Convolutional Neural Network Model for Stratification of LiDAR 3D Point Cloud Data

PL. Chithra[1], S. Lakshmi Bala[2]

{chitrasp2001@yahoo.com[1], lakshmibala.sj@gmail.com[2]}

Professor, Department of Computer Science, University of Madras, Chennai – 600 025[1],
Ph.D Research Scholar, Department of Computer Science, University of Madras, Chennai – 600 025[2]

Abstract. Analysis of the 3D LiDAR point cloud data is the fundamental for computer vision and robotics, with potential applications ranging from object detection to scene understanding. In this proposed work, introduced an approach that leverages the power of contrastive learning combined with self-supervised learning for feature extraction from the LiDAR 3D point cloud data. On-the-fly extensions to include rotation on invariance of 3D point cloud data, jitter adding noise in coordinates of point cloud data, and displacement and shear to improve model robustness and invariance to geometric transformations. Each data point is paired with two extensions to form positive pairs and facilitating the contrastive learning. The focus of proposed method is to implement the point cloud feature learning layer, specifically on DynamicEdgeConv. This layer uses the parameter 'k' to identify nearest neighbors in 3D point cloud data that together create a subgraph, facilitating improved feature learning. Applying enhancements during training is important that ensures the adaptability of model to real-world data fluctuations. Evaluation results determines the efficacy of proposed approach and that shows that significant improvements in clustering performance and outlier detection accuracy compared to common methods. This work has advances of the state of self-supervised learning of LiDAR 3D point cloud data, but also paves the way for more robust 3D object detection, scene understanding, and applications in the fields of robotics.

Keywords: 3D point cloud data, LiDAR, Contrastive learning, Self-supervised learning, DynamicEdgeConv, Outlier detection.

1 Introduction

LiDAR is the Light Detection and Ranging sensor used for obtaining the 3D point cloud dataset. 3D point cloud data has become an important source of information for applications in a variety of fields, from autonomous navigation and robotics to augmented reality and environmental modeling. Accurately and efficiently classifying objects and scenes represented by 3D point clouds is a fundamental task with significant impact on real-world systems. However, this task is inherently difficult because point cloud data is unstructured and sparse. Although traditional supervised learning methods had great success in classifying 2D images,

IACIDS 2023, November 23-25, Lavasa, India

DOI 10.4108/eai.23-11-2023.2343218

they face limitations when applied to 3D data. Additionally, the rich spatial and geometric information encoded in these point clouds may be underutilized. Self-supervised learning, a paradigm that exploits the inherent structure of data to generate monitoring signals, represents a promising way to address these challenges. In 3D point cloud classification, self-supervised learning algorithms aim to automatically learn distinctive features directly from the data. Raw data eliminates the need for extensive manual annotation. This proposed work addresses a self-supervised learning for 3D point cloud classification. The proposed model introduced an approach that leverages the power of self-supervised learning techniques to extract meaningful features from unannotated 3D point cloud data. Through careful design and research, the aim to address several key challenges in this field, including:

- **Feature Learning:** The strategy focuses on creating a framework for self-supervised learning that is specifically adapted to the unique features of 3D point cloud data. The goal is to learn a highly descriptive and compact representation that captures the underlying geometry and semantics.
- **Data efficiency:** By eliminating dependence on large labeled datasets, the method significantly reduces data annotation effort, making it suitable for practical applications.
- **Robustness to variation:** Real-world 3D data is subject to significant variation due to factors such as lighting conditions, sensor noise, and changes in viewing angle. The approach includes mechanisms that improve the model's robustness and adaptability to these challenges.
- **Generalization:** The goal is to create a model that generalizes well across a variety of 3D point cloud datasets and scenarios, leading to broader applications in areas such as robotics, self-driving cars, and augmented reality.

The following sections are organized as Section II gives an overview of related work in 3D point cloud processing, geometric transformation, contrastive learning in LiDAR 3D point cloud data, and Dynamic Edge convolution model. Section III describes a methodology for Classification model for 3D point cloud data and describes the self-supervised learning model architecture. Section IV presents experimental results and performance analysis and evaluations on the benchmark dataset, and discussing the results. Finally, Section V concludes the paper by highlighting the contributions and possible of future directions of this research.

