



Review

Deep Learning on Point Clouds and Its Application: A Survey

Weiping Liu ¹, Jia Sun ², Wanyi Li ^{2,*} , Ting Hu ^{1,*}  and Peng Wang ²

¹ School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China; weipingliu_17@whu.edu.cn

² Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China; jia.sun@ia.ac.cn (J.S.); peng_wang@ia.ac.cn (P.W.)

* Correspondence: wanyi.li@ia.ac.cn (W.L.); tinghu@whu.edu.cn (T.H.)

Received: 16 August 2019; Accepted: 13 September 2019; Published: 26 September 2019



Abstract: Point cloud is a widely used 3D data form, which can be produced by depth sensors, such as Light Detection and Ranging (LIDAR) and RGB-D cameras. Being unordered and irregular, many researchers focused on the feature engineering of the point cloud. Being able to learn complex hierarchical structures, deep learning has achieved great success with images from cameras. Recently, many researchers have adapted it into the applications of the point cloud. In this paper, the recent existing point cloud feature learning methods are classified as point-based and tree-based. The former directly takes the raw point cloud as the input for deep learning. The latter first employs a k-dimensional tree (Kd-tree) structure to represent the point cloud with a regular representation and then feeds these representations into deep learning models. Their advantages and disadvantages are analyzed. The applications related to point cloud feature learning, including 3D object classification, semantic segmentation, and 3D object detection, are introduced, and the datasets and evaluation metrics are also collected. Finally, the future research trend is predicted.

Keywords: feature learning; deep learning; point cloud; application of point cloud

1. Introduction

Providing detailed information for objects and environments, the point cloud is widely used in various applications such as digital preservation, reverse engineering, surveying, architecture, 3D gaming, robotics, and virtual reality. Some detailed examples are given here. In the digital preservation area, visually aesthetic and detailed 3D models of buildings and historical cities are generated by laser scanning and digital photogrammetry [1,2]. In the robotics area, point clouds are used to recognize the identity, pose, and location of the target object and obstacles for robot movement and manipulation [3,4].

Point clouds are generally produced by 3D scanners, Light Detection and Ranging (LIDAR), structure-from-motion (SFM) techniques, and recently available 3D sensors, such as Kinect and Xtion. SFM- and photogrammetry-generated point clouds usually have a low and sparse point density, while 3D scanners, LIDAR, and depth sensors can generate point clouds with more points. However, compared to the continuous surface of a 3D scene, sensed point clouds are still quite sparse. For this reason, as a pre-processing step, some techniques have been developed for densifying these point clouds, such as dense image matching. Another strategy is to use complementary data obtained from other techniques; an example is to complement data generated from structure-from-motion techniques with laser scanning. In some point clouds occlusions often occur, which request to use additional techniques for making up gaps. A common strategy in studies related to digital preservation is combining laser scanner with photogrammetry. Regarding the point density of generated point clouds, it is affected by the laser device mechanism and the object reflectivity. As an example, a typical LIDAR model, such as the HDL-64E [5], can generate a point cloud of up to ~2.2 million points

per second with a range of up to 120 m. Usually, a specific device offers a user-selectable parameter range, such as rotation rate for the LIDAR sensor, to determine the density of data points. Moreover, the range accuracy of produced points can be up to ± 2 cm. Point cloud consists of points with 3D unstructured vectors. Each point can be expressed by a vector, indicating its 3D coordinate and some extra feature channels, such as the intensity of reflection, color, and normals. There are three core properties for the point cloud [6], including being unordered, interaction among points, and invariance under transformations. Traditional approaches for dealing with point clouds are highly dependent on handcrafted features and well-designed optimization approaches. Features on point clouds describing their statistical properties can be divided into intrinsic or extrinsic which are invariant to several transformations [7,8]. Optimization methods are usually designed for a given application. Therefore, they have poor generalization [9,10].

Being automatically learning discriminative features, deep learning has achieved great success in object classification, semantic segmentation, object detection, etc. with optical images [11–13]. Recently, inspired by dense convolution, which can acquire translation invariance, feature learning approaches have been adapted to address point clouds in recent years [14–17]. These methods transform the sparse point clouds into dense tensors, including volumetric forms [18–21] and 2D images [20,22,23], or extract feature descriptors from the point clouds [24], and give these as input to deep neural networks (ConvNets). These methods usually missed much information, and the accuracy of proxy of original points became worse since they require quantization of point clouds with certain resolutions or extract descriptors from the 3D data before feeding information to ConvNets. [25] summarized the related literature and provided several directions. Different from [25], this paper focuses on methods which consume point clouds directly or convert them lossless before feature learning.

Since point clouds are important, and works of point cloud with deep learning have not been summarized yet, this paper provides an overview of the state-of-the-art progress on point clouds based on deep learning. The existing point cloud feature learning methods are classified and summarized, and their advantages and disadvantages are analyzed in this paper. Applications related to point cloud feature learning are introduced, and the related data sets and evaluation indexes are introduced. The contribution of this review has two aspects:

1. Recent advances on point clouds with deep learning are surveyed. The architectures can be classified into two categories, i.e., raw point-based and tree-based architectures. Additionally, their differences from unstructured and disordered point clouds are highlighted.
2. Applications of point clouds with deep learning are compared, and the future direction is given.

The organization of this review is as follows. The most related work of this survey is shown in Section 2. Feature learning with point clouds is introduced in Section 3, including raw point-based and tree-based types. Following this, the applications of point clouds, containing 3D object classification, semantic segmentation, and 3D object detection are described in Section 4. The performance discussion and future direction are given in Section 5. Finally, the conclusion is given in Section 6.

2. Related Works

Due to the availability of 3D point clouds from 3D scanners, they are widely used. Traditional methods depend on discriminative feature extractors [9,15,26]. Since deep learning has achieved great success in object classification [11,27–29], semantic segmentation [30–32], object detection [33–39], etc., it has been applied to address the corresponding tasks with point clouds [16,25]. The main contents of the related works are shown in Table 1.

Table 1. Related works on surveys of point clouds and their application.

Reference	Main Contents
Nygren et al. 2016 [26]	The traditional algorithms for 3D point cloud segmentation and classification
Nguyen et al. 2013 [15]	The segmentation methods for the 3D point cloud.
Ahmed et al. 2018 [16]	The 3D data from Euclidean and the non-Euclidean geometry and a discussion on how to apply deep learning to the 3D dataset.
Hana et al. 2018 [9]	The feature descriptors of point clouds with three classes, i.e., local-based, global-based, and hybrid-based.
Garcia et al. 2017 [40]	The semantic segmentation methods based on deep learning.
Bronstein et al. 2017 [41]	The problems of geometric deep learning, extending grid-like deep learning methods to non-Euclidean structures.
Griffiths et al. 2019 [42]	The classification models for processing 3D unstructured Euclidean data.

The methods in these surveys address point clouds without raw input, missing information, or inducing heavy computing. With the emergence of PointNet, there are deep learning models taking the raw point cloud as input. Since these methods have not been surveyed yet, we will survey the recent papers in this paper.

3. Feature Learning on Point Cloud

At present, feature learning has been widely used with point clouds. The methods can be classified into two categories, (1) raw point-based methods, which directly consume unstructured and unordered point clouds for deep learning models and (2) k-dimensional tree (Kd-tree) based methods, which represent the point cloud regularly before feeding information into the models. Currently, there are state-of-the-art deep learning models directly addressing point clouds [6,43–47], and the main 18 methods are shown in Figure 1. We will first introduce the raw point-based deep learning and then the tree-based deep learning method.

