IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 23, 2019, accepted May 14, 2019, date of publication May 24, 2019, date of current version June 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2918862

DPRNet: Deep 3D Point Based Residual Network for Semantic Segmentation and Classification of 3D Point Clouds

SAIRA ARSHAD[1], MUHAMMAD SHAHZAD[1,2], (Member, IEEE), QAISER RIAZ[1], AND MUHAMMAD MOAZAM FRAZ[1,2,3,4], (Senior Member, IEEE)

[1]School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[2]Deep Learning Laboratory, National Center of Artificial Intelligence, Islamabad 44000, Pakistan
[3]Department of Computer Science, University of Warwick, Coventry CV47AL, U.K.
[4]The Alan Turing Institute, British Library, London NW1 2DB, U.K.

Corresponding author: Muhammad Shahzad (muhammad.shehzad@seecs.edu.pk)

This work was financially supported by NUST, Islamabad. Moreover, the authors gratefully acknowledge the donation of Titan X GPU by Nvidia.

ABSTRACT Point clouds are an important type of geometric data obtained from a variety of 3D sensors. They do not have an explicit neighborhood structure and therefore several researchers often perform a voxelization step to obtain structured 3D neighborhood. This, however, comes with certain disadvantages, e.g., it makes the data unnecessarily voluminous, enforces additional computation effort and can potentially introduce quantization errors that may not only hinder in extracting implicit 3D shape information but also in capturing the essential data invariances for the required segmentation and recognition task. In this context, this paper addresses the highly challenging problem of semantic segmentation and 3D object recognition using raw unstructured 3D point cloud data. Specifically, the deep network architecture has been proposed which consists of a cascaded combination of 3D point-based residual networks for simultaneous semantic scene segmentation and object classification. It exploits the 3D point-based convolutions for representational learning from raw unstructured 3D point cloud data. The proposed architecture has a simple design, easier implementation, and the performance which is better than the existing state-of-the architectures particularly for semantic scene segmentation over three public datasets. The implementation and evaluation are made public here https://github.com/saira05/DPRNet.

INDEX TERMS Object recognition, 3D point cloud, deep residual learning, 3D semantic segmentation.

I. INTRODUCTION

Semantic segmentation of point cloud refers to labeling each 3D point as belonging to a particular predefined object category. It is particularly useful for 3D object detection that enables to determine precise object contours together with their label in 3D space. It has pivotal role in scene understanding which in turn has wide range of diverse applications in different fields including robotics (e.g., autonomous navigation, terrestrial mapping, housekeeping, old-age assistance, agriculture), augmented/virtual reality, remote sensing (e.g., urban modeling, vegetation monitoring, surveying), 3D medical imaging, and many others.

Before the era of deep learning, majority of earlier approaches that addressed the point cloud segmentation problem may be broadly grouped into clustering (e.g., *k*-means and meanshift) based methods, similarity based region growing algorithms [1], edge-based segmentation techniques [2], [3] model fitting based methods [4], [5] and approaches that model the segmentation as a graph partitioning problem where optimal solution is typically determined using energy minimization techniques. These approaches assign the label to each point based on some kind of coherence/similarity typically defined in terms of low-level cues e.g., euclidean distance, surface normals, scatter, entropy, eigen value analysis etc. The task of semantic segmentation is slightly different from such coherent segmentation in a way that it attempts to partition the whole point cloud into semantic groups that are further classified as belonging to one of the pre-defined object categories. To perform

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

such semantic segmentation, most techniques relied on extracting 3D hand-crafted features that are later fed to conventional machine learning classifiers, such as support vector machine (SVM) [6], [7] or random forests [8] etc., for prediction [9]–[16] and [17]. The classification stage is often coupled with post processing stage that exploit the high representational ability of graphical models where such techniques typically combine the classifier module with the conditional random field (CRF) to ensure smoothness constraint while performing prediction of each data point [14], [18]–[21]. Although this works well (particularly with random forest classifier [14], [15]), but often, the usually adopted modular wise training limits the flow of information between the classifier and the CRF module.

With the recent success of deep neural architectures on 2D images, a variety of deep learning based methods have been proposed to address the problem of semantic segmentation of 3D point clouds. Since the traditional convolution neural network (CNN) architectures are designed in a way that they only accept structured type input, therefore several researchers essentially focused on converting point clouds into 3D rasterized voxel grids prior to actual processing. For instance, Maturana and Scherer [22] developed VoxNet that is a 3D convolutional neural network (CNN) architecture based on a volumetric occupancy grid representation for point cloud segmentation. Tchapmi *et al.* [20] proposed an end-to-end framework which adopts voxel representation to obtain fine grained point-level segmentation of input point cloud by enforcing smoothness constraints using fully connected CRFs. Similarly, to ensure global consistency, Kim *et al.* [19] also employed CRF as a post processing step over a 3D volume capturing semantic and geometric relationships among the neighboring voxels to semantically label each input point by exploiting an effective prediction strategy. Although the voxelization step is helpful in obtaining a regular neighborhood structure which aids in applying conventional convolution technique to learn feature representations, but it also comes with certain disadvantages. E.g., it makes the data unnecessarily voluminous, enforces additional computation effort and can potentially introduce quantization errors that may not only hinder in extracting implicit 3D shape information but also in capturing the essential data invariances for the required segmentation task.

Directly processing on 3D point clouds can potentially overcome these aforementioned limitations. In this regard, few researchers have proposed solutions that directly does the processing on input 3D point clouds without the additional step of voxelization [23]–[28]and [29]. Among them, the most notable and pioneering architecture is PointNet, proposed by Qi *et al.* [25], which presents a unified architecture aimed at solving different applications including 3D object classification, part segmentation and its extension to semantic segmentation. It is highly efficient and robust but does not take into account the variations in the local structures which consequently limits its capability to classify and segment at fine-grained level. To overcome this, a recursive application of PointNet over a nested partitioning of the input point set has been proposed in [26] which exploits the metric space distances to allow learning local and contextual features at increasing scales.

Since the point cloud are unordered and highly irregular, therefore the direct application of kernel convolutions may lead to loss of shape information and variance to point ordering. To cope with it, Hua *et al.* [30] recently introduced a point-wise convolution operator that has the ability to learn point level features by applying it on every point of the input point cloud using a pointwise CNN to perform semantic segmentation and object recognition. The capability of using such point-wise convolutions using deeper networks have been studied and found limited [30] since the deeper architectures resulted in relatively degraded performance when compared to the base pointwise architecture.

To this end, in this paper we have proposed a deep 3D point based CNN architecture that semantically segment individual points belonging to a particular class by exploiting the idea of residual learning [31] in the pointwise CNN architectures [30]. The proposed architecture includes the skip connection that help to avoid the vanishing gradient problem while training the network from scratch. The use of residual learning has been a great success in the domain of 2D image classification problems as it allows to add more layers without sacrificing the network performance. Despite of this, up to our knowledge, the idea of residual learning in unstructured (raw) point cloud segmentation has not been explored. The proposed architecture essentially translates the idea of using the 2D deep residual network design towards 3D convolutional architecture able to consume unstructured point cloud as input. Following are the key contributions of the proposed approach:

- The proposed Deep Point based Residual Network (DPRNet) architecture comprises of a cascaded combination of 3D point based residual networks for simultaneous semantic scene segmentation and object classification.
- DPRNet uses raw unstructured 3D point cloud data for representational learning using 3D point based convolutions.
- DPRNet is evaluated on three public datasets with its performance being comparable with the existing state-of-the architectures for scene segmentation. The implementation and evaluation is made public here https://github.com/saira05/DPRNet.

The paper is organized as follows: Section II contains an overview of the related literature. Section III present in detail the proposed deep point based residual network. Section IV provides the qualitative and quantitative results together with the performance analysis and comparison with the existing state-of-the-art architectures. Finally, in Section V, the conclusions are drawn and possible future avenues are discussed.

