M. S. Brown, and I. Kweon, "High quality depth map upsampling for 3D-TOF cameras," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1623–1630.

[47] B. Huhle, T. Schairer, P. Jenke, and W. Strasser, "Robust non-local denoising of colored depth data," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2008, pp. 1–7.

[48] P. Jenke, M. Wand, M. Bokeloh, A. Schilling, and W. Straßer, "Bayesian point cloud reconstruction," *Comput. Graph. Forum*, vol. 25, no. 3, pp. 379–388, Sep. 2006.

[49] X.-F. Han, J. S. Jin, M.-J. Wang, and W. Jiang, "Guided 3D point cloud filtering," *Multimedia Tools Appl.*, vol. 77, no. 13, pp. 17397–17411, Jul. 2018.

[50] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1817–1824.

[51] J. Xiao, A. Owens, and A. Torralba, "SUN3D: A database of big spaces reconstructed using SfM and object labels," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1625–1632.

[52] G. Fanelli, T. Weise, J. Gall, and L. Van Gool, "Real time head pose estimation from consumer depth cameras," in *Proc. 33rd Annu. Symp. German Assoc. Pattern Recognit. (DAGM)*, 2011.

[53] M. Zollhöfer, C. Theobalt, M. Stamminger, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, and C. Loop, "Real-time non-rigid reconstruction using an RGB-D camera," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 1–12, Jul. 2014.

[54] N. Erdogmus and S. Marcel, "Spoofing in 2D face recognition with 3D masks and anti-spoofing with Kinect," in *Proc. IEEE 6th Int. Conf. Biometrics, Theory, Appl. Syst. (BTAS)*, Sep. 2013, pp. 1–6.

[55] C.-H. Lee, J.-L. Shih, K.-M. Yu, H.-Y. Chang, and Y.-C. Chiou, "Projection of shape features for 3D model retrieval," in *Proc. Int. Conf. Multimedia Technol.*, Jul. 2011.

[56] J.-B. Li, L.-J. Yu, and S.-Z. Guo, "Extracting invariable features using geometry projection based histogram model for 3D object description," in *Proc. 5th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, Sep. 2009, pp. 12–14.

[57] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2107–2115.

[58] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jan. 2017, pp. 6620–6629.

[59] A. A. Soltani, H. Huang, J. Wu, T. D. Kulkarni, and J. B. Tenenbaum, "Synthesizing 3D shapes via modeling multi-view depth maps and silhouettes with deep generative networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2511–2519.

[60] A. Brock, T. Lim, J. Ritchie, and N. Weston, "Generative and discriminative voxel modeling with convolutional neural networks," in *Proc. 3D Deep Learn. Workshop (NIPS)*, 2016, pp. 1–9.

[61] C. B. Choy, D. Xu, J. Y. Gwak, K. Chen, and S. Savarese, "3D-R2N2: A unified approach for single and multi-view 3D object reconstruction," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 628–644.

[62] S. Zhi, Y. Liu, X. Li, and Y. Guo, "Toward real-time 3D object recognition: A lightweight volumetric CNN framework using multitask learning," *Comput. Graph.*, vol. 71, pp. 199–207, Apr. 2018.

[63] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Data-driven 3D voxel patterns for object category recognition," in *Proc. Comput. Graph.*, Pergamon, Turkey, vol. 71, 2018, pp. 199–207.

[64] D. V. Vranic, D. Saupe, and J. Richter, "Tools for 3D-object retrieval: Karhunen–Loeve transform and spherical harmonics," in *Proc. IEEE 4th Workshop Multimedia Signal Process.*, Oct. 2001, pp. 293–298.

[65] D. McWherter, M. Peabody, W. C. Regli, and A. Shokoufandeh, "An approach to indexing databases of graphs," Drexel Univ., Philadelphia, PA, USA, Tech. Rep., Sep. 2001.

[66] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii, "Topology matching for fully automatic similarity estimation of 3D shapes," in *Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, 2001, pp. 203–212.

[67] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson, "Skeleton based shape matching and retrieval," in *Proc. Shape Modeling Int.*, 2003, pp. 130–139.

[68] M. Fey, J. E. Lenssen, F. Weichert, and H. Muller, "SplineCNN: Fast geometric deep learning with continuous B-spline kernels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 869–877.

[69] S. Bai, X. Bai, Z. Zhou, Z. Zhang, and L. J. Latecki, "GIFT: A real-time and scalable 3D shape search engine," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5023–5032.

