UNDERSTANDING 3D POINT CLOUD DEEP NEURAL NETWORKS BY VISUALIZATION TECHNIQUES

Yuwei Cao[1,]*, Mattia Previtali[1], Marco Scaioni[1]

[1] Department of Architecture, Built Environment and Construction Engineering, Politecnico di Milano
via Ponzio 31, 20133 Milano, Italy - emails: {yuwei.cao, mattia.previtali, marco.scaioni}@polimi.it

Commission II, WG II/6

KEY WORDS: Deep Learning, Reconstruction, Visualization, 3DCAM, 3D Point Cloud

ABSTRACT:

In the wake of the success of Deep Learning Networks (DLN) for image recognition, object detection, shape classification and semantic segmentation, this approach has proven to be both a major breakthrough and an excellent tool in point cloud classification. However, understanding how different types of DLN achieve still lacks. In several studies the output of segmentation/classification process is compared against benchmarks, but the network is treated as a "black-box" and intermediate steps are not deeply analysed. Specifically, here the following questions are discussed: (1) what exactly did DLN learn from a point cloud? (2) On the basis of what information do DLN make decisions? To conduct such a quantitative investigation of these DLN applied to point clouds, this paper investigates the visual interpretability for the decision-making process. Firstly, we introduce a reconstruction network able to reconstruct and visualise the learned features, in order to face with question (1). Then, we propose 3DCAM to indicate the discriminative point cloud regions used by these networks to identify that category, thus dealing with question (2). Through answering the above two questions, the paper would like to offer some initial solutions to better understand the application of DLN to point clouds.

1. INTRODUCTION

Inspired by human brains, Deep Learning (DL) is a subset of Machine Learning techniques that teaches computers to do what comes naturally to humans: learn from experience. Recent studies have shown that the features learned by various deep learning architectures (e.g., Convolutional Neural Network - CNN, Fully Connected Network - FCN, Recurrent Neural Network – RNN, etc.) are highly successful in image recognition, object detection, shape classification and semantic segmentation tasks. In the wake of the success of DL in those fields and the rapid development of 3D acquisition technologies, DL has been attracting more and more attention and proven to be a major breakthrough in point cloud classification tasks as well. Some examples of these techniques are PointNet (Qi et al., 2017), PointConv (Wu et al., 2019), PointWeb (Zhao et al., 2019), PointCNN (Li et al., 2018).

The extraction of 3D information from point clouds has played an important role in the last thirty years, i.e., in the meanwhile 3D digital data have quickly spreaded out. This task involved more subsets, that roughly may be classified in *segmentation* and *classification*. The former refers to the reorganization of the point cloud in subsets featuring similar properties. The latter consist in labeling each group of points. Both tasks may be also carried out in joint manner.

While at the beginning the scope was to filter out off-terrain data in digital surface models obtained by means of digital image correlation techniques (Gruen, 2012) or LiDAR (Kraus and Pfeifer, 1998) as far as more and more data sets have become available at an even growing resolution and accuracy, the need of extracting other types of objects has promoted relevant research. In topographic and mapping applications, two main approaches were followed, that however have been also merged and cascaded to build up complex processing pipelines. On one side, in the *data-driven* approaches (see, e.g., Forlani et al. 2006; Verma et al., 2006; Sohn et al., 2008; Crosilla et al., 2013; Guo et al., 2015), points with similar characteristics were clustered together to construct classes of homogeneous points. In *model-driven* approaches (see, e.g., Haala et al., 1998; Maas and Vosselman, 1999; Dorninger and Pfeifer, 2008), some geometric models were sought in the point cloud to detect some specific objects, such as houses, roofs, trees, etc. But with the diffusion of 3D imaging and scanning techniques able to collect point clouds of buildings (indoor and outdoor – see Previtali et al., 2018), a huge amount of data requiring a semantic interpretation at a high Level-of-Detail (LoD) transferred the interest to this kind of data as well. Even though important results were achieved by these traditional approaches, working in an efficient way for at least one category of data still faces challenges. The complexity of real data cannot be only interpreted by models or geometric descriptors. Due to this reason, the chance of using methods that could learn from the real world, such as DLNs are supposed to do, make the application of these tools really promising for segmentation and classification of 3D point clouds (see Wang et al., 2020).