II. RELATED WORK

A. TRADITIONAL METHODS

Before the wide adoption of deep learning based architectures, hand-crafted features have been traditionally used for discriminative representation. For instance, 3D data was typically transformed into their respective 2D counterparts and the feature representation was carried out using simple features such as histograms, bag-of-feature models or more discrimintatve harmonic [32] and light field descriptors [33] for representing 3D shapes.

Initially, the work of semantic labelling of point clouds mainly focused aerial laser scans to segment different objects for reconstruction purposes. The typical workflow follows the strategy of converting the raw point clouds into regular 2.5D rasterized grid over which conventional image processing algorithms for edge detection and texture analysis [34] together with maximum likelihood classification [35], or bottom-up iterative classification rules [36], [37] have been employed to semantically segment 3D point cloud. In urban environments, a plethora of approaches have been presented that aim to perform point cloud segmentation. For instance, Hackel *et al.* [38] proposed semantic segmentation of 3D point clouds by using Random forest as a classifier after extracting 3D features based on eigenvalues/eigenvectors analysis. Vosselman [39] combined traditional methods of segmentation (like region growing and connected components etc.) awith additional rule based post processing steps to perform meaningful segmentation of point clouds. Qiu and Neumann [10] proposed using pre-segmented exemplar shapes/models for individual categories and used model fitting to segment each point. Instead of adopting a model fitting approach, Pang and Neumann [9] combined machine learning methods with 3D local features and performed object recognition on cluttered point cloud scenes. Huang and You [13] combined learning based classification with local descriptors for object detection in the industrial point cloud scenes.

Although these aforementioned hand-crafted 3D features based approaches work fairly well but often fail to generalize in conditions when the particular hand crafted feature does not capture the specific underlying dynamics of scene. In contrast, such dynamics are better incorporated in features learned via recent state-of-the-art deep neural network (DNN) architectures presented in the next subsections.

B. DEEP LEARNING METHODS

1) 3D OBJECT RECOGNITION

Among other DNN architecture, the Convolutional neural network (CNNs) have gained a lot of attention due to their ability to progressively learn hierarchical discriminative features from the input images. The CNN learned features have outperformed the conventional hand-crafted features for various tasks including classification and object detection. The pioneering work in the use of CNNs has been presented by Alexnet [40] which proposed a rather shallow 5 convolutional layers network architecture. Later, it has been demonstrated that more deep architectures allow to encapsulate high level feature extraction which in turn is much better in distinctive representation [41], [42]. Training of deep architectures, however, was challenging owing to the sol-called vanishing gradient problem. To cope with this, [31] proposed the concept of residual learning which allows to add short cut connections in the network which allows effective training of deep architectures with out losing performance. ResNet architectures based on the residual concept won the ILSVRC 2015 with a remarkable low error rate of 3.6% (surpassing humans) setting up a new record in image recognition and localization via a single network architecture.

Owing to widespread success of CNNs, many approaches have been proposed which translte the idea of 2D CNN to 3D. In this regard, Voxnet [22] used a voxel-based representation of point clouds to perform real-time object recognition. Similar use of 3D CNN on voxelized shapes have been presented in [43] and [44]. These networks adopted the volumetric representation which has a disadvantage of consuming more memory and being computationally expensive. To overcome this, Multi-view CNNs [43], [45] have been proposed to render 3D point cloud into 2D images and subsequently employ 2D convolution network for classification purpose. Although this approach achieved good performance for object recognition and shape retrieval purposes but their extension to scene understanding was rather limited [46].

To cope with an additional overhead of voxelization, recently few researchers have aimed to segment raw unstructured point cloud. Among them, PointNet [25] is a pioneering architecture which directly consumed unstructured 3D point clouds. PointNet architecture is robust and has the ability to learn an order-invariance function. The drawback of PointNet architecture is its inability to capture the relationship among the neighboring points. PointNet++ [26], an extension to PointNet, is a hierarchical type neural network which is able to better capture such neighboring relationship to learn more complex point-features. Klokov and Lempitsky [29] adapted the 3D indexing structure (i.e., kd-tree) to build computational graphs and proposed a deep architecture that mimics hierarchical convolution networks with learnable shared parameters. The deep kd-tree network is efficient in terms of memory and computation time as compared to uniform 3D voxel grids based architectures. PointCNN [47] attempted to present a generalization of typical CNNs to learn point features (using feed forward multilayer perceptrons) that are later passed to a hierarchical network, where $\mathcal{X}$-conv is applied on transformed features prior to element-wise convolution operation. Similarly, to improve learning of 3D features, Hua *et al.* [30] developed a 4-layer CNN architecture for scene understanding in which the point features are learned using 3D kernel based pointwise convolution operations.

2) 3D SEMANTIC SEGMENTATION

In the context of 3D, SemanticFusion [48] explored the idea of transferring the semantic segmentation predictions from 2D to 3D domain. SSCNet [49] proposed an architecture in which CNNs are used for 3D volumetric representation to assign a label to each voxel in the scene. This method could be easily adapted for real-time scene reconstruction based techniques, e.g., Voxel hashing [50] and KineticFusion [51] which are based on volumetric representations. SEGCloud [20] also covert point clouds into 3D voxel grids and applied 3D CNN for voxel-based prediction. The architecture incorporated the interpolation layer for transferring the individual voxel label to points and used additional CRF layers for post-processing. The combined training of 3D CNN along with interpolation and CRF layers enable the network to assign a label to each point. PointNet [25] and PointNet++ [26] proposed separate architectures for semantic segmentation by making modifications in their classification network. Wang *et al.* [27] proposed a similarity group proposal network for instance segmentation by utilizing features of PointNet/PointNet++. The group proposals are generated by computing the euclidean distance distance between pair of point-features. Engelmann *et al.* [28] also explored the spatial context for improving semantic segmentation of 3D point clouds using PointNet and proposed two architectures based on multi-scale blocks with consolidation units and grid-blocks with recurrent consolidation units respectively. Qi *et al.* [52] proposed a 3D graph neural network for semantic segmentation using RGBD images. Hua *et al.* [30] used 3D kernel based convolutions to learn features and used them with color attributes to semantically segment individual points. Here, in this paper, we also propose an idea similar to [30] but instead developed a more deep architecture based on the concept of residual learning to which provides relatively better accuracies particularly in the indoor scene segmentation scenario.

III. METHODOLOGY

A. SPATIAL SORTING (PREPROCESSING)

In contrast to 3D convolution over voxel space, point ordering (or structuring) is essential while working with unstructured 3D point cloud data. It is necessary because raw point cloud of a scene containing multiple objects contains no structural information that yields the neighbourhood relationship, i.e., points stored in memory next to each other may belong to completely different and spatially distinct objects. Typically, such point ordering/structuring may be partially achieved by employing hierarchical data structures such as kd-tree [29], octree [53] etc., that build efficient computational graphs to explicitly capture the neighbourhood relationships. Without such explicit relationship (which exists in 2-D images or voxel representation), directly feeding the unstructured point clouds to the deep network would require the network to be invariant to N! input permutations [25]. Such a constraint has been resolved in the literature either by defining a symmetric function to accumulate information retrieved by each point [25] or sorting the input point cloud based some criterion [30], e.g., using cartesian x, y, z coordinates or Morton curve [54] where both shows the comparable results. In the proposed approach, we employed the spatial sorting strategy, i.e., using the XYZ order as a pre-processing step in our deep network to tackle the point ordering issue. The use of spatial sorting is practically more suitable as it makes the learned classification/segmentation network more generic by eliminating the need to specifically learn a model (e.g., symmetric function [25]) to make the network invariant to point order.

B. POINT-BASED DILATED 3D CONVOLUTION

An essential component contributing to the success of CNNs is the primitive convolution operator that has the ability to exploit the correlation characteristics in the local neighborhood of the data (e.g., 2D images) expressed in dense cells (or grids/pixels). Presumably, such an effective operation is not efficient in feature learning in unstructured point clouds owing to the absence of an explicit structured neighborhood. For this reason, researchers applying the deep CNNs based approaches to segment point clouds often perform voxelization step to avoid the irregularity. Although this allows to simply translate the 2D convolution operator to its 3D counterpart but have a resulting trade-off in terms of consequent quantization effects.