2 Related Work

For a range of computer graphics applications, point clouds offer a versatile geometric representation. includes real-time use of the majority of 3D data capturing tools. Features that were hand-designed are highlighted. CNNs for image analysis point to the benefits of translating CNN-derived insights to point clouds. They described the EdgeConv neural network module, which is appropriate for CNN-based applications on point clouds. There is a description of classification and segmentation [1]. For analysing 3D point clouds, they described a graph neural network (GNN) framework. Vertex employing MLP to represent points with significant local geometric information after nonlinear projection. The second contribution is a proposed model that enhances 3D point cloud GNN graph creation [2]. Natural language processing is being revolutionised by self-attention networks, which are also considerably advancing image analysis tasks like image categorization and object recognition. This achievement led to research into the use of self-attention networks for processing 3D point clouds [3]. The general architecture of graphical neural networks (GNNs) is given for analysing 3D point clouds. The construction of local neighbourhood graphs and advancements in point representation are described [4]. Machine learning is utilised in the infrastructure and construction sectors to automate procedures. A plan for future research is discussed in light of the found research gaps [5]. The sector of developing, protecting, and maintaining data for cultural property is seeing an increase in the importance of automatic segmentation and classification models [6]. A deep neural network called PointNet receives 3D point cloud data directly as input. Due to its great stability and computational efficiency, PointNet has emerged as one of the most popular point cloud categorization algorithms for practical applications [7]. A dynamic graph convolutional neural network (DGCNN) point-wise deep learning approach was created and deployed for 3D point cloud classification, extending its classification applicability from indoor scenarios to aerial point cloud data. This was described [8]. Large human-annotated datasets are utilised in conjunction with aircraft LiDAR point cloud classification [9]. a 3D point cloud classification technique based on segmentation. For the purpose of data training, this supervised technique needs a simple data description [11]. The foundation of point cloud analysis is point cloud categorization, and numerous deep learning-based approaches are being discussed[12]. discusses how to automatically interpret 3D point cloud data by giving each 3D point cloud a class label [13]. We have discussed the use of 3D convolutional neural networks (CNNs) for feature extraction as well as a voxel-based technique for transforming point clouds into normal 3D grids (voxels) [16]. Unreliable points are successfully hidden and the spatial breadth is decreased when statistical subspace outlier detection and logarithmic transformation (LPCT) are used together. outlined a deep learning model for the quantization of 3D LiDAR point cloud taken from the air [15].

3 Methodology

This methodology includes as step 1 in data preprocessing to improve the robustness of point cloud data, step 2 is self-supervised learning to extract meaningful features, step 3 as clustering for data reduction, and task-specific adaptation. Includes fine-tuning as step 4 and step 5 is outlier detection to ensure data quality. Integrating a point cloud feature learning layer like EdgeConv plays an important Role in feature extraction from point cloud data. Taken together, these steps enable the model to effectively learn and use 3D point cloud analysis classification representations. In further the steps are elaborated briefly:

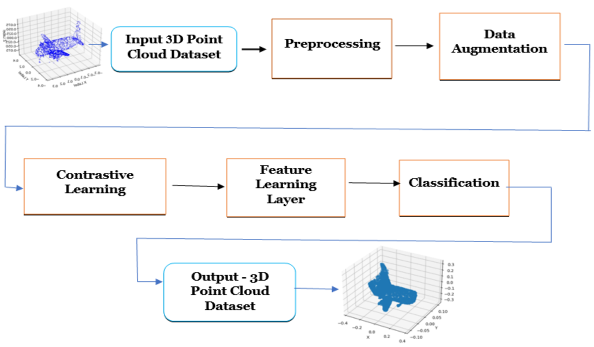

Fig. 1. Architecture of 3D point cloud data for proposed model

3.1 Description of proposed Architecture:

The diagram above shows the proposed architecture of a 3D point cloud dataset for classifying data. In the first process of this methodology, 3D point cloud data is preprocessed and converted into tensors. A further process is data augmentation of 3D point cloud data. The feature learning layer is an edge convolution layer and is used as a convolutional neural network on 3D point cloud data. With the help of edge convolution layers, feature learning is performed through self-supervised learning on 3D point cloud data. Classification is performed by determining outlier detection using a K-nearest neighbor algorithm on point cloud data.

3.2 Steps used in Proposed work

Step:1 Data preprocessing

Before training the model, the raw 3D point cloud data undergoes a preprocessing step. This includes data augmentation techniques that improve model robustness and processing of point cloud features:

1. **Rotation** (if the plane used is not rotation invariant): A random rotation is applied to the 3d point cloud data for variations in the point cloud orientation. This step allows the model to generalize across different orientations of objects in the point cloud. Randomly rotating each point cloud by a random transformation matrix (R):

$$R = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

   Applied the rotation to each point p in the 3D point cloud:

$$p' = Rp \qquad (2)$$

2. **Jittering:** Noise is added to the coordinates of a point cloud data, effectively simulating the presence of measurement errors or measurement uncertainties. This makes the model more robust and allows it to learn from noisy real-world data. Adding Gaussian noise to the coordinates of each point:

$$p' = p + \in . n \qquad (3)$$

   Where $\in$ is the noise level and n is a vector of random values sampled from a Gaussian distribution.