[70] 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. (ICCV)*, Dec. 2015, pp. 945–953.

[71] *Network of Excellence AIM@SHAPE*. Accessed: Feb. 11, 2019. [Online]. Available: https://www.csd.uoc.gr/~pitikakis/project/aimatshape/

[72] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[73] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[74] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–3440.

[75] X. Shu, G.-J. Qi, J. Tang, and J. Wang, "Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation," in *Proc. 23rd ACM Int. Conf. Multimedia (MM)*, 2015, pp. 35–44.

[76] J. Tang, X. Shu, Z. Li, G.-J. Qi, and J. Wang, "Generalized deep transfer networks for knowledge propagation in heterogeneous domains," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 12, no. 4s, pp. 68:1–68:22, Nov. 2016.

[77] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world Web image database from National University of Singapore," in *Proc. ACM Int. Conf. Image Video Retr.*, New York, NY, USA, vol. 48, pp. 1–9, doi: 10.1145/1646396.1646452.

[78] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.

[79] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst, "Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks," *Comput. Graph. Forum*, vol. 34, no. 5, pp. 13–23, Aug. 2015.

[80] Y. Zhang and M. Rabbat, "A graph-CNN for 3D point cloud classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 6279–6283.

[81] G. Kyriakaki, A. Doulamis, N. Doulamis, M. Ioannides, K. Makantasis, E. Protopapadakis, A. Hadjiprocopis, K. Wenzel, D. Fritsch, M. Klein, and G. Weinlinger, "4D reconstruction of tangible cultural heritage objects from Web-retrieved images," *Int. J. Heritage Digit. Era*, vol. 3, no. 2, pp. 431–451, Jun. 2014.

[82] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943.

[83] M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature: A quantum mechanical approach to shape analysis," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Nov. 2011, pp. 1626–1633.

[84] M. M. Bronstein and I. Kokkinos, "Scale-invariant heat kernel signatures for non-rigid shape recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1704–1711.

[85] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 3384–3391.

[86] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 286–299, Feb. 2007.

[87] O. Vinyals, S. Bengio, and M. Kudlur, "Order matters: Sequence to sequence for sets," 2015, *arXiv:1511.06391*. [Online]. Available: http://arxiv.org/abs/1511.06391

[88] S. Ravanbakhsh, J. Schneider, and B. Poczos, "Deep learning with sets and point clouds," 2016, *arXiv:1611.04500*. [Online]. Available: http://arxiv.org/abs/1611.04500

[89] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 3392–3402.

[90] P. Daras and A. Axenopoulos, "A 3D shape retrieval framework supporting multimodal queries," *Int. J. Comput. Vis.*, vol. 89, nos. 2–3, pp. 229–247, Sep. 2010.

[91] 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. (ICCV)*, Oct. 2017, pp. 863–872.

[92] R. Roveri, L. Rahmann, A. C. Oztireli, and M. Gross, "A network architecture for point cloud classification via automatic depth images generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4176–4184.

[93] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2016, pp. 770–778.

[94] Y. Yang, C. Feng, Y. Shen, and D. Tian, "FoldingNet: Point cloud autoencoder via deep grid deformation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 206–215.

[95] J. Li, B. M. Chen, and G. H. Lee, "SO-net: Self-organizing network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 9397–9406.

[96] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8895–8904.

[97] F. Engelmann, T. Kontogianni, J. Schult, and B. Leibe, "Know what your neighbors do: 3D semantic segmentation of point clouds," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, 2019, pp. 8895–8904.

[98] M. Jiang, Y. Wu, T. Zhao, Z. Zhao, and C. Lu, "PointSIFT: A SIFT-like network module for 3D point cloud semantic segmentation," 2018, *arXiv:1807.00652*. [Online]. Available: http://arxiv.org/abs/1807.00652

[99] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz, "SPLATNet: Sparse lattice networks for point cloud processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2530–2539.

[100] P. Wang, W. Li, Z. Gao, C. Tang, J. Zhang, and P. Ogunbona, "ConvNets-based action recognition from depth maps through virtual cameras and pseudocoloring," in *Proc. 23rd ACM Int. Conf. Multimedia (MM)*, 2015, pp. 1119–1122.

[101] R. Socher, B. Huval, B. Bhat, C. Manning, and A. Ng, "Convolutional-recursive deep learning for 3D object classification," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 1, 2012, pp. 656–664.