Deep Learning architectures may look like a kind of "black box" to the end users. For instance, the processing workflow starts from input data (i.e., imagery and/or 3D point clouds) and through convolutions, different kinds of transformations in convolution layers/pooling layers, it ends up with some sets of class scores or some types of understandable outputs (such as labeled points, bounding box positions, etc.). However, when

* Corresponding author

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-651-2020

we look at the outputs and the learnt parameters from such a "black box", it is not always clear what is learnt by these layers inside the network. In several studies the output of segmentation/classification process is compared against benchmarks where the semantic meaning of each object is already known. Monitoring the contrastive loss, the classification error or also the class scores during training does not always prohibit the network from learning incorrect features for the expected detection, classification or segmentation tasks. Therefore, it is necessary to understand what is learnt by intermediate layers in DL networks (DLN).

A few previous works (Erhan et al., 2009; Springenberg et al., 2014; Zeiler and Fergus, 2014; Mahendran and Vedaldi, 2015; Dosovitskiy and Brox, 2016) tried to explain what happened in DLN designed for the classification and object recognition from images; see, e.g., He et al., 2016; Krizhevsky et al., 2012; Olah et al., 2018 among others. A growing number of researchers have studied the interpretability of bi-dimensional DL methods. However, there is still limited understanding of how DLN and their intermediate layers achieve their final outputs in the domain of 3D point clouds. Indeed, this type of three-dimensional spatial data sets have a totally different structure compared to images.

Images are based on a lattice structure where each pixel has a discrete position within a regular grid, which commonly refers to a rectangular matrix structure. The proximity between pixels is governed by a precise topology, where each element has its neighbours. Each pixel is also characterized by a radiometric value, which is described by single (in the case of monochromatic imagery) or multiple channels (in RGB or multispectral imagery).

In *3D point clouds* the position of a point is referred to a given spatial reference system, which can be intrinsically defined during data acquisition process (e.g., the intrinsic reference system of a laser scanning sensor (Vosselman and Maas, 2010) or the arbitrary reference system established when processing a photogrammetric block without ground constrain (Luhmann et al., 2014), or may come from the transformation into another reference systems (for example, in the case of georeferencing based on ground control points). The spatial position of a point is defined by fixing 3 degrees-of-freedom, which can be parameterized using cartesian or polar coordinates. Beyond the resolution of measured points, these are not cast into a discrete structure, and no topological relationships may be defined unless the ones based on distances between points. Each point may also have additional information, such as laser intensity (Scaioni et al., 2018), RGB value or other semantic or quantitative attributes. Of course, the additional content of each point beyond its position in space may be useful for the segmentation/classification process.

From a scientific viewpoint, the understanding of point cloud classification based on DLNs is an open issue and current knowledge about it is deeply unsatisfactory. To conduct such a quantitative explanation of how these networks work, in this paper we will investigate the visual interpretability for the decision-making process of these architectures and discuss two questions:

1. What exactly did DLNs learn from point clouds?
2. On the basis of what information in the point cloud are DLNs making a decision?

The tentative answer to these questions is given by proposing two approaches:

1. By means of reconstruction-based feature visualizations, the understanding of what DLNs learn in their intermediate layers is investigated. This method allows us to observe the evolution of features during training. From the insightful observations we have gained via this feature-visualization approach, future research directions may be inspired and supported; and
2. A 3D-CAM attribution visualization, which allows us to observe what information from point clouds pushes the decision-making process in DLNs.

2. RELATED WORK

While various kinds of DLNs are continuously developed and improved in either 2D image analysis and 3D point cloud classification, understanding of how these results are achieved has not been paid too much attention. This question has sparked the interest of various researchers and in response several approaches are emerging as ways of understanding DLNs by using *visualization* techniques. Several approaches for understanding and visualizing convolutional networks applied to 2D images have been developed in the literature, partly as a response to the common criticism that those learned features in a deep neural network are not interpretable. In general, these approaches can be divided into three groups (Olah et al., 2018):

1. methods based on the visualization of features;
2. methods based on the visualization of attributes; and
3. methods based on visualizing by reduction .

In this section we discuss methods (1) and (2), i.e., feature visualization and attribution visualization. The aim is to analyse the properties of these visualization methods to figure out if and how they can be transferred to 3D point clouds.

2.1 Feature Visualization

Feature visualization may help answer the question about what a DLN - or parts of a DLN - are looking for by generating examples. Although several methods have been proposed for visualizing the feature maps extracted by networks, in this section we will focus on the ones that we have retained most valuable.

2.1.1 Visualization by Reconstructing Representation. Feature representation is composed of all neuron activation patterns within a layer. Therefore, an idea of visualizing 2D image neural networks is to reconstruct the image through features and compare the reconstruction result with the original image to analyze which features of the image are retained in each layer of the DLN. This visualization idea regards the process of DLN feature extraction as the process of *encoding*, and the process of reconstructing the extracted features is exactly the reverse process of *encoding*, which is called the *decoding* process.