Directly being able to apply convolutions on unstructured point clouds may overcome such effects. To this end, we employ a point based convolutional operator (introduced in [30]) which can convolve over each point and extract the features from each 3D point. The concept of point based convolution follows the concept similar to that of convolution in volumes in the 2D domain. Figure 2 shows the graphical illustration of how point based convolution kernel works on unstructured 3D points. The convolution kernel is placed on each 3D point and the neighborhood is binned (using nearest neighbors) into a certain number of 3D grid cells where the number of grid cells depend upon the size of the convolution kernel $u \times v \times q$, e.g., 27 grid cells are obtained for a $3 \times 3 \times 3$ kernel size. Points within each grid cell share same weights and the convolution results are obtained as follows [30]:

$$p_k^{l+1} = \sum_{i=1}^{u\times v\times q} \left[\frac{1}{|\xi_k\,(i)|} \sum_{j=1}^{|\xi_k(i)|} p_j^l \right] \cdot w_i \tag{1}$$

where w_i represents the kernel weight at ith grid cell while $p_j^l \in \xi_k\,(i)$ denotes the value of any jth point lying within ith grid cell $\xi_k\,(i)$ in the previous layer l. For object recognition, the values of points in the first layer (i.e., p_j^1) of any jth point is initialized with its coordinate values $\left(x_j, y_j, z_j\right)$ only while for semantic segmentation it also includes the other attributes (i.e., RGB values etc.).

The aforementioned point based convolution can be extended to $\grave{a}$-trous (or dilated) convolution in which an additional stride parameter can also be used that defines the gap between the grid cells and filter (or kernel) upsampling. I.e., the stride factor s (equivalent to normal convolution for $s = 1$) introduces $s - 1$ gaps between the kernel cells.

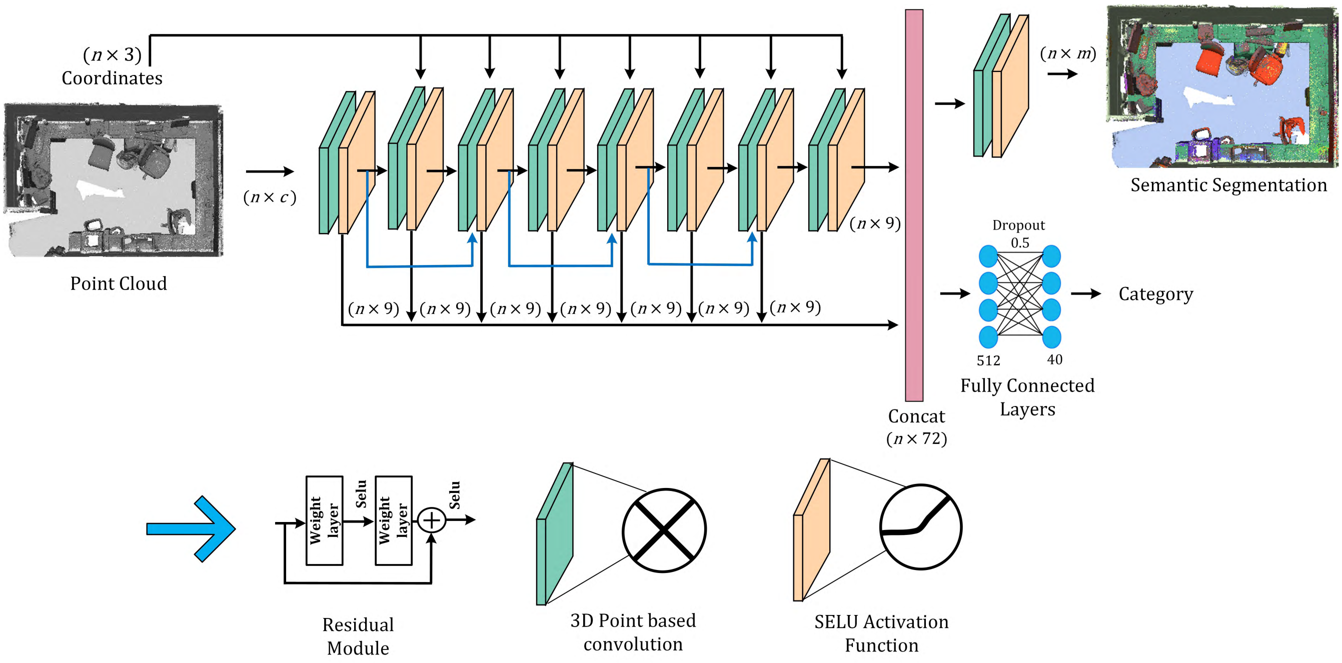


FIGURE 1. Proposed 8-layer Deep Point based Residual Network (DPRNet) architecture for both semantic segmentation and object recognition. Here *n* represents the number of points while *m* denotes the number of output classes for the semantic scene segmentation. The blue arrows show shortcut connections for residual learning.

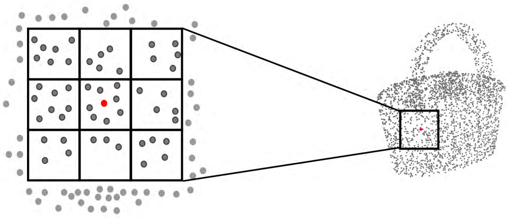


FIGURE 2. Illustration of the point based convolution kernel. The kernel is placed on each point. The red point shows the point of interest whose convolution value is being computed. The nearest neighbor points are binned into 3 by 3 grid cells and contribute in the computation of convolution.

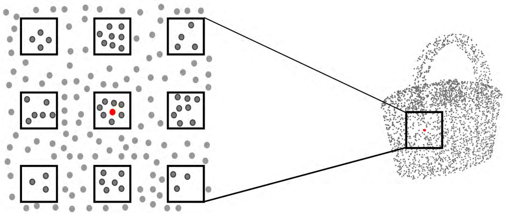


FIGURE 3. Illustration of dilated convolution using the kernel with stride of 2. Points in the kernel gaps do not contribute in the convolution computation.

Figure 3 shows how the $\grave{a}$-trous convolution extend kernel size with the gap being equal to the size of the grid cell (i.e., stride of 2). The benefit of using $\grave{a}$-trous convolution is two-fold: Firstly, it improves the network speed by avoiding processing of too many points. Secondly, by changing the stride, the perceptive field of the kernel can also be extended. In this way, the filter has a trade-off between two perceptive fields, i.e., small perceptive field allows accurate localization while the large perceptive field helps in context assimilation.

C. PROPOSED NETWORK ARCHITECTURE – DEEP CONVOLUTIONS WITH SHORTCUT CONNECTIONS

Figure 1 provides an overview of the proposed Deep Point based Residual Network (DPRNet). It takes as input the spatially sorted 3-D unstructured point cloud and assigns a label to each point to semantically segment the whole input scene and output the label of each input point, i.e., each point is assigned a label category as belonging to a particular object class. Additionally, the proposed architecture also recognizes the type of object solely based on the geometric 3D point representations. The workflow begins by applying a series of pointwise 3D stacked convolutions to obtain hierarchical point feature maps that are concatenated and subsequently fed to a fully connected layer for object recognition and to an additional pointwise convolution layer to achieve semantic segmentation. To allow stacking the convolution layers, two design choices are plausible, i.e., gradually increase either the kernel radius for each layer or the stride incase of $\grave{a}$-trous convolution in each layer. The latter choice is adopted since increasing the kernel radius takes much more time for the network to converge. Thus, the point features are thus learned using a fixed kernel radius for all layers while gradually increasing stride in each layer for all experiments.