[102] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, "Indoor semantic segmentation using depth information," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2013, pp. 1–8.

[103] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust RGB-D object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 681–687.

[104] J. Feng, Y. Wang, and S.-F. Chang, "3D shape retrieval using a single depth image from low-cost sensors," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–9.

[105] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[106] Z. Li, Y. Gan, X. Liang, Y. Yu, H. Cheng, and L. Lin, "LSTM-CF: Unifying context modeling and fusion with LSTMs for RGB-D scene labeling," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 541–557.

[107] L. A. Alexandre, "3D object recognition using convolutional neural networks with transfer learning between input channels," in *Intelligent Autonomous Systems* (Advances in Intelligent Systems and Computing), vol. 302, H. Yamaguchi, Ed. Cham, Switzerland: Springer, 2016, pp. 889–898.

[108] M. Schwarz, H. Schulz, and S. Behnke, "RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1329–1335.

[109] J. Zhu, J.-R. Rizzo, and Y. Fang, "Learning domain-invariant feature for robust depth-image-based 3D shape retrieval," *Pattern Recognit. Lett.*, vol. 119, pp. 24–33, Mar. 2019.

[110] Z. Zhu, X. Wang, S. Bai, C. Yao, and X. Bai, "Deep learning representation using autoencoder for 3D shape retrieval," *Neurocomputing*, vol. 204, pp. 41–50, Sep. 2016.

[111] B. Leng, S. Guo, X. Zhang, and Z. Xiong, "3D object retrieval with stacked local convolutional autoencoder," *Signal Process.*, vol. 112, pp. 119–128, Jul. 2015.

[112] A. Sinha, J. Bai, and K. Ramani, "Deep learning 3D shape surfaces using geometry images," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 223–240.

[113] Z. Cao, Q. Huang, and R. Karthik, "3D object classification via spherical projections," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 566–574.

[114] K. Sfikas, T. Theoharis, and I. Pratikakis, "Exploiting the panorama representation for convolutional neural network classification and retrieval," in *Proc. Eurographics Workshop 3D Object Retr. EG (3DOR)*, Apr. 2017, pp. 1–7.

[115] 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.

[116] B. Leng, X. Zhang, M. Yao, and Z. Xiong, "3D object classification using deep belief networks," in *MultiMedia Modeling. MMM* (Lecture Notes in Computer Science), vol. 8326, C. Gurrin, F. Hopfgartner, W. Hurst, H. Johansen, H. Lee, and N. O'Connor, Eds. Cham, Switzerland: Springer, 2014, pp. 128–139.

[117] Z. Xie, K. Xu, W. Shan, L. Liu, Y. Xiong, and H. Huang, "Projective feature learning for 3D shapes with multi-view depth images," *Comput. Graph. Forum*, vol. 34, no. 7, pp. 1–11, Oct. 2015.

[118] B. Leng, Y. Liu, K. Yu, X. Zhang, and Z. Xiong, "3D object understanding with 3D convolutional neural networks," *Inf. Sci.*, vol. 366, pp. 188–201, Oct. 2016.

[119] E. Johns, S. Leutenegger, and A. J. Davison, "Pairwise decomposition of image sequences for active multi-view recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3813–3822.

[120] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–11.

[121] P. Zanuttigh and L. Minto, "Deep learning for 3D shape classification from multiple depth maps," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3615–3619.

[122] A.-A. Liu, W.-Z. Nie, and Y.-T. Su, "3D object retrieval based on multi-view latent variable model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 3, pp. 868–880, Mar. 2019.

[123] 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.

[124] E. Paquet, M. Rioux, A. Murching, T. Naveen, and A. Tabatabai, "Description of shape information for 2-D and 3-D objects," *Signal Process., Image Commun.*, vol. 16, nos. 1–2, p. 103–122, 2000.

[125] R. Ohbuchi, K. Osada, T. Furuya, and T. Banno, "Salient local visual features for shape-based 3D model retrieval," in *Proc. IEEE Int. Conf. Shape Modeling Appl.*, Jun. 2008, pp. 93–102.

[126] J. W. H. Tangelder and R. C. Veltkamp, "Polyhedral model retrieval using weighted point sets," *Int. J. Image Graph.*, vol. 3, no. 1, pp. 209–229, Jan. 2003.

[127] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, and S. Dickinson, "Retrieving articulated 3-D models using medial surfaces," *Mach. Vis. Appl.*, vol. 19, no. 4, pp. 261–275, Jul. 2008.