In 2015, Mahendran and Vedaldi (2015) proposed this idea of reconstructing features to visually analyze Convolutional Neural Networks (CNN). The gradient descent method with regularization terms is used to reconstruct the image of each layer, so that the visual information contained in the image features can be analyzed. Later, representation reconstruction based on UpconvNet was proposed by (Dosovitskiy and Brox, 2016), who reconstructed the image by training a UpconvNet network.

In 3D space, due to the fact that we usually work with them as abstract vectors (e.g., $a_{1,1}$ = [17, 0, 0, 0, 0, 89, 41.9, ...]), feature representations are usually hard to analyze and understand, even when they are extracted from original visual data. With feature visualization, however, we can transform these abstract vectors into more intuitive representations. In general, canonical examples are a more natural way to represent the abstractions that neural networks learn than abstract vectors. To intuitively understand and interpret these DLNs for point-cloud classification, in this paper, we visualise point cloud representations by folding the learned features into canonical point clouds. Moreover, much of the existing work on visualization is concerned with a DLN's input and output layers. However, the power of DLN lies in their hidden layers, as at every building block, the network discovers a new feature representation of the input. So, in this paper we visualize the output of the last pooling layer by forming a general 2D-to-3D mapping.

2.2 Attribution Visualization

The second approach to visualization is known as a *visualizing attribution* approach, because it relies on probing the network with attribution studies. The aim is to highlight which part of a training set is responsible for a specific activation of a DLN.

2.2.1 Visualizing by Class Activation Map (CAM). Inspired by Lin et al. (2013), CAM (Zhou et al., 2016) was proposed using global average pooling (GAP) in Network In Network (Lin et al., 2013) and GoogLeNet (Szegedy et al., 2015) to indicate which part of an image is responsible for the classification results in different networks. CAM replaced the last fully connected layer with GAP, and it shows its decision as a "saliency map". The improvement of this structure can effectively locate the important regions in the image for predicting the semantic meaning. Despite CAM has achieved a very good performance in the visualization task, it is focused on 2D image space only.

However, the above-mentioned methods just focus on 2D images. In 3D space, Huang et al. (2019) developed a 3D *class attentive interpretable mapping* (CLAIM) approach to visualize the impact of each point during the decision-making process inside PointNet CNN. CLAIM allowed us to highlight which regions of a point cloud are actually being used for different point cloud classification tasks. We would like to emphasize that while 3DCAM is not a novel technique that we propose here, the observation that it can be applied for various 3D point cloud tasks rather than images or only PointNet is, to the best of our knowledge, unique to our work.

3. METHOD

To conduct a quantitative explanation of point cloud classification process based on DLN, we started from the two questions posed in Section 1.

For the first question, we design a reconstruction network to reconstruct the features learned from different DLNs. In general, the last layer before the fully-connected Softmax layer is the most informative representation of the input data, thus we visualize the output of the last pooling layer by forming a universal 2D-to-3D mapping.

Feature visualization helps us answer what the network detects, but it does not answer how the network assembles these individual points in point cloud to arrive at later decisions, or why these decisions are made. Therefore, for the second question, we visualise the attributions of input data by 3D Classification Activation Map (3DCAM) in. We generate 3DCAMs using the global average pooling (GAP) in networks. This method has been derived from existing interpretation methods (Huang et al., 2019; Zhou et al., 2016) as a starting point and modifying them to be used in different 3D DLNs.

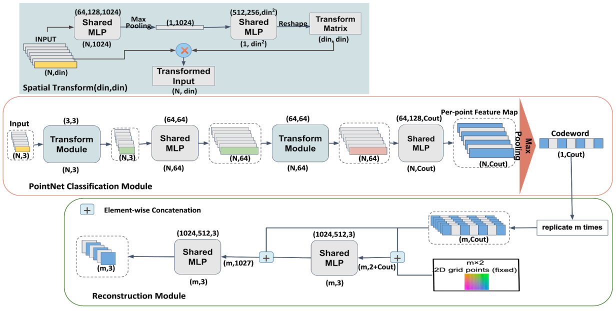

Figure 1: The architecture of our 3D reconstruction.based feature visualization network.

3.1 3D reconstruction-based feature visualization

FoldingNet (Yang et al., 2018) incorporates a decoder which is based on two consecutive 3-layer perceptrons to warp a fixed 2D grid into a point cloud. Inspired by this, we proposed a network which can learn different 3D representations of different visual abstractions. Thus, we can visualize and reconstruct the features learned from various point cloud classification DLNs by forming a universal 2D-to-3D mapping, and thus answer what these layers learned in the networks.