1) RESIDUAL LEARNING

To allow deeper training by avoiding the so-called vanishing gradient problem, we employed the concept of residual

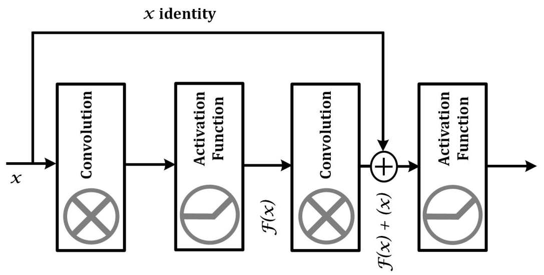


FIGURE 4. **Basic building block used for residual learning. The same design is followed for whole deep convolution neural networks.**

learning by adding shortcut (or skip) connections [28] that apply identity mappings whose results later get added to the output of the two or more bypassed (or in-parallel) stacked convolution layers, i.e., instead of learning a function, the network learns the residuals to ensure an identity mapping in every layer. More specifically, let us suppose that $\mathcal{H}(x)$ represents an underlying mapping function that should be fit after a few stacked layers with the input x to first layers. If one can make this hypothesis that complicated functions can be asymptotically approximated by multiple nonlinear stacked layers, then same is also the case for residual functions, i.e., $\mathcal{H}(x)$ $-x$ (assuming dimension of input and output is same). Instead of expecting approximation of $\mathcal{H}(x)$ by stacked layers, let these stacked layers approximate the residual function $\mathcal{F}(x) := \mathcal{H}(x) + x$ and then original form of residual function becomes $\mathcal{F}(x) + x$ [31]. It is easier to approximate identity mapping with residual functions as compared to approximate with multiple nonlinear stacked layers [31], [55], [56]. The main thing is that identity mapping must be optimal, but it is difficult in real cases, so if we design a function that is optimal and closer to identity mapping (than to a zero mapping), then it is easier for a solver to find the perturbations with reference to an identity mapping. He *et al.* [31] showed how such identity mapping can be performed by a shortcut connection in a simple way. This is the essence of residual learning where the basic building block is the residual module (Figure 4) which can be represented as:

$$y = \mathcal{F}(x, \{W_i\}) + x \tag{2}$$

Shortcut connections can be constructed by stacking multiple convolution layers. Element-wise addition is performed between two feature maps, channel by channel. They does not require an extra parameter or extra training time and allow lower layer to jump directly toward higher layer by skipping intermediate layers. Addition of such short cut connections thus allow adding more point based convolution layers in the proposed architecture consequently yielding better performance.

2) IMPLEMENTATION & NETWORK TRAINING

The whole DPRN architecture is trained from scratch using stochastic gradient descent (SGD) algorithm with the following parameters: learning rate = 0.001, momentum = 0.9, decay rate 0.96 and batch size of 32. The architecture does not use batch normalization. Instead, self-normalizing activation function [57] – Scaled Exponential Linear Units (SELU) – has been employed which induce self-normalizing properties by enabling convergence of activation neurons to zero mean and unit variance in automatic manner. Moreover, they have the ability to learn faster and better in comparison to other activation functions e.g., ReLU [57]. As mentioned earlier, we gradually increase the stride parameter for dilated convolution in each layer and simply added the output of layer1 to the next by stacking two layers in 8-layers model i.e., (layer3) and so on. In this way, the network turns into its correspondence residual version. Although, the gradual increase in the stride parameter on each layer enlarges the perceptive field of the kernel but does not alter the output dimension of layers. Moreover, there is no pooling layer in the architecture which downsample the dimension of the feature maps on each layer. As a consequent, there is no difference in the input and output dimensions of layers. In case a pooling layer is to be incorporated, one can use either of the two options to cope with the difference in input and output dimensions: 1) Use zero-padding for increasing/decreasing dimensions (does not require any extra parameter); or 2) Use a linear projection W_s (performed by 1 × 1 convolution) for matching dimensions in the residual formulation as [31]:

$$y = \mathcal{F}(x, \{W_i\}) + W_s x \tag{3}$$

The training and implementation of the proposed deep point based residual CNN using above configurations is simple and outperforms in terms of accuracy many existing semantic segmentation networks. However, for thorough evaluations reported in the following section IV, we have also analyzed the performance of our point based residual network by training with the addition of layers before and after SELU activation function and by using the pre-activation residual module in which activation layer precedes the convolutional layer. In this way the network allows the information to flow unimpeded throughout the entire network [58]. Figure 5 shows the testing designs for all these various configurations of the activation function.

For the object recognition network, we also followed the concatenation of output from all layers and then passed the concatenated features through two fully connected layers and used the dropout value of 0.5 between the last two fully connected layers. Finally, for scene segmentation network, we added one point based convolutional layer to obtain the point wise semantic labelling.

IV. EXPERIMENTAL RESULTS & EVALUATION

A. DATASETS

We have evaluated the proposed DPRNet network using three benchmark datasets including ModelNet40 [44], S3DIS [59] and SceneNN [60]. The first dataset is the standard benchmark for object recognition while the latter two datasets pertains to the problem of semantic segmentation.

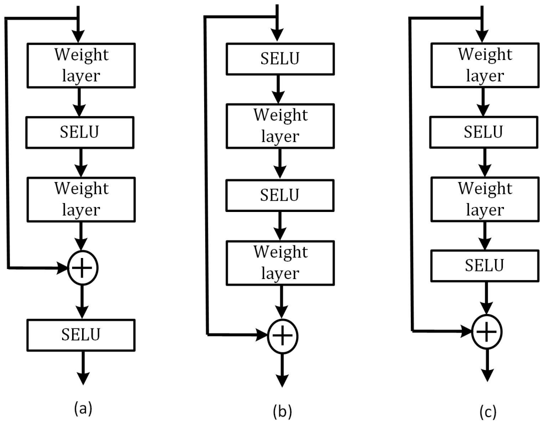


FIGURE 5. Figure shows the design for various usage of activation functions: (a) Shows the design where the addition of layers operation is performed before the SELU activation function; (b) shows the design of pre-activation (i.e., SELU as pre-activation) residual module; (c) shows the design of shortcut connection in which the addition of layers operation is performed after the SELU activation function.

ModelNet40 dataset comprises of 3D CAD models of 40 object categories. The total number of CAD models for these objects are 12311 CAD models which are split into 9,843 for training and 2,468 for testing. For object recognition evaluation task, we follow the same experimental settings as used in PointNet [25] in which 2048 points are uniformly sampled from mesh and normalize to the unit sphere. The input to the classification network is 2048 points with (x, y, z) coordinates.

S3DIS dataset consists of 6 areas including 271 rooms captured by matterport scanner. Each point is annotated with one semantic label from a pool of 13 categories. Each point has 9 attributes: XYZ coordinates, RGB value and normalized coordinates with reference to the room place it belongs to. For scene segmentation, each scene is divided into 1 × 1 square-meter blocks and sample to 4096 points. Each block is fed into the network for training. Prediction of the entire scene is measure by gathering the prediction of all blocks.

SceneNN dataset is comprised of complex indoor scenes with relatively higher clutter compared to S3DIS and is therefore quite challenging dataset in terms of semantic segmentation. It is essentially an RGB-D scene dataset in which all scenes have been reconstructed into triangular meshes. The annotations are available for every vertex of the reconstructed mesh along with the per-pixel annotation. For semantic segmentation training and evaluation, we have used the same experimental settings as used in [30] where 76 scenes are annotated from SceneNN dataset including 40 categories defined by NYUv2 [61] and are further divided into train and test set. For training, we used 56 scenes and for testing, we used 20 scenes as done in [30]. Again, scenes divided into 2 × 2 square-meter blocks and each block contained 4069 points. Prediction of the entire scene is measure by gathering the prediction of all blocks.