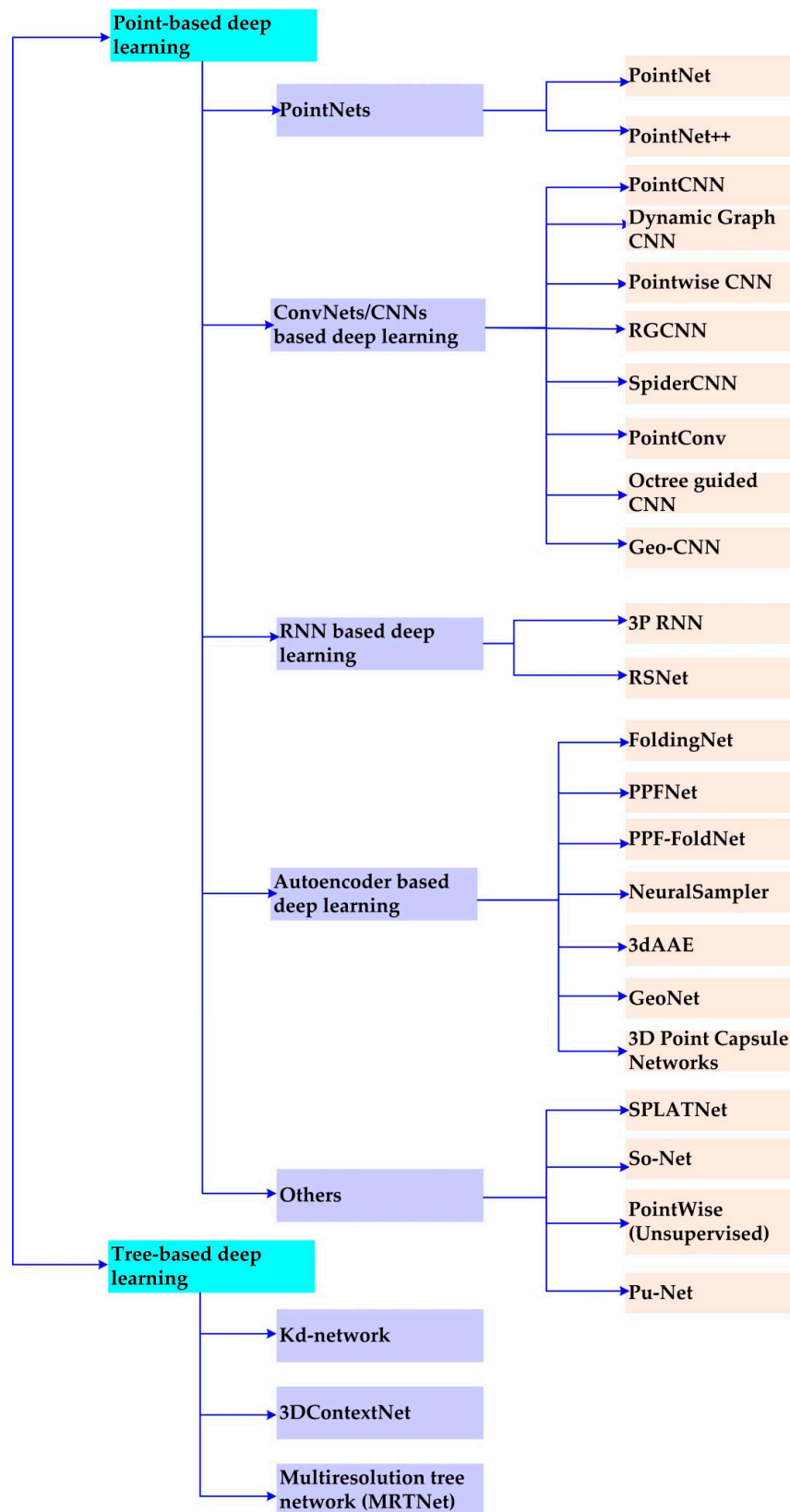


Figure 1. The main models for feature learning with raw point clouds as input.

3.1. Raw Point-Based Deep Learning

Currently, there are several models directly consuming a raw point cloud without losing information [6,43,48–58]. Based on the basic module of these models, they are divided into five categories, i.e., PointNet-based, deep convolutional neural networks (ConvNets)-based, recurrent neural networks (RNN)-based, autoencoder (AE)-based, and others as shown in Figure 1.

3.1.1. PointNet-Based Deep Learning

There are two main architectures, including PointNet [6] and PointNet++ [43] in this section. The representative work proposed by Stanford University researchers is PointNet, which is used to directly process point clouds. Since PointNet cannot capture the local features of the point clouds, PointNet++ was then proposed. PointNet was first introduced and PointNet++ followed.

PointNet is the pioneering work with raw point clouds as input for deep learning. It has been used for 3D object detection and semantic segmentation. It was proposed to address unstructured point cloud data considering the invariance of the input point cloud arrangement. Specifically, it has two core building blocks, i.e., the transformation networks (T-Net) and the symmetric function. The former is used to align the model with the input and aggregate information from each point. It uses a spatial transformation network (STN) [59] to solve the rotation problem. STN in the computer vision community was proposed to deal with spatial invariance of objects. STN learns the rotation matrix that is most conducive to network classification or segmentation by learning the attitude information of the point cloud itself. Moreover, it employs STN twice. The first input conversion is to adjust the point cloud in the space. Intuitively, the PointNet rotates out of an angle that is more conducive to sorting or segmentation, such as turning the object to the front. The second feature transformation is to align the extracted 64-dimensional features by converting the point cloud at the feature level. Max pooling is adopted as the symmetric function for processing the point cloud. Specifically, it aggregates the high-dimensional local features of each point, which is learned from multi-layer perception (MLP) [60]. It has the capability to tackle the disorder problem and the invariance under transformations. This is because the global features of the entire point clouds can be extracted through max-pooling [12].

Since the MLP only learns the local features of each point and ignores the connections between points, PointNet fails to represent the local features of neighboring points, thus limiting its performance in complicated scenes. Based on the above analysis, PointNet cannot adequately handle local feature extraction, to address this, PointNet++ was proposed by constructing a class pyramid feature aggregation scheme. It is also used for point classification and semantic segmentation. Specifically, there are two aspects for PointNet++ to encode the local features: (1) how to divide the point cloud locally and (2) how to extract local features from the point cloud. For the first aspect, hierarchical feature learning for the point cloud is proposed. It consists of three components: the sampling layer, the grouping layer, and the PointNet layer. The sampling layer selects a series of points in the input point cloud to define the center of the local area. The sampling algorithm uses iterative farthest point sampling (FPS). Especially, FPS randomly selects a point and chooses the point furthest from the point as the starting point and then continues iteration until the desired number is selected. As for the second, PointNet++ employs PointNet to extract local features after grouping the point clouds. Therefore, the original PointNet network became a subnet in the PointNet++ network, extracting features in hierarchical iterations. Even though PointNet++ can encode the local features of the point clouds, it fails to utilize the spatial distribution of the input point cloud. This is because hierarchical feature learning fails to encode the spatial distribution in the division of the point clouds.

3.1.2. ConvNets-Based Deep Learning

ConvNets is a type of feed-forward neural networks and short for deep convolutional neural networks [12,61]. Inspired by biological processes, the architecture of ConvNets is similar to the organization of the visual cortex in animals. Especially, each cortical neuron only responds to the

stimuli in the receptive field. To respond to the whole field, there is overlapping area among the receptive fields in various neurons. It is always stacked with a convolution layer, rectified linear units, and pooling layers to distill features from low-level to high-level features [12,13]. ConvNets has the benefits of shared-weights, translation invariance, and feature extraction without human interference [12]. Currently, there are seven models, including Dynamic Graph convolutional neural networks (CNN) [49], PointCNN [48], regularized graph CNN (RGCNN) [50], Pointwise CNN [62], PointConv [63], Geo-CNN [64], and SpiderCNN [65], addressing the raw point cloud. These methods bring regular representation into the network before ConvNets.

Dynamic Graph CNN is a new network for classifying and dividing point cloud data and is a modification inspired by PointNet and PointNet++. PointNet only processes each point independently to achieve permutation invariance, but it ignores local features between points. To obtain the local features, Dynamic Graph CNN includes an EdgeConv layer, which solves the local feature processing problem that PointNet does not have. PointNet++ can be compared with Dynamic Graph CNN. Different from PointNet, Dynamic Graph CNN employs EdgeConv to extract features. Specifically, the EdgeConv layer is proposed to obtain local features with the tensor of $N \times F$ (N and F are the number and the dimension of the input clouds, respectively) as the input and then be applied to each given layer $\{a_1, a_2, \dots, a_n\}$ in the MLP along the length of the output tensor to calculate the peripheral features. Finally, the merge operation is performed along adjacent peripheral features to generate a new tensor. The input includes nearby raw data X_i and nearby K points. Specifically, each point of the original data and the attached K point will be first to generate $K N \times M$ features (M is the number of the labeled classes). Then, the $N \times M$ function will output through the pool operation.

PointCNN uses hierarchical convolution and x -Conv operators to capture local information. The benefit of x -Conv is that it considers the shapes of points without focusing on the input order of the data. It has been used in 3D object classification and semantic segmentation. Similar to the space transformation network (STN) [66], K points are taken from the data of the previous layer to predict an x -transformation matrix of $K \times K$ size (x -transformation). The features in the previous layer of the x matrix are transformed, and then the transformed features are convoluted. The convolution layer in image CNN is different from the x -Conv layer in PointCNN in only two aspects, i.e., $K \times K$ region in image CNN and K adjacent points around PointCNN representing points. In addition, the deep network assembled with the x -Conv layer is not very different from the convolution layer in the Dynamic Graph CNN. It turned out that the learning ability of the model is very strong, but the generalization is not the most advanced.

RGCNN directly consumes the point clouds with irregularity and is evaluated on point cloud segmentation. It is also accessed on point clouds with high noise. It has been used for object classification and semantic segmentation in the 3D point cloud. There are two main features of RGCNN. The first feature is that RGCNN takes the features of the points as a node on the graph based on the spectral graph theory to overcome the irregularity of the point cloud. The other feature is that it introduces the convolutional operation by Chebyshev polynomial approximation for localized filtering. As for the former method, it first collects the raw features of a point, such as color and coordinates and represents each point cloud as a vector p_i and then feeds n points to the graph convolutional operation defined on the graph. As for the latter, it filters the nodes in the spectral domain and leverages Chebyshev approximation to dramatically decrease the computational complexity.

The new pointwise convolutional operation is proposed and then used to construct the architecture in Pointwise CNN. It is used to explore semantic segmentation and object classification in the point clouds. A pointwise convolution is introduced at each point. To implement segmentation and recognition, two pointwise convolutions are designed. The architecture of Pointwise CNN can be effective for learning local features because of the benefits of convolution operation, which uses a small kernel, such as the 3×3 kernel, to extract features. Unlike the traditional convolutional operation in 2D images, there is only pointwise convolution in Pointwise CNN without down-sampling or up-sampling the point clouds.

PointConv [63] is a novel convolutional operation and can be used to construct the architecture of deep convolutional neural networks addressing the irregular and unordered point clouds. It takes the coordinates of the point clouds as inputs. Especially, it is extended by the dynamic filter with non-uniform sampling. The weights in the convolution are learned by MLP, and density functions are acquired by the kernel density estimation to satisfy non-uniform sampling. This network has the scalability to deal with translation-invariant and permutation-invariant point clouds.