The procedure for generating representations of learned features in PointNet is illustrated in Figure 1. We remove the encoder of FoldingNet and replace it with other point cloud classification architectures (e.g., PointNet). The output of the last 1D convolutional layer in PointNet is passed to a feature-wise maximum to produce a C_{out}-dimensional "codeword" which is the basis for our decoder (reconstruction module). The reconstruction module in our reconstruction network is similar to FoldingNet's decoder that has two successive folding operations. The first folding operation folds the 2D manifold into 3D space, and the second one operates inside the 3D space. We have modified the decoder of FoldingNet to make it usable with different sizes of input codeword instead of a fixed size codeword (1 × 512) in FoldingNet. Before feeding the codeword into the reconstruction module, we replicate the codeword θ m times and concatenate the replicated (m, C_{out}) matrix with an $(m,2)$ matrix, which contains the m grid points (U) on a square centered at the origin. As each row of U is a two-dimensional grid point, we define the i-th row of U is u_i. Thus, the i-th row of the input matrix to the first folding operation is $[u_i,\ C]$ after above concatenation. The following folding operation essentially forms a universal 2D-to-3D mapping by 2 successive 3-multi layer perceptron (MLP). The MLP is applied in parallel to each row of the input matrix. We denote the *i*-th row of the output matrix as $f([u_i, C])$, where f is approximated by the MLPs which can be tuned by the input "codeword" and approximate multiple arbitrary 2D-3D reconstructions $f([u_i, C_1]), f([u_i, C_2])$... With point cloud classification module (PointNet) and reconstruction module, we learn a set of visual abstractions/learned features and the representations of these learned features by reconstructing them.

3.2 3DCAM-based attribution visualization

A 3D Class Activation Map (3DCAM) for a particular category indicates the discriminative point cloud regions used by DLNs to identify that category. Our network architectures depend on point cloud classification architectures, then we just perform a global average pooling (GCP) layer on the per-point feature maps before the final Max pooling layer, as GCP can retain its remarkable localization ability until the final layer, which helps us to easily identify the discriminative point cloud regions in a single forward-pass (Zhou et al., 2016). And then we append a fully-connected layer which produces weights to generate the desired classification output and produce 3DCAMs.

The procedure for generating class activation maps is illustrated in Figure 2. The classification module is the first part of our 3DCAM network, it could be different point cloud classification networks (PointNet or others), in which we will get a per-point feature map. The second part is used for generation of 3DCAM. A GCP layer is introduced to output the spatial average of the per-point feature map the first part produced. To generate the final output and class activation maps, we have projected back the weights of the classification layer onto the per-point feature maps, as illustrated in Figure 2.

For a point cloud $P\ \{p_1, p_2, \cdots, p_n\}$ made up of N unordered points, a single point in the given point cloud is defined as $p_i(x, y, z)$. The per-point feature $f(p_i)$ for the single point p_i, which was produced in the classification module, at activation unit k is denoted as $f_k(p_i)$. After we perform GCP on per-point feature map, for activation unit k, we the spatial average of the per-point feature map F_k:

$$F_k = \sum_{p_i} f_k(p_i) = \frac{1}{N}\sum_{i=1}^{N} f_k(p_i) \qquad (1)$$

Thus, for a class c, the classification score S_c can be computed:

$$Sc = \sum_k w_k^c F_k \qquad (2)$$

where w_k^c is the weight of unit k for class c, which represents the importance of F_k for class c.

Here, we define the 3DCAM M_c for class c is. In that case of a single point p_i in point cloud P, the value of $M_c(p_i)$ can be calculated as:

$$M_c(p_i) = \sum_k w_k^c f_k(p_i) \qquad (3)$$

Thus,

$$Sc = \sum_{p_i} M_c(p_i) \qquad (4)$$

Intuitively, based on prior studies (Huang et al., 2019; Zhou et al., 2016) the class activation map can be computed which is simply a weighted sum of the $f_k(p_i)$ at different 3D spatial locations. And from equation (3) and (4), we can also prove that $M_c(p_i)$ indicates the importance of the activation at $p_i(x, y, z)$ leading to the classification of a point cloud to class c.