B. SEMANTIC SEGMENTATION

For scene segmentation task, we have evaluated our DPR-Net 8-layers network (with single shortcut connections and SELU after addition). Table 1 shows the scene segmentation results obtained on S3DIS dataset. We compared accuracy of the proposed DPRNet 8-layers segmentation network with PointNet [25], SGPN [27], pointwise CNN [30] and the segmentation networks proposed by Engelman et al. [28]. Engelman et al. [28] have proposed two architectures MS + CU (multi-scale blocks with consolidation units) and G + RCU (grid-blocks with recurrent consolidation units) and explored the spatial context by using PointNet [25] as the feature learner. SGPN [27] also used PointNet [25] as feature learner and proposed instance segmentation network which also outputs the semantic segmentation score. As can be seen, the proposed DRPN 8-layers segmentation network and attained 83.8% accuracy on S3DIS dataset outperforming all these state-of-the-art architectures.

TABLE 1. Comparison of semantic segmentation accuracy on S3DIS dataset.

Network	Overall Accuracy	Accuracy (per-class)
PointNet [25]	78.7	—
MS + CU [28]	79.2	—
G + RCU [28]	81.1	—
SGPN [27]	80.8	—
Pointwise CNN [30]	81.5	56.5
DPRNet (8-layers)	**83.8**	**59.5**

Figure 6 provides visualization of resultant scenes after semantic segmentation on S3DIS dataset [59]. As can be seen that the results are much better in case of non-overlapping structures e.g., walls, floor etc. For overlapping regions, we see some prediction inconsistencies which could be addressed using an additional post processing smoothing step (e.g., inclusion of conditional random field).

Table 2 presents the per-class accuracy results on S3DIS dataset. In terms of quantitative analysis, we can see that the proposed DPRNet per-class accuracy outperforms other state-of-the-art architectures including PointNet [25] and pointwise CNN [30]in many common classes.

TABLE 2. Comparison of semantic segmentation per-class accuracy on S3DIS dataset [59].

Object Classes	**Networks**		
	PointNet [25]	Pointwise CNN [30]	**DPRNet (8-layers)**
ceiling	**98.3**	97.4	97.8
floor	98.8	99.1	**99.4**
wall	83.3	89.1	**89.5**
door	**84.6**	62.9	71.7
window	**63.4**	61.2	54.9
clutter	69.0	65.2	**70.8**
table	70.3	73.7	**81.7**
chair	66.0	68.4	**71.1**
beam	56.7	58.8	**66.9**
bookcase	—	57.5	**59.5**

To further evaluate the performance of the proposed DPRNet 8-layers segmentation network, we obtained the

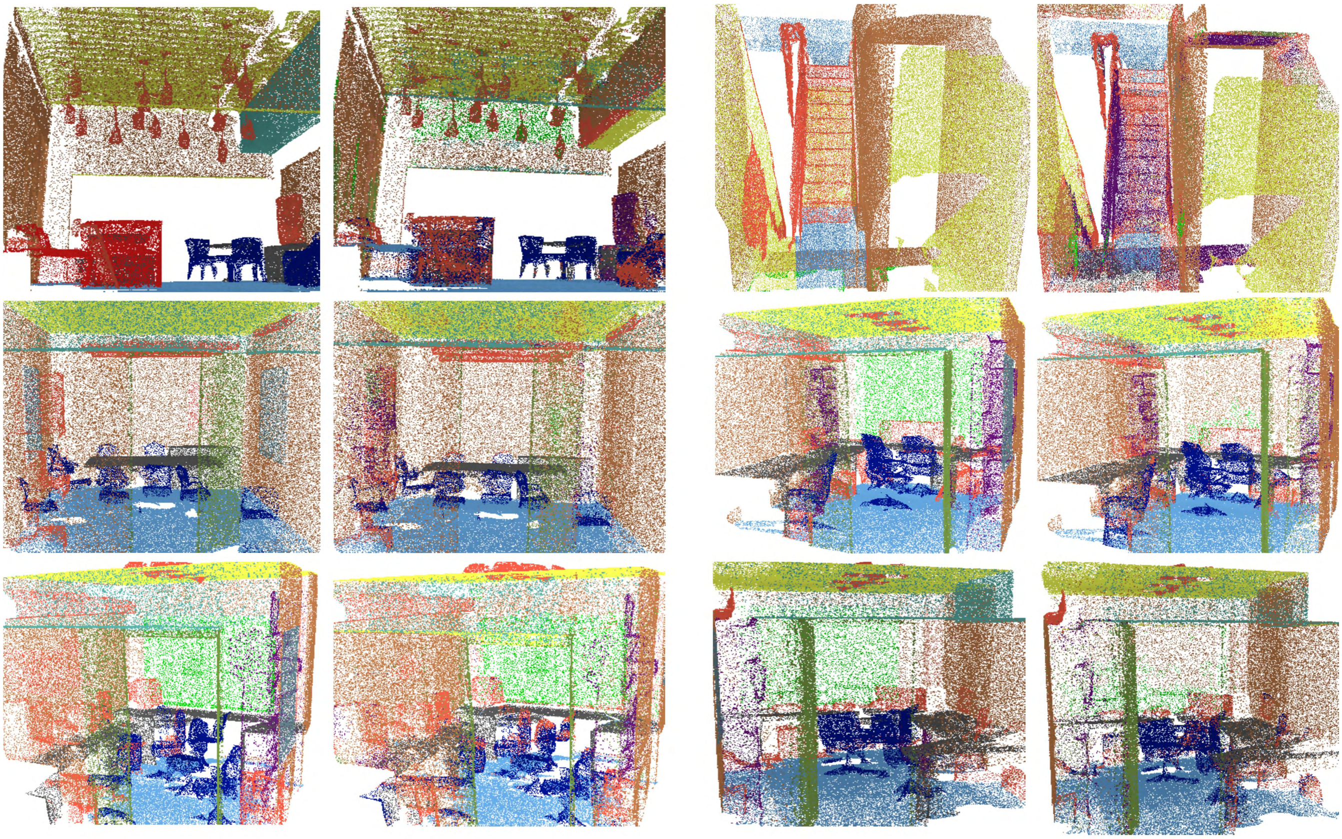


FIGURE 6. Visualization of semantic scene segmentation results obtained using S3DIS dataset. In this figure (a) shows ground truth segmentation and (b) shows DPRNet segmentation.

TABLE 3. Comparison of semantic segmentation per-class accuracy on sceneNN dataset [60].

Network	wall	floor	cabinet	bed	chair
Voxnet [22]	82.8	74.3	—	—	3.1
SemanticFusion [48]	72.8	94.4	—	46.3	—
PointNet [25]	**89.7**	89.1	09.0	45.7	59.6
Pointwise CNN [30]	86.8	86.4	21.4	**51.3**	63.9
DPRNet (8-layers)	87.4	**94.7**	**22.8**	38.0	**70.0**
Network	table	sofa	desk	tv	pillow
Voxnet [22]	0.8	—	5.4	—	—
SemanticFusion [48]	**70.1**	—	28.1	—	—
PointNet [25]	23.5	16.7	31.0	11.4	06.7
Pointwise CNN [30]	41.2	**29.8**	36.2	17.8	17.5
DPRNet (8-layers)	42.8	10.0	**38.3**	**26.3**	**17.7**