Inspired by the benefits of local features in the point clouds, Geo-CNN [64] aims to encode the geometric structure for a point and its corresponding neighboring point clouds through a convolutional operation. Firstly, edge features are extracted by GeoConv to encode the geometric structure with a vector and decomposed into three orthorhombic orientations. Secondly, features distilled from these directions are combined to represent the geometric structure of the point clouds with the vector and the three bases to acquire the local features.

Similar to Geo-CNN, to distill geometric features from the irregular point clouds, SpiderCNN [65] defines a novel convolutional operation. The proposed convolution is extended from the regular grids to the irregular point sets. The filter in the convolution is the product of step functions to encode the local geometric information of the point clouds. The Taylor polynomial is used to ensure the expressiveness of the SpiderCNN.

3.1.3. RNN-Based Deep Learning

A recurrent neural network (RNN) is a class of artificial neural network (ANN) where connections between nodes form a directed graph along a temporal sequence, encoding the temporal data [67]. The architecture of RNN is expressed in Figure 2, where X_i ($i = 0, 1, 2, \dots, t$) encodes the temporal data at the time i , and a_i ($i = 0, 1, 2, \dots, t$) are the inputs for the next time steps, while h_i ($i = 0, 1, 2, \dots, t$) is the output of the current time step. It is obvious that the connections between the points are a directed graph. Unlike ConvNets, RNN employs the internal states, i.e., a_i ($i = 0, 1, 2, \dots, t$), to process sequential inputs, thus making it possible to deal with the sequential tasks, such as speech recognition. It has many variations, such as Long Short-Term Memory (LSTM) [68] and bidirectional RNN [68]. Being able to capture the context, bidirectional RNN has been applied to pointwise pyramid pooling RNN (3P-RNN) [51] and recurrent slice networks (RSNets) [69] to better deal with the point clouds.

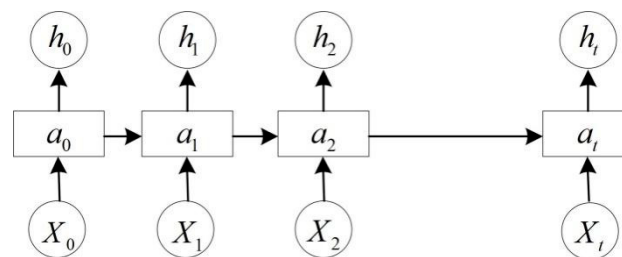


Figure 2. The architecture of a recurrent neural network (RNN).

Considering the benefits of RNN, 3P-RNN is proposed to address the semantic segmentation with raw point clouds as input. There are two main components in 3P-RNN, i.e., a pyramid pooling module and a bidirectional RNN. The former is used to extract the local spatial information, and the latter is used to acquire the global context information. 3P-RNN is inspired by PointNet as shown in Section 3.1.1. Unlike pooling in PointNet++, pointwise pyramid pooling is used to acquire the local features in 3P-RNN, which has faster speed.

RSNets is proposed to capture local structures in point clouds. The core component of the RSNets is a lightweight local dependency module. This part is the combination of the designed slice pooling layer, RNN layer, and slice unpooling layer. Specifically, the slice pooling layer is used to transform the

project features of the disorder point clouds to the ordered sequence with feature vectors to be fed to the RNN layer.

3.1.4. Autoencoder-Based Deep Learning

Autoencoders (AEs) can be used to learn the representation of given data in an unsupervised manner [70] as shown in Figure 3. It is obvious that there are three stages in an autoencoder, i.e., encoder, internal representation, and decoder. Currently, it has become widely used for generative models to represent the data. It has the capability to encode the irregularity of point clouds and address the sparsity at the up-sampling stage. Researchers are beginning to employ AEs to represent them [52,54,55]. There are seven main models as shown in Figure 1, including FoldingNet, Point Pair Feature Network (PPFNet), PPF-FoldNet, NeuralSampler [55], GeoNet [71], 3D Adversarial Autoencoder (3dAAE) [72], and 3D Point-Capsule Networks.

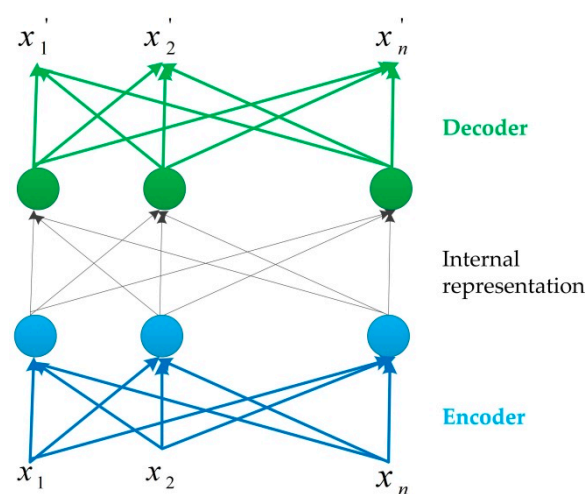


Figure 3. The architecture of an autoencoder.

FoldingNet is proposed to represent the point cloud from 2D to 3D with small reconstruction errors. Firstly, a graph-based encoder, combining MLP and a graph-based pooling layer, is used to acquire the local features. Secondly, a folding-based decoder is used to reconstruct the 3D point cloud from 2D images. As for the reconstruction error, the chamfer distance is used [73]. When it is used for classification, it achieves the best accuracy in the ModelNet40 dataset detailed in Section 4.1, Table 2.

The point pair feature network (PPFNet) is designed to learn globally 3D local features to discover the correspondences in unordered and sparse point clouds [53]. A novel N-tuple loss is employed to increase the intra-class difference and decrease the intra-class variations. Global information is injected into local descriptors. Integrating point pair features with normals, their corresponding 3D representations are calculated. It is designed to represent the local features of the raw point sets, which is sensitive to the global context. Inspired by PointNet, it also takes the permutation invariant network into consideration.

PPF-FoldNet was proposed to tackle the problem that PPFNet is sensitive to the rotation of the point clouds and was also used for unsupervised 3D local descriptors learning on the raw point clouds. Based on the well-known point-to-feature folding-based automatic coding, PPF-FoldNet has many desirable features: it does not require supervision or a sensitive local reference frame and can acquire rotation invariant descriptors.

NeuralSampler [55] addresses 3D point clouds of various sizes and has been used for object classification. It learns the feature representation by decoupling shape generation from surface sampling with a convolutional auto-encoder. The encoder is used to address the irregularity of the

point cloud and the decoder to deal with the sparsity. Especially, a latent vector representation is calculated to encode given points, such as a surface or bounding cube.

GeoNet [71] was proposed to encode the connectivity information in the point clouds. It takes surface topology and object geometry into consideration for representing the point clouds. GeoNet employs the learned topological features for a geodesic-aware point cloud analysis. There are two components in this architecture, i.e., an autoencoder to extract a feature vector for each point and a geodesic matching (GM) layer that acts as a learned kernel function for estimating geodesic neighborhoods using the latent features.

3dAAE [72] obtains the representations of 3D shapes. It has the ability of end-to-end learning the representation of 3D point clouds. This model firstly learns a latent space for 3D shapes, and then adversarial training is used to generate the output. The authors of 3dAAE extended the autoencoder to 3D, which takes the 3D data as input and generates the corresponding 3D output.

3D Point-Capsule Networks [74] were proposed to address the sparse 3D point clouds without changing spatial arrangements. Especially, an AE is designed to do this task. This network was extended from 2D capsule networks to 3D to tackle the sparsity of the point clouds. PointNet-like input layers are employed to encode the sparsity of point clouds, and then latent capsules are used to capture information not spatially but semantically across the shape.

3.1.5. Others

As stated in Section 1, there are three characteristics: unordered structure, interaction among points, and invariance under transformations. Many researchers have designed deep learning models with the raw point cloud as input. Except for the above four kinds, researchers employ special strategies to tackle the raw point dataset. For example, Self-Organizing Network (SO-Net) [57], Pointwise [58], and Pu-Net [75] use unsupervised approaches to learn the representation. SO-Net will be briefly introduced, followed by unsupervised approaches representing the point cloud.

SO-Net is a permutation-invariant network structure dealing with unordered point clouds. It utilizes the spatial distribution of the point cloud by designing a network with a constant arrangement and simulates the spatial distribution by constructing a self-organizing map (SOM) [57]. Especially, SOMs are used to acquire the hierarchical features in SO-Net. After the construction of the SOM, a feature vector is used to represent the point cloud. The point cloud automatic encoder is proposed to improve the network performance at different tasks. To maintain the order of the input point cloud, there are two core factors behind this, i.e., special network architecture and alternative SOM training. SOM does not change the topology of the input point clouds. Little information is missing before the processed point clouds feed to the network and transform the point cloud into a feature matrix, speeding up the procedure, which has tremendous advantages. There are many applications of SO-Net, including object classification, semantic segmentation, shape retrieval, etc. Due to the parallelism and simplicity of the proposed architecture, the training speed is much faster than the existing point cloud recognition network.

To calculate the hierarchical and spatial features of the point cloud, a sparse and efficient mesh filter in a lattice with high number of dimensions is proposed in Sparse Lattice Networks (SPLATNet) [56]. Similar to the architectures of ConvNets, SPLATNet makes filter neighborhoods easy to be regulated and uses hash tables to pass on only the location of the data convolved to effectively handle the sparse point cloud. It makes converting 2D points to 3D space easy and vice versa. SPLATNet uses the permutohedral lattice convolution in the Bilateral Convolution Layer, which is a generalization of bilateral filtering fusing a sparse filter into neural networks [56] to place the organization of the point cloud in each convolution operation.