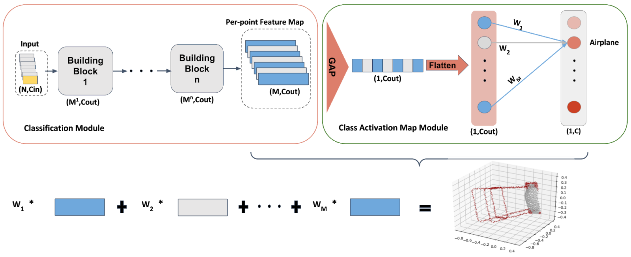

Figure 2: Illustration of the proposed 3DCAM: the predicted class score is mapped back to the previous last layer to generate the class activation maps (CAMs). The CAM highlights the class-specific discriminative regions.

4. EXPERIMENTS

4.1 3D reconstruction of learned feature

In this section, we have adopted PointNet (Qi et al., 2017) as an example of classification DLN. We train the proposed reconstruction network to visualize the learned feature in PointNet (Qi et al., 2017). Since the intermediate folding steps in the reconstruction module and the training process can be illustrated by reconstructed points, the gradual change of the folding process can be visualized. The input to our classification DLN is a point cloud with 2048 points (2048 × 3 matrix). The classification architecture can accomplish point cloud classification on the basis of different optional strategies. Here we follow the design principle of PointNet to test the methodology for visualization of reconstructed features. The following options have been set up:

- Shared MLP layers with an increasing dimension (64,64,64,128,1024) of features, implemented by five 1-D convolutional layers, each followed by a ReLU and a Batch-Normalization layer; and
- A "symmetric function:" a feature-wise Max pooling layer is followed by the MLP layers to generate a global feature representation.

The output of the last MLP layer is the input for the Max pooling layer to produce a C_{out}-dimensional abstraction vector ("codeword") which is the basis for our reconstruction network. Our reconstruction network transforms the codeword using two 3-layers MLP to produce a 2500 × 3 output. The reconstruction network is trained using the ShapeNet part dataset (Chang et al., 2015) which contains 16 categories of the ShapeNet dataset. We employ ADAM as an optimizer with an initial learning rate 0.0001, batch size 1, and weight decay 1e−6, during 250 epochs. We have trained networks to reconstruct point clouds from different feature representations. Several reconstructed point clouds after different numbers of training iterations are reported in Figure 3. From the training process, we see that initial 2D codewords can be converted to point clouds, thus we can get insight of how a learned feature looks, and answer what point cloud classification networks learned.

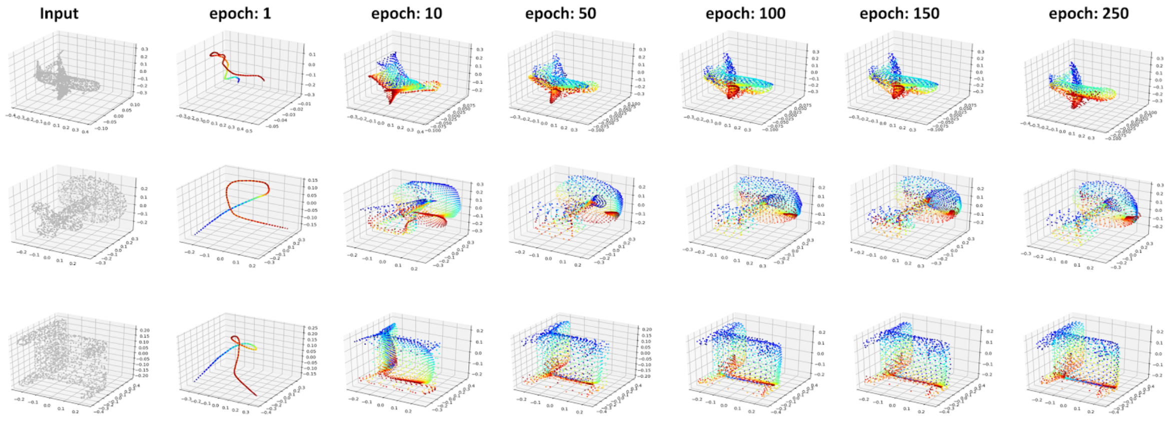

Figure 3. Illustration of the training process to show how different codewords gradually transfer into a meaningful point cloud. The left most column contains the different input features; the other columns show the reconstructed corresponding point clouds in different epochs.

4.2 3DCAM

Given the 3DCAM simple connectivity structure, we can start identifying the importance of the point cloud regions. Thus, we can answer the second question posed in the Introduction: based on what information in the point cloud is the DLNs making a decision? In this section, we evaluate the interpretation ability of 3DCAM when trained on the ModelNet40 benchmark data set (Wu et al., 2015).