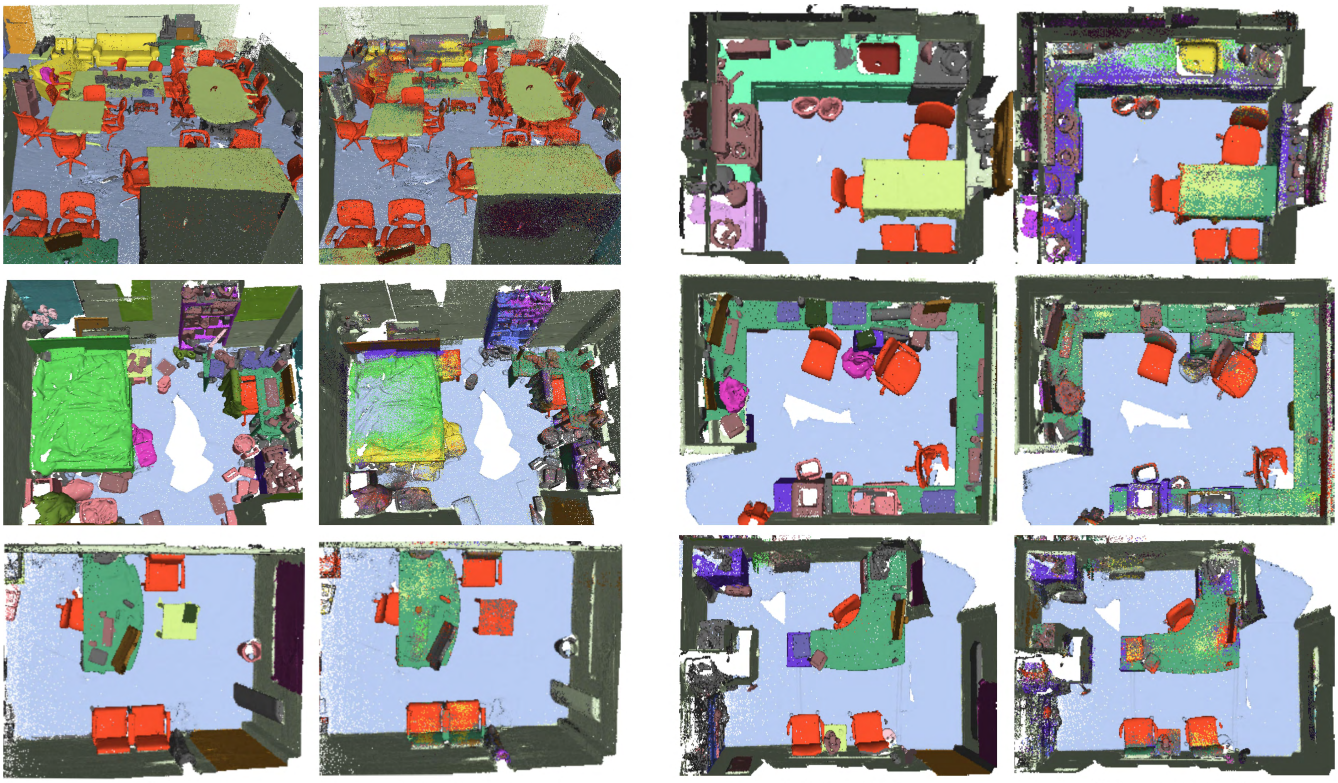
results on SceneNN containing complex indoor scenes of e.g., bedroom, kitchen, living room etc. Figure 7 provides visualization of resultant semantically segmented scenes of SceneNN dataset. While the results obtained over per-class accuracies and their comparison with reported Voxnet [22], SemanticFusion [48], PointNet [25] and pointwise [30] segmentation networks are presented in 3. It can be seen that the proposed DPRNet is competitive to other state-of-the-art networks in different object categories. In general we see a degraded performance of the VoxNet architecture owing to limited resolution. Moreover, the proposed DPRNet network has significantly improved accuracies in cases of categories including floor, chair, and bed. Furthermore, these results are obtained without performing and any post processing to smooth label predictions which some approaches (e.g., SemanticFusion) does to smooth the label predictions.

TABLE 4. Comparison of accuracy between ours DPRNet networks and the pointwise CNN deep plain networks. These experiments are done on ModelNet40 [44] dataset.

Network Architectures	Accuracy	Accuracy (per-class)
PointNet [25]	89.2	86.2
Pointwise CNN [30] 4-layers	86.1	81.4
Pointwise CNN [30] (increased to 8-layers)	82.1	—
Pointwise CNN [30] (increased to 16-layers)	82.6	—
DPRNet 8-layers	86.1	81.9
DPRNet 16-layers	85.4	82.1

C. OBJECT RECOGNITION

Table 4 and Table 5 shows a comparison of overall and per-class accuracies obtained over ModelNet40 dataset using the proposed DPRNet 8-layers, and 16-layers (with SELU after addition) with PointNet [25] and pointwise CNN [30] respectively. It can be seen that the performance in object recognition task is comparable to existing state-of-the-art



(a) Ground Truth (b) DPRNet Segmentation (a) Ground Truth (b) DPRNet Segmentation

FIGURE 7. **Visualization of semantic scene segmentation results obtained using SceneNN dataset. In this figure (a) shows ground truth segmentation and (b) shows DPRNet segmentation.**

TABLE 5. **Comparison of per-class classification accuracy on ModelNet40 [44] dataset.**

Network	airplane	bathtub	bed	bench	bookshelf	bottle	bowl	car
PointNet [25]	100.0	80.0	94.0	75.0	93.0	94.0	100.0	97.9
Pointwise [30]	100.0	82.0	93.0	68.4	91.8	93.9	95.0	95.6
DPRNet	100.0	76.0	95.0	80.0	85.0	95.0	95.0	91.0
Network	chair	cone	cup	curtain	desk	door	dresser	flowerpot
PointNet [25]	96.0	100.0	70.0	90.0	79.0	95.0	65.1	30.0
Pointwise [30]	96.0	80.0	60.0	80.0	76.7	75.0	67.4	10.0
DPRNet	97.0	90.0	70.0	80.0	86.0	85.0	60.5	25.0
Network	glassbox	guitar	keyboard	lamp	laptop	mental	monitor	night stand
PointNet [25]	94.0	100.0	100.0	90.0	100.0	96.0	95.0	82.6
Pointwise [30]	80.8	98.0	100.0	83.3	95.0	93.9	92.9	70.2
DPRNet	86.0	100.0	100.0	80.0	100.0	93.0	96.0	70.9
Network	person	piano	plant	radio	range hood	sink	sofa	stairs
PointNet [25]	85.0	88.8	73.0	70.0	91.0	80.0	96.0	85.0
Pointwise [30]	89.5	84.5	78.8	65.0	88.9	65.0	96.0	80.0
DPRNet	90.0	83.0	83.0	55.0	89.9	70.0	93.0	75.0
Network	stool	table	tent	toilet	tv stand	vase	wardrobe	xbox
PointNet [25]	90.0	88.0	95.0	99.0	87.0	78.8	60.0	70.0
Pointwise [30]	83.3	90.9	90.0	94.9	84.5	81.3	30.0	75.0
DPRNet	70.0	77.0	90.0	95.0	89.0	80.0	20.0	80.0

architectures. Despite of the fact that PointNet achieves higher accuracy, the proposed DPRNet is much simpler in design. We also compared the accuracies obtained on Pointwise CNN extended to 8 and 16 layers (Table 4) to see the effect of merely increasing the network layers without residual connections. As evident, the use of residual learning does increase the overall network accuracy.

D. ABLATION STUDY

We have performed all experiments for object recognition with batch size of 32, and initial learning rate of 0.001 with decay rate of 0.96 and momentum of 0.9. The batch size is empirically set and the effect of it to the resulting accuracy is depicted in Table 6 where network accuracies with batch sizes of 32, 64 and 128 are reported.