To learn the point-wise description of the point cloud, [58] uses an embedding for the cloud point through neural networks. First, an embedding space is clustered in the latent space with local structures to encode the geometric information of the point cloud. Second, the semantic point analogies are derived by computing Euclidean distance. Finally, point-correspondence is obtained by retrieving

nearest-neighbors. There are two kinds of loss used in this framework, i.e., patch reconstruction loss and triplet loss. The former considers the context of the point cloud, and the latter considers that the point clouds have similar representations at the near distance and different ones at a far distance.

Pu-Net [75] is a data-driven model to learn the sparse and irregular point cloud with the raw point clouds as input. It learns the multi-level features of each point and uses the multi-branch convolution to acquire the expanded feature, which is then split to reconstruct the point cloud. There are four parts in Pu-Net, including patch extraction to acquire d point clouds with various sizes, point feature embedding to obtain the local and global geometric information of the d point clouds, feature expansion to enlarge the number of features, and coordinate reconstruction to implement the 3D coordinates of the expanded features.

Point Contextual Attention Network (PCAN) [76] is also used to encode local features. Different from PointNet++ and other neural networks, PCAN considers the task-relevant features. Especially, it first uses PointNet to extract local features and then exploits a NetVLAD layer [77] to aggregate global features. When fusing features into a discriminative global descriptor, the sampling and grouping layers in PointNet++ are first used to obtain the attention map with multi-scale contextual information, and then task-relevant features are focused.

3.2. Tree-Based Deep Learning

A Kd-tree is built on an eight-point point cloud. Nodes are numbered from root to leaf in the Kd-tree. Due to the irregularity of the point cloud, approaches based on a Kd-tree were proposed to explore the local and global context. Kd-tree based models take point clouds as regular presentations before feeding information into deep learning models. These methods gradually learn the representation vector of the point cloud along the tree. Experimental results on challenging datasets have shown that the Kd-tree provides distinguishing point cloud features. There are three methods, including the Kd-network [78], 3D contextual network (3DContextNet) [44], and Multiresolution Tree Networks (MRTNet) [46].

The Kd-network works with an unstructured point cloud and is designed for 3D model recognition tasks. The architecture performs a multiplication transformation and shares the parameters of these transformations according to the subdivision of the point cloud to which the Kd-tree applies. Unlike the main convolution architecture that typically requires rasterization on a uniform two- or three-dimensional grid, the Kd-network does not rely on such a mesh in any way, thus avoiding poor scaling behavior. The point layer features are hierarchically calculated at different levels in the feature learning phase. For a level, each point is processed using a shared multilayer perceptron network (MLP) as a function h in the equation. After that, a different local area representation is calculated for the same level of nodes by the corresponding function.

Just like the Kd-network, 3DContextNet was proposed to capture the local and global features of the point clouds using a Kd-tree structure. Different from the Kd-network defining operation on a Kd tree, 3DContextNet employs the Kd-tree to represent the 3D point clouds without changing the spatial relationships and can be used for 3D object classification and semantic segmentation. There are two main components in this architecture, i.e., feature learning at multi-scale and feature aggregation to extract global contextual information.

Different from Kd-network and 3DContextNet, the point clouds are first sorted using the Kd-tree in MRTNet [46]. The Kd-tree used can represent the point clouds in a hierarchical and locality-preserving order [46]. Especially, the pooling operation defined in [46] can be used to construct the hierarchical sorting, and multiresolution scaling of the point clouds is useful for preserving the locality. Since the Kd-tree partitions the point clouds, the dependence among them is no longer kept. After the point clouds are sorted, 1D convolution and pooling are used to build the MRTNet. Experimental results on shape classification reveal the MRTNet has the benefits of small memory cost and fast convergence speed during training. MRTNet can also be used as an encoder and decoder for shape generation.

4. Applications of Point Clouds Using Deep Learning

There are numerous applications of the models mentioned in Section 3, which directly take the raw point cloud as input. Here, we mainly focus on three aspects, 3D object classification, semantic segmentation, and 3D object detection. First, the datasets used to evaluate the performance of the models in Section 3 are shown, and then evaluation indicators and performances of the reviewed methods regarding the three applications in each application are provided.

4.1. Datasets

Datasets can be divided into two categories: indoor datasets by Kinect and outdoor datasets typically obtained by 3D scanners such as LIDAR. These public datasets make it possible to compare and access various models and analyze their advantages and disadvantages. The available datasets and their descriptions and application tasks are shown in Table 2.

Table 2. Available point cloud datasets for classification, segmentation, and object detection.

Datasets Name	Descriptions	Application Tasks
ModelNet40 [18]	It consists of 12,311 CAD models in 40 object classes.	3D object classification [48,50,51,79] and shape classification [45]
ShapeNet part [80]	There are 16,881 shapes represented by 3D CAD models in 16 categories with a total of 50 parts annotated.	Part segmentation [44,48,49,56,80], shapes generation [46], and representation learning [52]
Stanford 3D semantic parsing [81]	This dataset has 271 rooms in six areas captured by 3D Matterport scanners captured by Matterport Camera.	Semantic Segmentation [8,43,44,46,48,49,78,82]
SHREC15 [18]	There are 1200 shapes in 50 categories by scanning real human participants and using 3D modeling software [79]. Each class has 24 shapes and most of these shapes are organic with different postures.	Non-rigid shape classification [43]
SHREC16 [18]	It contains about 51,300 3D models in 55 categories.	3D shape retrieval [8]
ScanNet [83]	There are 1513 scanned and reconstructed indoor scenes.	Virtual scan generation [43], segmentation [48], and classification [48]
S3DIS [81]	It consists of 271 rooms in six areas captured by 3D Matterport scanners.	3D semantic segmentation [44,48] and representation
TU-Berlin [84]	It has sketches in 250 categories. Each category has 80 sketches.	Classification [48]
ShapeNetCore [85]	It has 51,300 3D shapes in 55 categories, which is indicated by triangular meshes. The dataset is labeled manually and a subset of the ShapeNet dataset.	3D shape retrieval task [78], 3D shape retrieval task [8], and classification [8]
ModelNet10 [18]	The 10-class of Model-Net (ModelNet10) benchmarks are used for 3D shape classification. They contain 4,899 and 12,311 models respectively.	Object classification [8] Shape classification [78]
RueMonge2014 [86]	The images are multi-view in high-resolution images from a street in Paris and the number of these images is 428.	3D point cloud labeling [56]
3DMatch Benchmark [87]	It contains a total of 62 scenes.	Point Cloud representation [54]
KITTI-3D Object Detection [88,89]	There are 16 classes, including 40,000 objects in 12,000 images captured by a Velodyne laser scanner.	3D object detection [20,23,24,90]
vKITTI [91]	This dataset includes a sparse point cloud captured by LiDAR without color information. It can be used for generalization verification, but it cannot be used for supervised training.	Semantic segmentation [51]

Table 2. Cont.

Datasets Name	Descriptions	Application Tasks
3DRMS [92]	This dataset comes from the challenge of combining 3D and semantic information in complex scenarios and was captured by a robot that drove through a semantically rich garden with beautiful geometric details.	Semantic segmentation [51]
Cornell RGBD Dataset	It has 52 labeled point cloud indoor scenes including 24 office scenes and 28 family scenarios with the Microsoft Kinect sensor. The data set consists of approximately 550 views with 2495 segments marked with 27 object classes.	Segmentation [14]
VMR-Oakland dataset	It contains point clouds captured by mobile platforms with Navlab11 around the Carnegie Mellon University (CMU) campus.	Segmentation [14]
Robot 3D Scanning Repository	The 3D point clouds acquired by Cyberware 3030 MS are provided for both indoor and outdoor environments. Heat and color information is included in some datasets.	Segmentation [14]
ATG4D [89]	There are over 1.2 million, 5,969, and 11,969 frames in the training, validation, and test datasets, respectively. This dataset is captured by a PrimeSense sensor.	Point object detection [20]
Paris-Lille-3D [60]	There are 50 classes in 143.1M point clouds acquired by Mobile Laser Scanning.	Segmentation and classification [60]
Semantic3D [93]	There are eight classes in 1660M point clouds acquired by static LIDAR scanners.	Semantic segmentation [93]
Paris-rueMadame [94]	There are 17 classes in 20M point clouds acquired by static LIDAR.	Segmentation, classification, and detection [94]
IQmulus [61]	There are 22 classes in 12M point clouds acquired by static LIDAR.	Classification and detection [61]
MLS 1 - TUM City Campus [95,96]	There are more than 16,000 scans captured by mobile laser scanning (MLS) in this dataset.	3D detection [95,96], city modeling [95,96], and 3D change detection

4.2. 3D Object Classification

The goal of 3D object classification is to recognize objects from a 3D point cloud [26,97–99], i.e., to provide a semantic object label to a separated point cloud. It has numerous applications in robotics, virtual reality, and city planning. Currently, there are several available datasets for 3D object classification in the point cloud, such as ModelNet40 and TU-Berlin as shown in Table 2. The challenges of data-related classification have three aspects, including missing data, noise, and rotation invariance.

- Missing data: Scanned models are usually occluded, and some data is lost.
- Noise: All sensors are noisy. There are different types of noise, including point perturbations and outliers. This means that a point has a certain probability within a certain radius around the location where it is sampled (disturbance), or it may appear at random locations (outliers) in space.
- Rotation invariance: Rotation and translation points should not affect classification.