For our experiments we evaluate the effect of using 3DCAM on PointNet, as done in the previous subsection. The input to our classification network is a point cloud with 1024 points (1024 × 3 matrix). We employ ADAM as an optimizer with an initial learning rate 0.001, batch size 16, and weight decay 1e−6, during 250 epochs. The setting of hidden layers is the same as PointNet, but in our implementation, we have removed the max pooling layer and fully-connected layers before the final output which can avoid lost too much information and largely decrease network parameters, respectively. Then we replace those removed layers with a GAP layer followed by a fully-connected layer to get the final classification output and class activation map. The implemented architecture is shown in Figure 4.

In Figure 5, we show some examples of the CAMs output using the above approach. We can see that the discriminative regions of the point clouds for various classes are highlighted. In Figure 6, we highlight the differences in the CAMs for a single point cloud when using different classes *c* to generate the maps. We observe that the discriminative regions for different classes are different even for a given point cloud. This suggests that our approach works as expected.

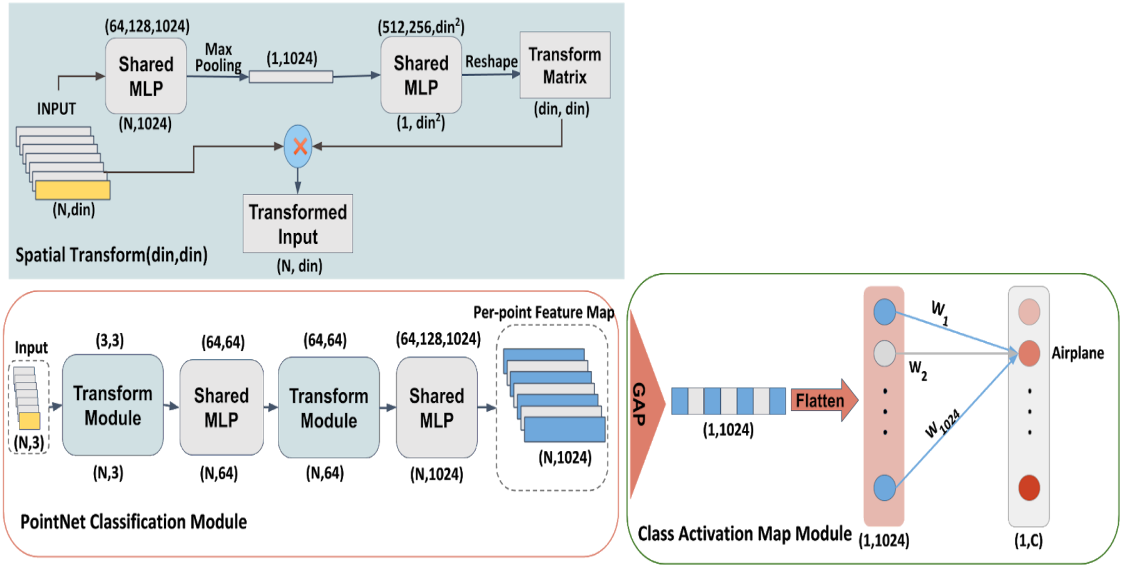

Figure 4. Illustration of PointNet-3DCAM architecture, including three parts: Spatial Transform Module, PointNet Classification Architecture, and class attentive weight computing part.

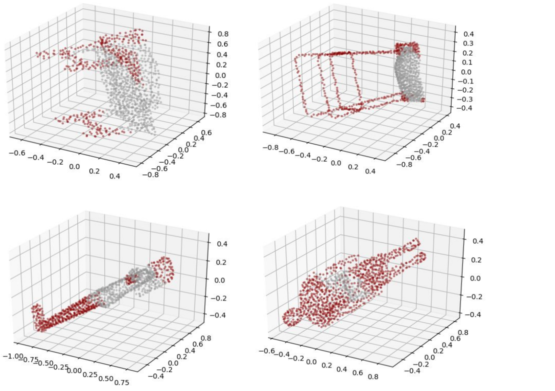

Figure 5. The CAMs of two classes from ModelNet40. The maps highlight the discriminative input point cloud regions used for point cloud classification, the chair legs for chairs and the head and leg in humans.