[128] Y. Gao, M. Wang, D. Tao, R. Ji, and Q. Dai, "3-D object retrieval and recognition with hypergraph analysis," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4290–4303, Sep. 2012.

[129] B. Bustos, D. A. Keim, D. Saupe, T. Schreck, and D. V. Vranić, "Feature-based similarity search in 3D object databases," *ACM Comput. Surv.*, vol. 37, no. 4, pp. 345–387, Dec. 2005.

[130] W. Nie, Q. Cao, A. Liu, and Y. Su, "Convolutional deep learning for 3D object retrieval," *ACM Comput. Surv.*, vol. 1, no. 1, p. 1–8, 2015.

[131] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit*, vol. 1, Jun. 2005, pp. 886–893.

[132] W.-Y. Kim and Y.-S. Kim, "A region-based shape descriptor using Zernike moments," *Signal Process., Image Commun.*, vol. 16, nos. 1–2, pp. 95–102, Sep. 2000.

[133] X. He, Y. Zhou, Z. Zhou, S. Bai, and X. Bai, "Triplet-center loss for multi-view 3D object retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1945–1954.

[134] J. Jiang, D. Bao, Z. Chen, X. Zhao, and Y. Gao, "MLVCNN: Multi-loop-view convolutional neural network for 3D shape retrieval," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 8513–8520.

[135] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 609–616.

[136] 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.

[137] N. Sedaghat, M. Zolfaghari, E. Amiri, and T. Brox, "Orientation-boosted voxel nets for 3D object recognition," 2016, *arXiv:1604.03351*. [Online]. Available: http://arxiv.org/abs/1604.03351

[138] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative adversarial modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 82–90.

[139] 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: http://arxiv.org/abs/1608.04236

[140] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. Conf. Assoc. Adv. Artif. Intell. (AAAI)*, vol. 4, 2017, p. 12.

[141] X. Xu and S. Todorovic, "Beam search for learning a deep convolutional neural network of 3D shapes," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 3505–3511.

[142] S. Song and J. Xiao, "Deep sliding shapes for amodal 3D object detection in RGB-D images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 808–816.

[143] C. Wang, M. Cheng, F. Sohel, M. Bennamoun, and J. Li, "NormalNet: A voxel-based CNN for 3D object classification and retrieval," *Neurocomputing*, vol. 323, pp. 139–147, Jan. 2019.

[144] C. Hane, S. Tulsiani, and J. Malik, "Hierarchical surface prediction for 3D object reconstruction," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 412–420.

[145] H. Hoppe, "View-dependent refinement of progressive meshes," in *Proc. 24th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, 1997, pp. 189–198.

[146] L. Zhang, M. João, and A. Ferreira, "Survey on 3D shape descriptors," in *Proc. DecorAR*, 2004, pp. 1–28.

[147] I. K. Kazmi, L. You, and J. J. Zhang, "A survey of 2D and 3D shape descriptors," in *Proc. 10th Int. Conf. Comput. Graph., Imag. Vis. (CGIV)*, Aug. 2013, pp. 1–10.

[148] R. Rostami, F. S. Bashiri, B. Rostami, and Z. Yu, "A survey on data-driven 3D shape descriptors," *Comput. Graph. Forum*, vol. 38, no. 1, pp. 356–393, Feb. 2019.

[149] F. Steinke, B. Schölkopf, and V. Blanz, "Learning dense 3D correspondence," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1313–1320.

[150] R. Toldo, U. Castellani, and A. Fusiello, "Visual vocabulary signature for 3D object retrieval and partial matching," in *Proc. 2nd Eurographics Conf. 3D Object Retr.* Aire-la-Ville, Switzerland: Eurographics Association, 2009, pp. 21–28.

[151] V. Nair and H. G. E. Inton, "3D object recognition with deep belief nets," in *Proc. Adv. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2009, pp. 1339–1347.

[152] G. Alain and Y. Bengio, "What regularized auto-encoders learn from the data-generating distribution," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3563–3593, 2015.

[153] Z. Liu, S. Chen, S. Bu, and K. Li, "High-level semantic feature for 3D shape based on deep belief networks," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2014, pp. 1–6.

[154] S. Bu, Z. Liu, J. Han, J. Wu, and R. Ji, "Learning high-level feature by deep belief networks for 3-D model retrieval and recognition," *IEEE Trans. Multimedia*, vol. 16, no. 8, pp. 2154–2167, Dec. 2014.