Accuracy is usually used to evaluate a classification model. In general, accuracy refers to the proportion of the model that predicts the correct outcome. Formally, accuracy is defined as in Formula (1) [12]. As for the error rate, it is the misclassification rate and equal to one minus accuracy, as shown in Formula (2).

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$\text{Error rate} = 1 - \text{accuracy} \quad (2)$$

where TP , TN , FP , and FN are the true positive case, true negative case, false-positive case, and false-negative case, respectively.

3D object classification is receiving more and more attention and has become a very active research field. Several methods can be used for classification, such as PointNet, PointNet++, SO-Net, Dynamic Graph CNN, PointCNN, Kd-Network, 3DContextNet, Multi-Resolution Tree Network, SPLATNet, FoldingNet, and NeuralSampler. Even though there are many datasets available, the widely used datasets to access the performance of various models are ModelNet 10 and ModelNet 40. The classification performance collected from the published literatures on point cloud with these models is shown in Table 3. Class accuracy and instance accuracy are the accuracies regarding class and instance, respectively.

Table 3. Classification performance on point cloud with different models.

Methods	ModelNet 10		ModelNet 40		Training
	Class Accuracy	Instance Accuracy	Class Accuracy	Instance Accuracy	
PointNet [6]	-	-	86.2	89.2	3–6 h
PointNet++ [43]	-	-	-	91.9	20 h
Deepsets [100]	-	-	-	90.0	-
SO-Net [57]	95.5	95.7	90.8	93.4	3 h
Dynamic Graph CNN [49]	-	-	-	92.2	-
PointCNN [48]	-	-	91.7	-	-
Kd-Net [78]	93.5	94.0	88.5	91.8	120 h
3DContextNet [44]	-	-	-	91.1	-
MRTNet [46]	-	-	-	91.7	-
SPLATNet [56]	-	-	83.7	86.4	-
FoldingNet [95]	-	94.4	-	88.4	-
NeuralSampler [55]	-	95.3	-	88.7	-

4.3. Semantic Segmentation

A point cloud is a collection of data points. It can be represented as a group, where each point can be represented by a vector, including its coordinates and additional feature channels. Once the point cloud is segmented, each segment (group) of points can be marked with a class, providing some semantics to the segment. The aim of point cloud semantic segmentation task [14,15,26,40,99] is to label each point in a point set with its corresponding semantically meaningful category.

The point cloud semantic segmentation algorithm should have three attributes:

- The segmentation algorithm should consider the specific properties of different ground objects.
- The segmentation algorithm should infer the attribute relationships of adjacent partition blocks.
- The segmentation algorithm should be applied to the point clouds acquired by different scanners.

The evaluation indicator is intersection over union (IoU) [12], a measuring accuracy of detecting corresponding objects, and is defined in Formula (3). The numerator in Formula (3) is the overlapping area between the predicted bounding box (A) and the ground-truth bounding box (B), and the denominator is the area encompassed by both A and B .

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} \quad (3)$$

The applications of point cloud segmentation include smart vehicles, autonomous mapping, navigation, etc. There are many methods that can be used for segmentation, such as PointNet, PointNet++, SO-Net, Dynamic Graph CNN, Kd-Network, 3DContextNet, Multiresolution Tree Networks, and SPLATNet. Considering the popularity, the ShapeNet part dataset was selected to evaluate the performance of these models because many approaches exploit it. The evaluation performance for segmentation of point clouds collected from the published literatures is shown in Tables 4 and 5 on the ShapeNet part dataset.

Table 4. Evaluation performance regarding for semantic segmentation on the ShapeNet part dataset [6,45,46,48,51,58,59,81].

	Intersection over Union (IoU)								
	Mean	Air- Place	Bag	Cap	Car	Chair	Ear- Phone	Guitar	Knife
PointNet [6]	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9
PointNet++ [43]	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9
SO-Net [57]	84.6	81.9	83.5	84.8	78.1	90.8	72.2	90.1	83.6
Dynamic Graph CNN [49]	85.1	84.2	83.7	84.4	77.1	90.9	78.5	91.5	87.3
Kd-Net [78]	82.3	80.1	74.6	74.3	70.3	88.6	73.5	90.2	87.2
3DContextNet [44]	84.3	83.3	78.0	84.2	77.2	90.1	73.1	91.6	85.9
MRTNet [46]	79.3	81.0	76.7	87.0	73.8	89.1	67.6	90.6	85.4
SPLATNet [56]	83.7	85.4	83.2	84.3	89.1	80.3	90.7	75.5	93.1

Table 5. Evaluation for segmentation for semantic segmentation on point cloud on ShapeNet part dataset [6,45,46,48,51,58,59,81].

	Intersection over Union (IoU)								
	Mean	Lamp	Laptop	Motor	Mug	Pistol	Rocket	Skate	Table
PointNet [6]	83.7	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
PointNet++ [43]	85.1	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
SO-Net [57]	84.6	82.3	95.2	69.3	94.2	80.0	51.6	73.1	82.6
Dynamic Graph CNN [49]	85.1	82.9	96.0	67.8	93.3	82.6	59.7	75.5	82.0
Kd-Net [78]	82.3	81.0	94.9	57.4	86.7	78.1	51.8	69.9	80.3
3DContextNet [44]	84.3	81.4	95.4	69.1	92.3	81.7	60.8	71.8	81.4
MRTNet [46]	79.3	80.6	95.1	64.4	91.8	79.7	57.0	69.1	80.6
SPLATNet [56]	83.7	83.9	96.3	75.6	95.8	83.8	64.0	75.5	81.8

4.4. 3D Object Detection

Unlike object classification, 3D object detection in point clouds not only assigns the labels to point sets but also locates the objects of interest with bounding boxes in 3D. It becomes a challenging problem due to its discrete sampling, noise scanning, occlusion, and cluttered scenes. Compared with 3D object classification and semantic segmentation, 3D object detection with a raw point cloud is still less explored. The reasons may be the lack of large labeled point dataset. Currently, the dataset used for object detection is mainly from optical images, such as VOC2007 [101] and COCO [66]. For point clouds, the widely used dataset is KITTI [102]. Considering that only a few models consume raw point clouds directly, we provide the related works, i.e., PointRCNN [103], VoxelNet [104], MVX-Net [105], FVNet [106], F-PointNet [107], and a deep Hough voting model [108].

There are some evaluation indicators that can be used for object detection, such as Precision, Recall, F_1 score, average precision (AP), and mean average precision (mAP) as expressed by Formulas (4)–(8) [108–114], respectively. Precision represents the proportion of all identified correct instances. That is to say, the recall represents the proportion of all true positive examples in the sample, and these examples are correct positive examples.

In theory, the AP should be an area surrounded by a precise recall curve and two axes. This is the integral of the precision–recall curve. The AP summarizes the shape of the precision–recall curve and is defined as the mean precision at a set of equally spaced recall levels. AP measures the quality of the learning model in each category, while mAP measures the quality of the learning model in all categories. After obtaining the AP, the calculation of the mAP (the average value of all APs) becomes very simple as shown in Formula (8).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

$$\text{AP} = \frac{1}{N} \sum_r P_{\text{interp}}(\text{Recall}') = \max_{\text{Recall}' > \text{Recall}} P(\text{Recall}') \quad (7)$$

$$\text{mAP} = \frac{1}{N_{\text{total}}} \sum \text{AP} \quad (8)$$

where N is the total number of equally spaced recall levels in Formula (7), and a value of 11 is usually used for N in practice. P_{interp} is the precision at each recall level Recall' , and is interpolated by taking the maximum precision measured for a method for which the corresponding recall exceeds Recall' . N_{total} is the total number of object categories in Formula (8).

Since the dataset KITTI is a publicly available point cloud, it was used to evaluate different models. mAP is widely used to evaluate the performance of models in 3D object detection and was selected as the indicator, especially, for the dataset with only the 'Car' category. ScanNet and SUN RGB-D were also used. The experimental results collected from the published literatures are shown in Table 6. PointRCNN encoding the multi-scale local and rotation invariance achieves the top performance for the KITTI dataset with only the 'Car' category.

Table 6. Point cloud object detection results [93,110]. $\text{mAP}_{\text{ScanNet}}$, $\text{mAP}_{\text{SUN RGB-D}}$, and mAP_{3D} results on ScanNet, SUN RGB-D, and KITTI datasets with only the 'Car' category.

Model	Feature Extraction	$\text{mAP}_{\text{ScanNet}}$	$\text{mAP}_{\text{SUN RGB-D}}$	mAP_{3D}		
				Easy	Moderate	Hard
FVNet [110]	PointNet	-	-	65.43	57.34	51.85
VoxelNet [108]	-	-	-	81.97	65.46	62.85
PointRCNN [107]	PointNet++, multi-scale grouping	-	-	88.88	78.63	77.38
F-PointNet [111]	PointNet++	-	-	81.20	70.39	62.19
MVX-Net [109]	VoxelNet	-	-	83.20	72.70	65.20
Deep Hough voting model [112]	PointNet++	46.80	57.70	-	-	-

5. Discussion and Future Direction

Considering point clouds are unstructured and in disorder, especially non-Euclidean and sparse data [26], it is necessary to encode their information as completely as possible. PointNet is the first approach to deal with point clouds based on raw inputs and achieves promising results for 3D object classification and semantic segmentation. Following this, architectures from deep learning, including RNN, AE, CNN, RNN, and generative adversarial networks (GAN) [12] are introduced. Furthermore, the Kd-tree is introduced in the point clouds. Models with the raw input are surveyed, and three typical applications, including 3D object classification, semantic segmentation, and 3D object detection, are summarized. Related datasets and evaluation metrics are introduced. In this section, we will first discuss the performance, strengths, and weaknesses of the reviewed methods, and then propose some future directions.