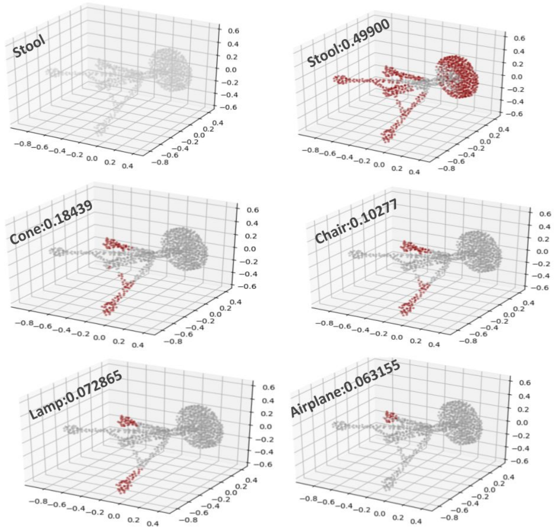

Figure 6. Examples of the CAMs generated from the top 5 predicted classes for the given point cloud with ground-truth as stool. The predicted class and its score are shown above each class activation map. We observe that the highlighted regions vary across predicted classes.

5. CONCLUSIONS

In this paper, we propose two visualization strategies to get a better understanding of Deep Learning Networks (DLNs) for point cloud classification. We first propose a reconstruction method to visualize what has been learned during the intermediate processing layers of a point cloud classification network. Then, we introduce a 3D Class Attentive Map (3DCAM) approach to visualize the discriminative regions in point clouds used by the classification networks to identify the category and understand the decision-making process. Experiments on ShapeNet and ModelNet40 indicate that the proposed visualization approaches can get a better understanding of the point cloud classification tasks.

A more extensive application of these methodologies to different categories of objects, in particular to the ones composing the urban and building environment, is planned to be accomplished. More DLNs for point cloud classification will be analysed in order to detect the most suitable architecture for classifying specific classes of objects.

Acknowledgements

Financial support from the program of China Scholarships Council (Grant No. 201906860014) is acknowledged. We are grateful to Dr. Shikun Huang from Tongji University for sharing with us the code with the method of CLAIM. We thank Dr. AX Chang et al. (Chang et al., 2015) for ShapeNet dataset and Dr. Zhirong Wu et al. (Wu et al., 2015) for ModelNet40 dataset.

REFERENCES

Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., et al., 2015. Shapenet: An information-rich 3d model repository, arXiv preprint arXiv:1512.03012.

Crosilla, F., Macorig, D., Scaioni, M., Sebastianutti, I., Visintini, D., 2013. LiDAR data filtering and classification by skewness and kurtosis iterative analysis of multiple point cloud data categories. Applied Geomatics 5(3), 225-240, DOI: 10.1007/s12518-013-0113-9.

Dorninger, P., Pfeifer, N., 2008. A comprehensive automated 3D approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds. Sensors 8 (11), 7323-7343, DOI: 10.3390/s8117323.

Dosovitskiy, A., Brox, T., 2016. Inverting visual representations with convolutional networks, In: Proceedings of the IEEE conference on computer vision and pattern recognition, Anonymous pp. 4829-4837, DOI: 10.1109/CVPR.2016.522.

Erhan, D., Bengio, Y., Courville, A., Vincent, P., 2009. Visualizing higher-layer features of a deep network. University of Montreal 1341 (3), 1.

Forlani, G., Nardinocchi, C., Scaioni, M., Zingaretti, P., 2006. Complete classification of raw LIDAR data and 3D reconstruction of buildings. Pattern Analysis and Applications 8 (4), 357-374, DOI: 10.1007/s10044-005-0018-2.

Gruen, A., 2012. Development and status of image matching in photogrammetry. The Photogrammetric Record 27 (137), 36-57, DOI: 10.1111/j.1477-9730.2011.00671.x.

Guo, B., Huang, X., Zhang, F., Sohn, G., 2015. Classification of airborne laser scanning data using JointBoost. ISPRS Journal of Photogrammetry and Remote Sensing 100 71-83, DOI: 10.1016/j.isprsjprs.2014.04.015.

Haala, N., Brenner, C., Anders, K., 1998. 3D urban GIS from laser altimeter and 2D map data. International Archives of Photogrammetry and Remote Sensing 32 339-346, DOI: 10.1.1.57.5307.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, In: Proceedings of the IEEE conference on computer vision and pattern recognition, Anonymous pp. 770-778, DOI: 10.1109/CVPR.2016.90.

Huang, S., Zhang, B., Shen, W., Wei, Z., 2019. A CLAIM Approach to Understanding the PointNet, In: Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence, Anonymous pp. 97-103, DOI: 10.1145/3377713.3377740.

Kraus, K., Pfeifer, N., 1998. Determination of terrain models in wooded areas with airborne laser scanner data. ISPRS Journal of Photogrammetry and remote Sensing 53 (4), 193-203, DOI: 10.1016/S0924-2716(98)00009-4.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, In: Advances in neural information processing systems, Anonymous pp. 1097-1105, DOI: 10.1145/3065386.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018. Pointcnn: Convolution on x-transformed points, In: Advances in neural information processing systems, Anonymous pp. 820-830, arXiv:1801.07791.