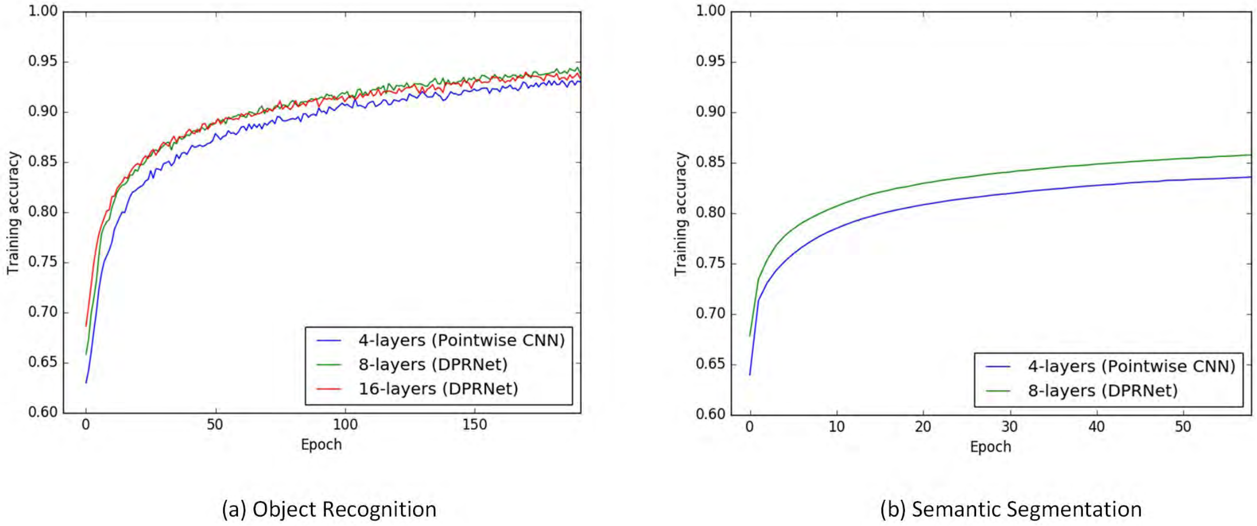


FIGURE 8. Training accuracy vs epoch graph for networks with different layers. The training accuracy of our proposed DPRNet 8-,16-layers compared with pointwise CNN [30] 4-layer network for object recognition and our proposed DPRNet 8-layers with pointwise CNN [30] 4-layer network for segmentation task, (a) shows comparison of training accuracy on object recongition task (b) shows comparison of training accuracy on scene semantic segmentation task.

TABLE 6. Difference in accuracy is depcited when batch size is increased.

Network	Accuracy	Batch size
DPRNet 8-layers	**86.1**	32
	85.2	64
	83.9	128

We have also evaluated the proposed DPRNet accuracy with the addition of layers output before and after self-normalizing activation function (SELU). We adopted this idea from [58] where different variants of residual learning are defined and different experiments are performed with various usage of activation functions. [58] experimented with 1000+ layers and showed that the error rate increases with the addition operation performed after ReLU activation function which is also the case in our experiments with SELU (although there was a minor difference in accuracies). Similarly, we have also evaluated our deep network accuracy with the pre-activation residual module in which activation function layer comes before the convolutional layer. In this way information is able to flow unimpeded throughout the entire network. Please refer to Figure 5 for visual illustration. Table 7 depicts the results achieved by applying SELU before and after the addition operation using pre-activation and post-activation residual module.

E. TRAINING TIMES & HARDWARE

Figure 8 shows the graphical representation of how training accuracy improves when more layers are added with shortcut connections. In comparison with the base 4-layer pointwise CNN [30] architecture, we see the improvement in training accuracies with the increase in network depth and addition of shortcut connections.

The proposed DPRNet 8-layers network takes around 4-5 days of training and DPRNet 16-layers network takes 7-8 days for scratch training. The previous estimates are obtained using batch size of 32 for all experiments with a single Tesla K80 GPU equipped desktop computer with following details: Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz and 16GB RAM. The total GPU memory consumed while training is around 10 GB. These training times can be easily improved using a multi GPU environment.

TABLE 7. Comparison of accuracy with various usage of activation function for DPRNet 8-layers network.

Network	Accuracy	Accuracy (per-class)
8-layers with SELU after addition	**86.1**	**81.9**
8-layers with SELU before addition	85.5	81.7
8-layers with pre-activation SELU	84.8	79.3
8-layers with post-activation SELU	**86.1**	**81.9**

V. CONCLUSION & OUTLOOK

In this paper, we have presented a deep point based convolution architecture that is able to perform semantic segmentation of individual points as well as recognize the object category using a group of points. The architecture is able to consume raw unstructured 3D point clouds as input and does not require an additional step of voxelization. Despite being a simple architecture, the proposed network design provides better accuracy competitive accuracies in comparision to the existing state-of-the-art architectures. The presented results are expected to further stimulate the use of deep architectures in processing raw point clouds obtained from variety of 3D sensors.

In relation to the proposed network architecture, following are few design parameters that are worth to be mentioned and further explored:

- *Point Based Convolution:* The point based convolution operator essentially takes the mean point values in each grid cell and assign them the same weight without taking into consideration their distance to the point of interest. In this regard, the grid cells may be weighted such that the nearest grid cells contribute more compared to the farther ones in convolution computation. A Gaussian weighting function assigning decaying weights to the grid cells may be good choice for initial future study.

- *Network Architecture Design:* The proposed architecture comprising of shortcut connections has the ability to semantically segment and recognize different objects using unstructured point clouds. Another worth to try concept is similar to the one proposed in DenseNet architecture [62] where each layer is directly connected to each other in the network making dense direction connections. DenseNets also avoid vanishing gradient problem and have the ability to reuse features while substantially reducing the number of parameters.
- *Neighborhood Selection (Grid Cell Size):* The size of the grid cells have been fixed in the proposed study. For optimal network design in terms of architecture with fewer hyperparameters, an adaptive grid size selection based on *k*-nearest neighbor approach, e.g., as proposed in [30], may be employed.
- *Hybrid features (Deep + hand crafted features):* Only deep point based convolution features have been used for both semantic segmentation and objection recognition tasks. Another possible future direction may be to incorporate conventional 3D point cloud features including surface normals, plane residuals, eigen based features, point density etc. with the deep features via concatenation before the fully connected layer. Such a combination may potentially yield good performance particularly for object recognition [63].

REFERENCES

[1] P. J. Besl and R. C. Jain, "Segmentation through variable-order surface fitting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 2, pp. 167–192, Mar. 1988.

[2] B. Bhanu, S. K. Lee, C. C. Ho, and T. C. Henderson, "Range data processing: Representation of surfaces by edges," in *Proc. IEEE 8th Int. Conf. Pattern Recognit.*, Oct. 1986, pp. 236–238.

[3] A. D. Sappa and M. Devy, "Fast range image segmentation by an edge detection strategy," in *Proc. IEEE 3rd Int. Conf. 3-D Digit. Imag. Modeling*, May/Jun. 2001, pp. 292–299.

[4] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," *Comput. Graph. Forum*, vol. 26, no. 2, pp. 214–226, 2007.

[5] F. Tarsha-Kurdi *et al.*, "Hough-transform and extended RANSAC algorithms for automatic detection of 3D building roof planes from lidar data," in *Proc. ISPRS Workshop Laser Scanning*, vol. 36, 2007, pp. 407–412.

[6] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. ACM 5th Annu. Workshop Comput. Learn. Theory*, 1992, pp. 144–152.

[7] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[8] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[9] G. Pang and U. Neumann, "Training-based object recognition in cluttered 3D point clouds," in *Proc. IEEE Int. Conf. 3D Vis. (3DV)*, Jun./Jul. 2013, pp. 87–94.

[10] R. Qiu and U. Neumann, "Exemplar-based 3D shape segmentation in point clouds," in *Proc. IEEE 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 203–211.

[11] A. Serna and B. Marcotegui, "Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning," *ISPRS J. Photogram. Remote Sens.*, vol. 93, pp. 243–255, Jul. 2014.

[12] A. K. Aijazi, A. Serna, B. Marcotegui, P. Checchin, and L. Trassoudaine, "Segmentation and classification of 3D urban point clouds: Comparison and combination of two approaches," in *Field and Service Robotics*. Cham, Switzerland: Springer, 2016, pp. 201–216.

[13] J. Huang and S. You, "Detecting objects in scene point cloud: A combinational approach," in *Proc. Int. Conf. 3D Vis.-3DV*, Jun./Jul. 2013, pp. 175–182.

[14] R. Zhang, S. A. Candra, K. Vetter, and A. Zakhor, "Sensor fusion for semantic segmentation of urban scenes," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1850–1857.

[15] D. Wolf, J. Prankl, and M. Vincze, "Enhancing semantic segmentation for robotics: The power of 3-D entangled forests," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 49–56, Jan. 2016.