5.1. Performance and Characteristics of Reviewed Methods

For 3D object classification, PointNet fails to extract the local features and only uses global features directly to obtain the probability for each class. From Table 3, we can see that the SO-Net achieves best classification performance on ModelNet 10 and ModelNet 40. The excellent performance stems from its powerful network. This may be to the special architecture of SO-Net. So-Net captures local features, global features, and a topological order of input points. Even in unsupervised learning of the point clouds, the models being able to extract local features, global features, and geometry of the point clouds have a better performance as shown in Table 3. Therefore, it is beneficial to incorporate the raw point clouds into the neural networks and also make full use of them without missing information.

For semantic segmentation, as shown in Tables 4 and 5, it is obvious that PointNet++ and Dynamic Graph CNN achieve top performance with the mean IoU. Both PointNet++ and Dynamic Graph CNN consider the local features, which benefits the segmentation results. SPLATNet achieves about 5% higher scores over several classes, such as Knife, Ear-phone, Car, and Motor, because it employs the spatial distribution of the point clouds. Based on these analyses, integrating the local and global features extracted by deep learning models with the spatial representation of the point clouds will be useful to design a model for semantic segmentation with top performance.

For 3D object detection, as shown in Table 6, we can see that compared with other models PointRCNN can detect examples in the car class of KITTI with a higher AP. This can be attributed to its direct representation of the point cloud. It directly generates proposals from the point clouds instead of projecting them to bird's eye view or voxels. These models show promising results for dealing with raw point clouds, encoding the point clouds that are missing little or no information.

5.2. Some Future Directions

From the application aspect, the models considering the spatial distribution, maintaining the topological order of input points, and extracting both global and local hierarchical features achieve the top performance. Based on those attributes contributing to a model with top performance, the further designed model should have representation power, including the spatial distribution of the whole point cloud, the topological order of input points, the global and local hierarchical features, and sparse representation. For example, one can encode the point cloud fed into the 3D neural networks. Despite much work having been done, compared with that of RGB images, the performances of methods based on point cloud processing networks for 3D object classification, semantic segmentation, and 3D object detection are still quite low. This difference due to the special inherent characteristics of the point cloud, i.e., irregular and sparse. Thus, there is still much work to conduct. Some of the aspects are stated in the following.

A promising solution is to address the raw point clouds with the ConvNets. Since ConvNets has the advantage of overlapping during convolutional operation [115–117], it may benefit the future architecture of deep learning models for the point cloud to take the characteristics, i.e., interaction among points, into consideration. Usually, ConvNets are used to extract multi-scale semantic features. Then, specific modules are designed for different applications. Taking semantic segmentation as an example, multi-scale features fused with skipped connections are often employed to obtain high performance, such as U-Net [31]. Recently, [118] designed a multi-resolution network for multi-scale point cloud processing and reported a 3.4% increase in IoU.

Another promising direction is to develop the architectures of the deep learning models like those in RGB images. There are many kinds of well-designed convolutional operations, such as residual module in ResNet [29] to extend the depth of the neural networks without losing accuracy, inception in GoogLeNet [27] to enlarge the width of the model with few parameters to be learned, and feature pyramid networks (FPN) [119] to extract multi-scale features. Various kinds of loss functions are also developed to train the models, such as focal loss [37] to balance the positive and negative examples and pay attention to hard examples. Since these ideas boost the application of deep models, it may be useful to design the models with the inherent characteristics of the raw point clouds in mind, such as irregular, sparse, and disorderly. For example, one can incorporate the sparse representation into the loss function to train the deep learning models for the point cloud.

Finally, zero-shot learning [115] is also an exciting topic for deep learning models directly processing raw point clouds. After obtaining the feature maps, it uses a semantic embedding for applications such as object detection. Moreover, it has the capability to recognize the unobserved class in the trained dataset. Since PointNet and EdgeConv extract global and local features of the point clouds, they can be used as feature extractors in zero-shot learning. It will facilitate learning the weights with a scarce dataset, especially in point clouds.

6. Conclusions

The recent existing feature learning approaches with the raw point clouds as input are classified as point-based and tree-based approaches. This survey of point cloud deep learning has a rich bibliographical content that can provide valuable insights on this important topic and encourage new research. Firstly, deep feature learning methods for raw point clouds are classified and reviewed, and the pros and cons of these methods are also analyzed. Secondly, the datasets and models with top performance regarding the applications in 3D object classification, semantic segmentation, and 3D object detection were investigated. Finally, some future directions, including model design based on ConvNets, incorporation of the inherent characteristics of point clouds with the networks, and zero-shot learning models after feature extraction by PointNet and EdgeConv, are proposed.

Author Contributions: W.L. mainly conceived the manuscript and organized the whole paper. W.L. supervised the research and participated in the discussion of the paper. J.S., T.H., and P.W. contributed to the organization of the paper and revised the paper.

Funding: This research was funded by National Natural Science Foundation of China (Nos. 61771471, 91748131, and U1613213), National Key Research and Development Plan of China under Grant 2017YFB1300202, China Postdoctoral Science Foundation (No. 2018M641523) and Youth Innovation Promotion Association Chinese Academy of Sciences (CAS) (No. 2015112).

Acknowledgments: The authors own many thanks to the anonymous reviewers for their instructive comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Balsabarreiro, J.; Fritsch, D. Generation of Visually Aesthetic and Detailed 3d Models of Historical Cities by Using Laser Scanning and Digital Photogrammetry. *Digit. Appl. Archaeol. Cult. Herit.* **2017**, *8*, 57–64.
- Balsa-Barreiro, J.; Fritsch, D. Generation of 3d/4d Photorealistic Building Models. The Testbed Area for 4d Cultural Heritage World Project: The Historical Center of Calw (Germany). In *Advances in Visual Computing, Proceedings of the 11th International Symposium (ISVC 2015), Las Vegas, NV, USA, 14–16 December 2015*; Springer: Berlin/Heidelberg, Germany, 2015.
- Oliveira, M.; Lopes, L.S.; Lim, G.H.; Kasaei, S.H.; Tomé, A.M.; Chauhan, A. 3D object perception and perceptual learning in the RACE project. *Robot. Auton. Syst.* **2016**, *75*, 614–626. [[CrossRef](#)]
- Mahler, J.; Matl, M.; Satish, V.; Danielczuk, M.; Deroose, B.; McKinley, S.; Goldberg, K. Learning ambidextrous robot grasping policies. *Sci. Robot.* **2019**, *4*, eaau4984. [[CrossRef](#)]
- Velodyne Hdl-64e Lidar Specification. Available online: <https://velodynelidar.com/hdl-64e.html> (accessed on 5 May 2019).
- Charles, R.Q.; Su, H.; Kaichun, M.; Guibas, L.J. Pointnet: Deep Learning on Point Sets for 3d Classification and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017*; pp. 77–85.
- Aubry, M.; Schlickewei, U.; Cremers, D. The Wave Kernel Signature: A Quantum Mechanical Approach to Shape Analysis. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV 2011 Workshops), Barcelona, Spain, 6–13 November 2011*.
- Bronstein, M.M.; Kokkinos, I. Scale-Invariant Heat Kernel Signatures for Non-Rigid Shape Recognition. In *Proceedings of the 23 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010), San Francisco, CA, USA, 13–18 June 2010*.
- Hana, X.-F.; Jin, J.S.; Xie, J.; Wang, M.-J.; Jiang, W. A Comprehensive Review of 3d Point Cloud Descriptors. *arXiv* **2018**, arXiv:1802.02297.
- Guo, Y.; Bennamoun, M.; Sohel, F.; Lu, M.; Wan, J. 3D Object Recognition in Cluttered Scenes with Local Surface Features: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2270–2287.
- Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet Classification with Deep Convolutional Neural Networks. In *Proceedings of the International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012*.
- Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; Adaptive Computation and Machine Learning; MIT Press: London, UK, 2016.

13. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)]
14. Grilli, E.; Menna, F.; Remondino, F. A review of point clouds segmentation and classification algorithms. *ISPRS Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2017**, *42*, 339–344. [[CrossRef](#)]
15. Nguyen, A.; Le, B. 3d Point Cloud Segmentation: A Survey. In Proceedings of the IEEE 6th International Conference on Robotics, Automation and Mechatronics (RAM 2013), Manila, Philippines, 12–15 November 2013.
16. Ahmed, E.; Saint, A.; Shabayek, A.E.R.; Cherenkova, K.; Das, R.; Gusev, G.; Aouada, D.; Ottersten, B. Deep Learning Advances on Different 3d Data Representations: A Survey. *arXiv* **2018**, arXiv:1808.01462.
17. Kaick, V.; Oliver, Z.H.; Hamarneh, G.; Cohen-Or, D. A Survey on Shape Correspondence. In Proceedings of the Eurographics 2010—State of the Art Reports, Norrköping, Sweden, 3–7 May 2010.
18. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3d Shapenets: A Deep Representation for Volumetric Shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), Boston, MA, USA, 7–12 June 2015.
19. Maturana, D.; Scherer, S. Voxnet: A 3d Convolutional Neural Network for Real-Time Object Recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2015), Hamburg, Germany, 28 September–2 October 2015.
20. Qi, R.C.; Su, H.; Nießner, M.; Dai, A.; Yan, M.; Guibas, L.J. Volumetric and Multi-View Cnns for Object Classification on 3d Data. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA, 27–30 June 2016.
21. Riegler, G.; Ulusoy, A.O.; Geiger, A. Octnet: Learning Deep 3d Representations at High Resolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 21–26 July 2017.
22. Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E. Multi-View Convolutional Neural Networks for 3d Shape Recognition. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV 2015), Santiago, Chile, 7–13 December 2015.
23. Savva, M.; Yu, F.; Su, H.; Aono, M.; Chen, B.; Cohen-Or, D.; Deng, W.; Su, H.; Bai, S.; Bai, X. Shrec'16 Track Large-Scale 3d Shape Retrieval from Shapenet Core55. In Proceedings of the Eurographics 2016 Workshop on 3D Object Retrieval, Lisbon, Portugal, 8 May 2016.
24. Fang, Y.; Xie, J.; Dai, G.; Wang, M.; Zhu, F.; Xu, T.; Wong, E. 3d Deep Shape Descriptor. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015), Boston, MA, USA, 7–12 June 2015.
25. Ioannidou, A.; Chatzilari, E.; Nikolopoulos, S.; Kompatsiaris, I. Deep Learning Advances in Computer Vision with 3d Data: A Survey. *ACM Comput. Surv. CSUR* **2017**, *50*, 20. [[CrossRef](#)]
26. Nygren, P.; Jasinski, M. A Comparative Study of Segmentation and Classification Methods for 3d Point Clouds. Master's Thesis, Chalmers University of Technology and University of Gothenburg, Gothenburg, Sweden, 2016.
27. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), Boston, MA, USA, 7–12 June 2015.
28. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
29. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
30. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Santiago, Chile, 7–13 December 2015.
31. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015.
32. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected Crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [[CrossRef](#)]
33. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014.

34. Girshick, R.; Cnn, R.F. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
35. Liu, X.; Wang, B.; Xu, X.; Liang, J.; Ren, J.; Wei, C. Modified Nearest Neighbor Fuzzy Classification Algorithm for Ship Target Recognition. In Proceedings of the Industrial Electronics and Applications, Langkawi, Malaysia, 25–28 September 2011.
36. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A.C. Ssd: Single Shot Multibox Detector. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherland, 11–14 October 2016.
37. Lin, T.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
38. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
39. Redmon, J.; Farhadi, A. Yolo9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
40. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Garcia-Rodriguez, J. A Review on Deep Learning Techniques Applied to Semantic Segmentation. *arXiv* **2017**, arXiv:1704.06857.
41. Bronstein, M.M.; Bruna, J.; LeCun, Y.; Szlam, A.; VanderGheynst, P. Geometric Deep Learning: Going beyond Euclidean data. *IEEE Signal Process. Mag.* **2017**, *34*, 18–42. [[CrossRef](#)]
42. Griffiths, D.; Boehm, J. A Review on Deep Learning Techniques for 3D Sensed Data Classification. *Remote. Sens.* **2019**, *11*, 1499. [[CrossRef](#)]
43. Qi, R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
44. Zeng, W.; Gevers, T. 3dcontextnet: Kd Tree Guided Hierarchical Learning of Point Clouds Using Local Contextual Cues. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 2–6 June 2018.
45. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. In Proceedings of the 4th International Conference on Learning Representations (ICLR 2016), San Juan, Puerto Rico, 2–4 May 2016.
46. Gadelha, M.; Wang, R.; Maji, S. Multiresolution Tree Networks for 3d Point Cloud Processing. In Proceedings of the Computer Vision—ECCV 2018—15th European Conference, Munich, Germany, 8–14 September 2018.
47. Zou, X.; Cheng, M.; Wang, C.; Xia, Y.; Li, J. Tree Classification in Complex Forest Point Clouds Based on Deep Learning. *IEEE Geosci. Remote. Sens. Lett.* **2017**, *14*, 2360–2364. [[CrossRef](#)]
48. Li, Y.; Bu, R.; Sun, M.; Pointcnn, B.C. PointCNN: Convolution On X-Transformed Points. In Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018 (NeurIPS 2018), Montréal, QC, Canada, 3–8 December 2018.
49. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic Graph Cnn for Learning on Point Clouds. *arXiv* **2018**, arXiv:1801.07829.
50. Te, G.; Hu, W.; Zheng, A.; Guo, Z. Rgcnn: Regularized Graph Cnn for Point Cloud Segmentation. In Proceedings of the 2018 ACM Multimedia Conference on Multimedia Conference (MM 2018), Seoul, Korea, 22–26 October 2018.
51. Ye, X.; Li, J.; Huang, H.; Du, L.; Zhang, X. 3d Recurrent Neural Networks with Context Fusion for Point Cloud Semantic Segmentation. In Proceedings of the Computer Vision—ECCV 2018—15th European Conference, Munich, Germany, 8–14 September 2018.
52. Yang, Y.; Feng, C.; Shen, Y.; Tian, D. Foldingnet: Point Cloud Auto-Encoder Via Deep Grid Deformation. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, 18–22 June 2018.
53. Deng, H.; Birdal, T.; Ilic, S. Ppfnet: Global Context Aware Local Features for Robust 3d Point Matching. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, 18–22 June 2018.
54. Deng, H.; Birdal, T.; Ilic, S. Ppf-Foldnet: Unsupervised Learning of Rotation Invariant 3d Local Descriptors. In Proceedings of the Computer Vision—ECCV 2018—15th European Conference, Munich, Germany, 8–14 September 2018.

55. Remelli, E.; Baqué, P.; Fua, P. Neuralsampler: Euclidean Point Cloud Auto-Encoder and Sampler. *arXiv* **2019**, arXiv:1901.09394.
56. Su, H.; Jampani, V.; Sun, S.D.; Maji, E.; Yang, Mi.; Kautz, J. Splatnet: Sparse Lattice Networks for Point Cloud Processing. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, 18–22 June 2018.
57. Li, J.; Chen, B.M.; Lee, G.H. So-Net: Self-Organizing Network for Point Cloud Analysis. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
58. Shoef, M.; Fogel, S.; Cohen-Or, D. Pointwise: An Unsupervised Point-Wise Feature Learning Network. *arXiv* **2019**, arXiv:1901.04544.
59. Sun, J.; Ovsjanikov, M.; Guibas, L. A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. In *Computer Graphics Forum*; Blackwell: Oxford, UK, 2009.
60. Roynard, X.; Deschaud, J.-E.; Goulette, F. Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *Int. J. Robot. Res.* **2018**, *37*, 545–557. [CrossRef]
61. Vallet, B.; Brédif, M.; Serna, A.; Marcotegui, B.; Paparoditis, N. Terramobilita/Iqmulus Urban Point Cloud Analysis Benchmark. *Comput. Graph.* **2015**, *49*, 126–133. [CrossRef]
62. Binh-Son, H.; Tran, Mi.; Yeung, Sa. Pointwise Convolutional Neural Networks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, 18–22 June 2018.
63. Wu, W.; Qi, Z.; Li, F. Pointconv: Deep Convolutional Networks on 3d Point Clouds. *arXiv* **2018**, arXiv:1811.07246.
64. Lan, S.; Yu, R.; Yu, G.; Davis, L.S. Modeling Local Geometric Structure of 3d Point Clouds Using Geo-Cnn. *arXiv* **2018**, arXiv:1811.07782.
65. Xu, Y.; Fan, T.; Xu, M.; Zeng, L.; Qiao, Y. SpiderCNN: Deep Learning on Point Sets with Parameterized Convolutional Filters. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.
66. Karargyris, A. Color Space Transformation Network. In *Computer Science*; National Library of Medicine: Bethesda, MD, USA, 2015.
67. Recurrent Neural Network. Available online: https://en.wikipedia.org/wiki/Recurrent_neural_network (accessed on 5 October 2018).
68. Sak, H.; Senior, A.; Beaufays, F. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling. In Proceedings of the 15th Annual Conference of the International Speech Communication Association (INTERSPEECH 2014), Singapore, 14–18 September 2014.
69. Huang, Q.; Wang, W.; Neumann, U. Recurrent Slice Networks for 3d Segmentation of Point Clouds. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, 18–22 June 2018.
70. Louizos, C.; Swersky, K.; Li, Y.; Welling, M.; Zemel, R. The Variational Fair Autoencoder. In Proceedings of the 4th International Conference on Learning Representations (ICLR 2016), San Juan, Puerto Rico, 2–4 May 2016.
71. He, T.; Huang, H.; Yi, L.; Zhou, Y.; Wu, C.; Wang, J.; Soatto, S. Geonet: Deep Geodesic Networks for Point Cloud Analysis. *arXiv* **2019**, arXiv:1901.00680.
72. Zamorski, M.; Zięba, M.; Nowak, R.; Stokowiec, W.; Trzciński, T. Adversarial Autoencoders for Generating 3d Point Clouds. *arXiv* **2018**, arXiv:1811.07605.
73. Paoletti, M.; Haut, J.; Plaza, J.; Plaza, A. A new deep convolutional neural network for fast hyperspectral image classification. *ISPRS J. Photogramm. Remote. Sens.* **2018**, *145*, 120–147. [CrossRef]
74. Zhao, Y.; Birdal, T.; Deng, H.; Tombari, F. 3d Point-Capsule Networks. *arXiv* **2018**, arXiv:1812.10775.
75. Yu, L.; Li, X.; Fu, C.; Cohen-Or, D.; Heng, P. Pu-Net: Point Cloud Upsampling Network. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
76. Zhang, W.; Xiao, C. Pcan: 3d Attention Map Learning Using Contextual Information for Point Cloud Based Retrieval. *arXiv* **2019**, arXiv:1904.09793.
77. Gronat, P.; Sivic, J.; Arandjelovic, R.; Torii, A.; Pajdla, T. Netvlad: Cnn Architecture for Weakly Supervised Place Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016.