Lin, M., Chen, Q., Yan, S., 2013. Network in network. arXiv preprint arXiv:1312.4400.

Luhmann, T., Robson, S., Kyle, S., Boehm, J., 2013. Close-range photogrammetry and 3D imaging, Berlin, Boston: De Gruyter, DOI: 10.1515/9783110607253.

Maas, H., Vosselman, G., 1999. Two algorithms for extracting building models from raw laser altimetry data. ISPRS Journal of photogrammetry and remote sensing 54 (2-3), 153-163, DOI: 10.1016/S0924-2716(99)00004-0.

Mahendran, A., Vedaldi, A., 2015. Understanding deep image representations by inverting them, In: Proceedings of the IEEE conference on computer vision and pattern recognition, Anonymous pp. 5188-5196, DOI: 10.1109/CVPR.2015.7299155.

Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., et al., 2018. The building blocks of interpretability. Distill 3 (3), e10, DOI: 10.23915/distill.00007.

Previtali, M., Barazzetti, L., Brumana, R., Cuca, B., Oreni, D., Roncoroni, F., Scaioni, M., 2014. Automatic façade modelling using point cloud data for energy efficient retrofitting." Applied Geomatics 6(2), 95-113, DOI: 10.1007/s12518-014-0129-9.

Previtali, M., Díaz-Vilariño, L., Scaioni, M., 2018. Indoor Building Reconstruction from Occluded Point Clouds Using Graph-Cut and Ray-Tracing. Applied Sciences 8(9), paper No. 1529, DOI: 10.3390/app8091529.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation, In: Proceedings of the IEEE conference on computer vision and pattern recognition, Anonymous pp. 652-660, DOI: 10.1109/CVPR.2017.16.

Scaioni, M., Höfle, B., Baungarten Kersting, A.P., Barazzetti, L., Previtali, M., Wujanz, D., 2018. Methods for Information Extraction from Lidar Intensity Data and Multispectral Lidar Technology. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., Vol. XLII, Part 3, 1503-1510, DOI: 10.5194/isprs-archives-XLII-3-1503-2018.

Sohn, G., Huang, X., Tao, V., 2008. Using a binary space partitioning tree for reconstructing polyhedral building models from airborne lidar data. Photogrammetric Engineering & Remote Sensing 74 (11), 1425-1438, DOI: 10.1.1.364.1114.

Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M., 2014. Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al., 2015. Going deeper with convolutions, In: Proceedings of the IEEE conference on computer vision and pattern recognition, Anonymous pp. 1-9, DOI: 10.1109/CVPR.2015.7298594.

Verma, V., Kumar, R., Hsu, S., 2006. 3d building detection and modeling from aerial lidar data, In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), Anonymous IEEE, pp. 2213-2220, DOI: 10.1109/CVPR.2006.12.

Vosselman, G., Maas, H., 2010. Airborne and terrestrial laser scanning, CRC press.

Wang, D., Wang, J., Scaioni, M., Si, Q., 2020. Coarse-to-fine classification of road infrastructure elements from mobile point clouds using symmetric ensemble point network and Euclidean Cluster Extraction. Sensors 20, paper No. 225, DOI: 10.3390/s20010225.

Wu, W., Qi, Z., Fuxin, L., 2019. Pointconv: Deep convolutional networks on 3d point clouds, In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anonymous pp. 9621-9630, DOI: 10.1109/CVPR.2019.00985.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., et al., 2015. 3d shapenets: A deep representation for volumetric shapes, In: Proceedings of the IEEE conference on computer vision and pattern recognition, Anonymous pp. 1912-1920, DOI: 10.1109/CVPR.2015.7298801.

Yang, Y., Feng, C., Shen, Y., Tian, D., 2018. Foldingnet: Point cloud auto-encoder via deep grid deformation, In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anonymous pp. 206-215, DOI: 10.1109/CVPR.2018.00029.

Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks, In: European conference on computer vision, Anonymous Springer, pp. 818-833, DOI: 10.1007/978-3-319-10590-1_53.

Zhao, H., Jiang, L., Fu, C., Jia, J., 2019. PointWeb: Enhancing local neighborhood features for point cloud processing, In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anonymous pp. 5565-5573, DOI: 10.1109/CVPR.2019.00571.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A., 2016. Learning deep features for discriminative localization, In: Proceedings of the IEEE conference on computer vision and pattern recognition, Anonymous pp. 2921-2929, DOI: 10.1109/CVPR.2016.319.