[155] S. Bu, P. Han, Z. Liu, J. Han, and H. Lin, "Local deep feature learning framework for 3D shape," *Comput. Graph.*, vol. 46, pp. 117–129, Feb. 2015.

[156] 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.

[157] M. Ren, L. Niu, and Y. Fang, "3D-A-nets: 3D deep dense descriptor for volumetric shapes with adversarial networks," 2017, *arXiv:1711.10108*. [Online]. Available: http://arxiv.org/abs/1711.10108

[158] J. Xie, Z. Zheng, R. Gao, W. Wang, S.-C. Zhu, and Y. N. Wu, "Learning descriptor networks for 3D shape synthesis and analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8629–8638.

[159] J. Xie, Y. Lu, S. Zhu, and Y. Wu, "A theory of generative ConvNet," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, vol. 6, 2016, pp. 3895–3904.

[160] S. Bu, L. Wang, P. Han, Z. Liu, and K. Li, "3D shape recognition and retrieval based on multi-modality deep learning," *Neurocomputing*, vol. 259, pp. 183–193, Oct. 2017.

[161] S. Zhao, H. Yao, Y. Zhang, Y. Wang, and S. Liu, "View-based 3D object retrieval via multi-modal graph learning," *Signal Process.*, vol. 112, pp. 110–118, Jul. 2015.

[162] U. Grenander and M. Miller, *Pattern Theory: From Representation to Inference*. London, U.K.: Oxford Univ. Press, 2007, pp. 468–493.

[163] V. A. Sindagi, Y. Zhou, and O. Tuzel, "MVX-net: Multimodal VoxelNet for 3D object detection," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 20–24.

[164] A. Narayan and P. H. O'N Roe, "Learning graph dynamics using deep neural networks," *IFAC-PapersOnLine*, vol. 51, no. 2, pp. 433–438, 2018.

[165] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–14.

[166] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.

[167] A. Kovnatsky, M. M. Bronstein, A. M. Bronstein, K. Glashoff, and R. Kimmel, "Coupled quasi-harmonic bases," *Comput. Graph. Forum*, vol. 32, no. 4, pp. 439–448, May 2013.

[168] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[169] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: http://arxiv.org/abs/1609.02907

[170] C. Wang, B. Samari, and K. Siddiqi, "Local spectral graph convolution for point set feature learning," 2018, *arXiv:1803.05827*. [Online]. Available: http://arxiv.org/abs/1803.05827

[171] F. Scarselli, M. Gori, A. Chung Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[172] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," 2015, *arXiv:1511.05493*. [Online]. Available: http://arxiv.org/abs/1511.05493

[173] S. Sukhbaatar, A. Szlam, and R. Fergus, "Learning multiagent communication with backpropagation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2252–2260.

[174] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2224–2232.

[175] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 2, Jul. 2005, pp. 729–734.

[176] A. Coates and A. Y. Ng, "Selecting receptive fields in deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2528–2536.

[177] K. Gregor and Y. LeCun, "Emergence of complex-like cells in a temporal product network with local receptive fields," 2010, *arXiv:1006.0448*. [Online]. Available: http://arxiv.org/abs/1006.0448

[178] K. Zhang, M. Hao, J. Wang, C. W. de Silva, and C. Fu, "Linked dynamic graph CNN: Learning on point cloud via linking hierarchical features," 2019, *arXiv:1904.10014*. [Online]. Available: http://arxiv.org/abs/1904.10014

[179] S. Bojja Venkatakrishnan, M. Alizadeh, and P. Viswanath, "Graph2Seq: Scalable learning dynamics for graphs," 2018, *arXiv:1802.04948*. [Online]. Available: http://arxiv.org/abs/1802.04948

[180] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 3558–3565.

[181] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on Riemannian manifolds," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, vol. 33, Dec. 2015, pp. 37–45.

[182] 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. 3197–3205.

[183] M. Fey, J. E. Lenssen, F. Weichert, and H. Muller, "SplineCNN: Fast geometric deep learning with continuous B-spline kernels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 869–877.

[184] F. Wang, L. Kang, and Y. Li, "Sketch-based 3D shape retrieval using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1875–1883.

[185] S. Ding, H. Zhao, Y. Zhang, X. Xu, and R. Nie, "Extreme learning machine: Theory and applications," *Artif. Intell. Rev.*, vol. 44, no. 1, pp. 103–115, 2015.