78. Klokov, R.; Lempitsky, V. Escape from Cells: Deep Kd-Networks for the Recognition of 3d Point Cloud Models. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
79. Pickup, D.; Sun, X.; Rosin, P.L.; Martin, R.R.; Cheng, Z.; Lian, Z.; Aono, M.; Ben Hamza, A.; Bronstein, A.; Bronstein, M.; et al. Shape Retrieval of Non-rigid 3D Human Models. *Int. J. Comput. Vis.* **2016**, *120*, 169–193. [[CrossRef](#)]
80. Yi, L.; Kim, V.G.; Ceylan, D.; Shen, I.; Yan, M.; Su, H.; Lu, C.; Huang, Q.; Sheffer, A.; Guibas, L. A Scalable Active Framework for Region Annotation in 3d Shape Collections. *ACM Trans. Graph.* **2016**, *35*, 210. [[CrossRef](#)]
81. Armeni, I.; Sener, O.; Zamir, A.R.; Jiang, H.; Brilakis, I.; Fischer, M.; Savarese, S. 3d Semantic Parsing of Large-Scale Indoor Spaces. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
82. Garcia-Garcia, A.; Gomez-Donoso, F.; Garcia-Rodriguez, J.; Orts-Escolano, S.; Cazorla, M.; Azorin-Lopez, J. Pointnet: A 3d Convolutional Neural Network for Real-Time Object Class Recognition. In Proceedings of the 2016 International Joint Conference on Neural Networks, Vancouver, BC, Canada, 24–29 July 2016.
83. Dai, A.; Chang, A.X.; Savva, M.; Halber, M.; Funkhouser, A.T.; Nießner, M. Scannet: Richly-Annotated 3d Reconstructions of Indoor Scenes. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 21–26 July 2017.
84. Eitz, M.; Richter, R.; Boubekur, T.; Hildebrand, K.; Alexa, M. Sketch-Based Shape Retrieval. *ACM Trans. Graph.* **2012**, *31*, 31. [[CrossRef](#)]
85. Chang, X.A.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H. Shapenet: An Information-Rich 3d Model Repository. *arXiv* **2015**, arXiv:1512.03012.
86. Riemenschneider, H.; Bódis-Szomorú, A.; Weissenberg, J.; van Gool, L. Learning Where to Classify in Multi-View Semantic Segmentation. In Proceedings of the Computer Vision—ECCV 2014—13th European Conference, Zurich, Switzerland, 6–12 September 2014.
87. Zeng, A.; Song, S.; Nießner, M.; Fisher, M.; Xiao, J.; Funkhouser, T. 3dmatch: Learning Local Geometric Descriptors from Rgb-D Reconstructions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
88. Gaidon, A.; Wang, Q.; Cabon, Y.; Vig, E. Virtual Worlds as Proxy for Multi-Object Tracking Analysis. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
89. Geiger, A.; Lenz, P.; Urtasun, R. Are We Ready for Autonomous Driving? The Kitti Vision Benchmark Suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2012), Providence, RI, USA, 16–21 June 2012.
90. Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y. Spectral Networks and Locally Connected Networks on Graphs. In Proceedings of the 2nd International Conference on Learning Representations (ICLR 2014), Banff, AB, Canada, 14–16 April 2014.
91. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [[CrossRef](#)]
92. Sattler, T.; Tylecek, R.; Brox, T.; Pollefeys, M.; Fisher, R.B. 3d Reconstruction Meets Semantics—Reconstruction Challenge 2017. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), Boston, MA, USA, 7–12 June 2015.
93. Hackel, T.; Wegner, J.D.; Schindler, K. Fast semantic segmentation of 3d point clouds with strongly varying density. *ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2016**, *3*, 177–184. [[CrossRef](#)]
94. Serna, A.; Marcotegui, B.; Goulette, F.; Deschaud, J. Paris-Rue-Madame Database: A 3d Mobile Laser Scanner Dataset for Benchmarking Urban Detection, Segmentation and Classification Methods. In Proceedings of the 4th International Conference on Pattern Recognition Applications and Methods—ICPRAM 2015, Lisbon, Portugal, 10–12 January 2015.
95. Gehring, J.; Hebel, M.; Arens, M.; Stilla, U. An Approach to Extract Moving Objects from Mls Data Using a Volumetric Background Representation. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*; Copernicus GmbH: Gottingen, Germany, 2017.
96. Borgmann, B.; Hebel, M.; Arens, M.; Stilla, U. Detection Of Persons In Mls Point Clouds. *ISPRS Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2017**, *2*, 203–210. [[CrossRef](#)]

97. Johnson, A.; Hebert, M. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 433–449. [[CrossRef](#)]
98. Chen, D.; Tian, X.; Shen, Y.; Ouhyoung, M. On Visual Similarity Based 3d Model Retrieval. In *Computer Graphics Forum*; Blackwell: Oxford, UK, 2009.
99. Douillard, B.; Underwood, J.; Vlaskine, V.; Quadros, A.; Singh, S. A Pipeline for the Segmentation and Classification of 3d Point Clouds. In Proceedings of the 12th International Symposium on Experimental Robotics (ISER 2010), New Delhi and Agra, India, 18–21 December 2010.
100. Zaheer, M.; Kottur, S.; Ravanbakhsh, S.; Póczos, B.; Salakhutdinov, R.; Smola, A.J. Deep Sets. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
101. Rusu, B.R.; Blodow, N.; Beetz, M. Fast Point Feature Histograms (Fpfh) for 3d Registration. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009), Kobe, Japan, 12–17 May 2009.
102. Kohonen, T. The Self-Organizing Map. *Proc. IEEE* **1990**, *78*, 1464–1480. [[CrossRef](#)]
103. Shi, S.; Wang, X.; Li, H. Pointcnn: 3d Object Proposal Generation and Detection from Point Cloud. *arXiv* **2018**, arXiv:1812.04244.
104. Zhou, Y.; Tuzel, O. Voxelnet: End-to-End Learning for Point Cloud Based 3d Object Detection. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, 18–22 June 2018.
105. Sindagi, A.V.; Zhou, Y.; Tuzel, O. Mvx-Net: Multimodal Voxelnet for 3d Object Detection. *arXiv* **2019**, arXiv:1904.01649.
106. Zhou, J.; Lu, X.; Tan, X.; Shao, Z.; Ding, S.; Ma, L. Fvnet: 3d Front-View Proposal Generation for Real-Time Object Detection from Point Clouds. *arXiv* **2019**, arXiv:1903.10750.
107. Qi, C.R.; Liu, W.; Wu, C.; Su, H.; Guibas, L.J. Frustum Pointnets for 3d Object Detection from Rgb-D Data. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, 18–22 June 2018.
108. Qi, R.C.; Litany, O.; He, K.; Guibas, L.J. Deep Hough Voting for 3d Object Detection in Point Clouds. *arXiv* **2019**, arXiv:1904.09664.
109. Lu, X.; Liu, Y.; Li, K. Fast 3d Line Segment Detection from Unorganized Point Cloud. *arXiv* **2019**, arXiv:1901.02532.
110. Li, X.; Guivant, J.E.; Kwok, N.; Xu, Y. 3d Backbone Network for 3d Object Detection. *arXiv* **2019**, arXiv:1901.08373.
111. Razlaw, J.; Quenzel, J.; Behnke, S. Detection and Tracking of Small Objects in Sparse 3d Laser Range Data. *arXiv* **2019**, arXiv:1903.05889.
112. Simon, M.; Amende, K.; Kraus, A.; Honer, J.; Sämann, T.; Kaulbersch, H.; Milz, S.; Gross, H. Complexer-Yolo: Real-Time 3d Object Detection and Tracking on Semantic Point Clouds. *arXiv* **2019**, arXiv:1904.07537.
113. Ku, J.; Pon, A.D.; Waslander, S.L. Monocular 3d Object Detection Leveraging Accurate Proposals and Shape Reconstruction. *arXiv* **2019**, arXiv:1904.01690.
114. Koguciuk, D.; Chechlinski, L. 3d Object Recognition with Ensemble Learning—A Study of Point Cloud-Based Deep Learning Models. *arXiv* **2019**, arXiv:1904.08159.
115. Cheraghian, A.; Rahman, S.; Petersson, L. Zero-Shot Learning of 3d Point Cloud Objects. *arXiv* **2019**, arXiv:1902.10272.
116. Voulodimos, A.; Doulamis, N.; Doulamis, A.; Protopapadakis, E. Deep learning for computer vision: A brief review. *Comput. Intel. Neurosci.* **2018**, *2018*, 13.
117. Bengio, Y.; Courville, A.C.; Vincent, P. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR* **2012**, *2012*, 1.
118. Le, E.T.; Kokkinos, I.; Mitra, N.J. Going Deeper with Point Networks. *Comput. Vis. Pattern Recognit.* **2019**.
119. Lin, T.; Dollár, P.; Girshick, R.B.; He, K.; Hariharan, B.; Belongie, S.J. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 8 September 2017.