[16] J. Zhang, X. Lin, and X. Ning, "SVM-based classification of segmented airborne LiDAR point clouds in urban areas," *Remote Sens.*, vol. 5, no. 8, pp. 3749–3775, Jul. 2013.

[17] R. Qiu and U. Neumann, "IPDC: Iterative part-based dense correspondence between point clouds," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–9.

[18] D. Wolf, J. Prankl, and M. Vincze, "Fast semantic segmentation of 3D point clouds using a dense crf with learned parameters," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 4867–4873.

[19] B.-S. Kim, P. Kohli, and S. Savarese, "3D scene understanding by Voxel-CRF," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1425–1432.

[20] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, "SEGCloud: Semantic segmentation of 3D point clouds," in *Proc. IEEE Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 537–547.

[21] B. Wu, A. Wan, X. Yue, and K. Keutzer, "SqueezeSeg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1887–1893.

[22] 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./Oct. 2015, pp. 922–928.

[23] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 3, Jul. 2017, pp. 3577–3586.

[24] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," in *Proc. CVPR*, Jul. 2017, pp. 3693–3702.

[25] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, vol. 1, no. 2, p. 4.

[26] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," 2017 *arXiv:1706.02413*. [Online]. Available: https://arxiv.org/abs/1706.02413

[27] W. Wang, R. Yu, Q. Huang, and U. Neumann, "SGPN: Similarity group proposal network for 3D point cloud instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2569–2578. [Online]. Available: http://arxiv.org/abs/1711.08588

[28] F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe, "Exploring spatial context for 3D semantic segmentation of point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2017, pp. 716–724.

[29] 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.

[30] B.-S. Hua, M.-K. Tran, and S.-K. Yeung, "Pointwise convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 984–993. [Online]. Available: https://arxiv.org/abs/1712.05245

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

[32] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Proc. Eurograph. Symp. Geometry Process.*, vol. 6, 2003, pp. 156–164.

[33] 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.

[34] C. Hug and A. Wehr, "Detecting and identifying topographic objects in imaging laser altimeter data," *Int. Arch. Photogram. Remote Sens.*, vol. 32, no. 3, pp. 19–26, 1997.

[35] H.-G. Maas, "The potential of height texture measures for the segmentation of airborne laserscanner data," in *Proc. 4th Int. Airborne Remote Sens. Conf. Exhib., 21st Can. Symp. Remote Sens.*, vol. 1, 1999, pp. 154–161.

[36] F. Rottensteiner and C. Briese, "A new method for building extraction in urban areas from high-resolution LIDAR data," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. 34, no. 3/A, pp. 295–301, 2002.

[37] N. Haala, C. Brenner, and K.-H. Anders, "3D urban GIS from laser altimeter and 2D map data," *Int. Arch. Photogram. Remote Sens.*, vol. 32, pp. 339–346, Jul. 1998.

[38] T. Hackel, J. D. Wegner, and K. Schindler, "Fast semantic segmentation of 3D point clouds with strongly varying density," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 3 pp. 177–184, Jul. 2016.

[39] G. Vosselman, "Point cloud segmentation for urban scene classification," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. 1, pp. 257–262, Nov. 2013.

[40] 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.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[43] 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.*, Jun. 2016, pp. 5648–5656.

[44] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1912–1920.

[45] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 945–953.

[46] M. Savva *et al.*, "SHREC'16 track large-scale 3D Shape retrieval from ShapeNet core55," in *Proc. Eurograph. Workshop 3D Object Retr.*, 2016, pp. 1–11.

[47] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," 2018, *arXiv:1801.07791*. [Online]. Available: https://arxiv.org/abs/1801.07791

[48] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3D semantic mapping with convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Jun. 2017, pp. 4628–4635.

[49] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 190–198.

[50] M. Nießner and M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3D reconstruction at scale using voxel hashing," *ACM Trans. Graph.*, vol. 32, no. 6, 2013, Art. no. 169.

[51] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. IEEE 10th Int. Symp. Mixed Augmented Reality (ISMAR)*, Oct. 2011, pp. 127–136.

[52] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun, "3D graph neural networks for rgbd semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2017, pp. 5199–5208.

[53] M. Bassier, M. Bonduel, B. Van Genechten, and M. Vergauwen, "Segmentation of large unstructured point clouds using octree-based region growing and conditional random fields," in *Proc. Int. Arch. Photogram., Remote Sens. Spatial Inf. Sci.*, vol. 42, 2017, pp. 25–30.

[54] G. M. Morton, "A computer oriented geodetic data base and a new technique in file sequencing," Int. Bus. Mach. Company, New York, NY, USA, 1966.

[55] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5353–5360.

[56] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," 2015, *arXiv:1505.00387*. [Online]. Available: https://arxiv.org/abs/1505.00387

[57] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 971–980. [Online]. Available: https://arxiv.org/abs/1706.02515

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 630–645.

[59] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1534–1543.

[60] B.-S. Hua, Q.-H. Pham, D. T. Nguyen, M.-K. Tran, L.-F. Yu, and S.-K. Yeung, "SceneNN: A scene meshes dataset with annotations," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, 2016, pp. 92–101.

[61] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2012, pp. 746–760.

[62] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *Proc. IEEE CVPR*, vol. 1, Jun. 2017, no. 2, p. 3.

[63] L. Jin, S. Gao, Z. Li, and J. Tang, "Hand-crafted features or machine learnt features? Together they improve RGB-D object recognition," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2014, pp. 311–319.

SAIRA ARSHAD received the B.S. degree in computer science from the University of the Punjab, Lahore, in 2014, and the M.S. degree in computer science from the School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan, in 2019. Her research interest includes semantic scene segmentation and classification using deep learning techniques applied over unstructured 3D point clouds.

MUHAMMAD SHAHZAD received the B.E. degree in electrical engineering from the National University of Sciences and Technology, Islamabad, Pakistan, in 2004, the M.S. degree in autonomous systems from the Bonn-Rhein-Sieg University of Applied Sciences, Sankt Augustin, Germany, in 2011, and the Ph.D. degree in radar remote sensing and image analysis from the Department of Signal Processing in Earth Observation (SiPEO), Technische Universität München (TUM), Munich, Germany, in 2016. He attended twice two weeks professional thermography training course at the Infrared Training Center, North Billerica, MA, USA, in 2005 and 2007, respectively. He was a Visiting Research Scholar with the Institute for Computer Graphics and Vision, Technical University of Graz, Austria. Since 2016, he has been a Senior Researcher with SiPEO, TUM, Germany, and an Assistant Professor with the School of Electrical Engineering and Computer Science, National University of Sciences and Technology. Moreover, he is also a Co-Principal Investigator of recently established Deep Learning Laboratory (DLL) under the umbrella of National Center of Artificial Intelligence (NCAI), Islamabad. His research interests include deep learning for processing unstructured/structured 3-D point clouds, optical RGBD data, and very high-resolution radar images.

QAISER RIAZ received the M.S. degree in autonomous systems from the Bonn-Rhein-Sieg-University of Applied Sciences, Sankt Augustin, Germany, in 2011, and the Ph.D. (Dr.rer.nat.) degree in computer science from the University of Bonn, Germany, in 2016. Since 2016, he has been an Assistant Professor with the Department of Computing, School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan. His research interests include motion capturing, human motion analysis and synthesis using low-cost sensors, character animation, and machine learning.

MUHAMMAD MOAZAM FRAZ received the Ph.D. degree from Kingston University London, U.K. He held a postdoctoral position from the University of Warwick, U.K. He was an Assistant Professor with NUST-SEECS, Islamabad, Pakistan. He is currently a Rutherford Fellow with The Alan Turing Institute, London; which is UK's National Center for Data Science and AI. Moreover, he is also affiliated with the recently established Deep Learning Laboratory (DLL) under the umbrella of National Center of Artificial Intelligence (NCAI), Islamabad. His research interests include retinal image analysis, automated visual surveillance, and visual recognition.