[186] L. Kasun, H. Zhou, G. Huang, and C. Vong, "Representational learning with ELMs for big data," *IEEE Intell. Syst.*, vol. 28, no. 6, pp. 31–34, Nov. 2013.

[187] Y. Ben-Shabat, M. Lindenbaum, and A. Fischer, "3D point cloud classification and segmentation using 3D modified Fisher vector representation for convolutional neural networks," 2017, *arXiv:1711.08241*. [Online]. Available: https://arxiv.org/abs/1711.08241

[188] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inceptionv4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.

[189] V. Hedge and R. Zadeh, "Fusionset: 3D object classification using multiple data representations," 2016, *arXiv:1607.05695*. [Online]. Available: https://arxiv.org/abs/1607.05695

[190] M. Blum, J. T. Springenberg, J. Wulfing, and M. Riedmiller, "A learned feature descriptor for object recognition in RGB-D data," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 1298–1303.

[191] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Computer Vision—ECCV* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7576, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Germany: Springer, 2012, pp. 746–760.

[192] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 567–576.

[193] M. Savva, A. X. Chang, and P. Hanrahan, "Semantically-enriched 3D models for common-sense knowledge," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 24–31.

[194] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton shape benchmark," in *Proc. Shape Model. Appl. (SMI)*, 2004, pp. 167–178.

[195] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "Scape: Shape completion and animation of people," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 408–416, 2005.

[196] M. B. Alexander, M. B. Michael, and K. Ron, "Numerical geometry of non-rigid shapes," *J. Chem. Inf. Model.*, vol. 53, p. 160, Sep. 2008.

[197] (2019). *3D Shape Retrieval Contest*. Accessed: Aug. 3, 2019. [Online]. Available: https://www.shrec.net/

[198] R. Fang, A. Godil, X. Li, and A. Wagan, "A new shape benchmark for 3D object retrieval," in *Advances in Visual Computing. ISVC* (Lecture Notes in Computer Science), vol. 5358, G. Bebis *et al.*, Eds. Berlin, Germany: Springer, 2008, pp. 381–392.

[199] R. Muthuganapathy and K. Ramani, "SHape REtrieval contest 2008: CAD models," in *Proc. IEEE Int. Conf. Shape Modeling Appl. (SMI)*, Jun. 2008, pp. 221–222.

[200] (2019). *Princeton ModelNet*. Accessed: Sep. 9, 2019. [Online]. Available: http://modelnet.cs.princeton.edu/

[201] T. K. Dey, K. Li, C. Luo, P. Ranjan, I. Safa, and Y. Wang, "Persistent heat signature for pose-oblivious matching of incomplete models," *Comput. Graph. Forum*, vol. 29, no. 5, pp. 1545–1554, 2010.

[202] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. NieBner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2432–2443.

[203] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani, "Three-dimensional shape searching: State-of-the-art review and future trends," *Comput.-Aided Des.*, vol. 37, no. 5, pp. 509–530, Apr. 2005.

ABUBAKAR SULAIMAN GEZAWA received the B.Sc. degree in computer science from Bayero University Kano and the M.Eng. degree in computer science and technology from the Liaoning University of Technology, China. He is currently pursuing the Ph.D. degree in computer science and technology with Xiamen University, China. His current research interests include computer vision, 3D deformable object recognition, and computer graphics.

YAN ZHANG received the bachelor's degree in computer science and technology from Guizhou Normal University, China, and the M.Eng. degree in software engineering from Guizhou University, China. She is currently pursuing the Ph.D. degree in computer technology with Xiamen University, China. Her current research interests include computer vision and image processing, neuron networks, image retrieval, object detection, and machine learning.

QICONG WANG received the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2007. He is currently an Associate Professor with the Department of Computer Science and Shenzhen Research Institute, Xiamen University, China. His research interests include machine vision, robot navigation, and machine learning.

LEI YUNQI received the B.Eng. degree in electronics from the University of Science and Technology of China, in 1982, the M.Eng. degree in marine electric engineering from the University of the Navy Engineering, China, in 1984, and the Ph.D. degree in automation from the National University of Defense Technology, China, in 1988. He is currently a Professor with the Computer Science Department, Xiamen University. His research interests include computer vision and image processing, machine learning, big data and cloud computing, and computer networks. He is a Senior Member of the China Electronics Society.