3. **Displacement and Shear**: Applying transformations and shear transformations to point cloud data further improves resilience to spatial variations. This is especially useful for correcting positional differences and non-uniformities in point cloud data.
   Shear in 3D point cloud is represented using an affine transformation matrix. A shear transformation involves moving one or more coordinates of an object in proportion to other coordinates.

i. **X-Shear:** The X-shear transform shears points along the X axis without changing the Y and Z coordinates.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \alpha & \beta & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad (4)$$

Here, α represents the shearing factor along the X-axis, and β is typically set to 0 for X-shearing.

ii. Y-Shear: The Y-shear transform shears points along the Y axis without changing the Z and X coordinates.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \alpha & 1 & \beta & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{5}$$

Here, α represents the shearing factor along the Y-axis, and β is typically set to 0 for Y-shearing.

iii. **Z-Shear:** The X-shear transform shears points along the Z axis without changing the X and Y coordinates.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \alpha & \beta & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{6}$$

Here, α represents the shearing factor along the Z-axis, and β is typically set to 0 for Z-shearing.

4. Point cloud feature learning layers (EdgeConv): These layers are integrated into the model architecture captures the local and global functionality from point cloud data. The EdgeConv facilitates the neighborhood of aggregation and feature extraction, allowing models to learn meaningful representations from input data.

Step: 2 Self-supervised learning

Self-supervised learning is used as the main learning paradigm for feature extraction. Point cloud data is split into positive and negative pairs. Positive pairs represent instances with common characteristics, and negative pairs represent instances that are different. The model is trained to maximize the similarity between positive pairs and minimize the similarity between negative pairs, thereby learning to distinguish features from the point cloud data for classification. Let $f(p)$ be the feature embedding of a point p obtained from the Point Cloud Feature-Learning layers used in EdgeConv layer:

Contrastive Loss:

For a pair of positive samples $p_i$and $p_j$and negative samples p_k:

Defining the similarity function $s(p_i, p_j)$ cosine similarity between the feature embeddings:

$$s(p_i, p_j) = \frac{f(p_i).f(p_j)}{||f(p_i)||\ ||f(p_j)||} \tag{7}$$

Step:3 Clustering

After an initial self-supervised learning phase, the learned feature representations are clustered with the help of KNN algorithm to group similar data points. Clustering helps organize data and serves as the basis for downstream tasks such as fine-tuning and outlier detection.

Using clustering algorithms like KNN (K-Nearest neighbor) to group the feature embeddings $f(p)$ into clusters. This can be represented as:

$$Cluster\ assignment: \quad C = Cluster\left(f(p_1)\right), \left(f(p_2)\right), \dots, \left(f(p_n)\right) \tag{8}$$

Step:4 Fine adjustment

Fine-tuning is performed on clustered data to adapt the model's learned features to a specific task of 3D point cloud classification. This phase includes self-supervised training on labeled data to adapt the model's features to classification tasks of 3D point cloud data.

$$Fine - Tuning\ Loss: \ L_{fine-tune} = L_{task}(model(p_i), label_i) \tag{9}$$

where model (p_i) represents the contrastive learning model's prediction for point (p_i) 3D point cloud data and $label_i$ is the ground truth label for particular point cloud data.

Step:5 Detecting outliers

Outlier detection mechanisms are applied to identify and handle outliers in point cloud data. These are EdgeConv machine learning models, dynamic graph CNN models specifically designed to label the data points that deviate significantly from expected patterns. Converting 3D point cloud data into a graphical representation. Each data point (3D point) becomes a node in the chart. Graph edges are defined based on similarity or adjacency metrics specific to 3D point cloud data. This is done using spatial proximity to generate an adjacency matrix A. Use graph convolutional networks (GCN) for outlier detection. GCN is a type of neural network designed for graph-structured data. The GCN layer allows information to pass through the graph and capture local and global relationships within 3D point cloud data.

4 Performance Analysis and Evaluations

The performance of the proposed convolutional neural network model includes contrastive learning on 3D point cloud data for classification self-supervised learning. The proposed work has been evaluated the effectiveness of our method, including on-the-fly extensions aimed at improving model robustness and invariance to geometric transformations. The experiments aim to verify the benefits of combining contrastive and self-supervised learning for this particular task. The proposed Model architecture: is to feature extraction model is based on the concept of DynamicEdgeConv, which uses the parameter “k” to identify nearest neighbour points by using the KNN algorithm in 3D point cloud data and create subgraphs to improve feature learning. Adam optimizer is used for learning model, learning rate is auto and epochs is 30 and had the loss of 1.1936 When training on a 3D point cloud dataset, the accuracy is 90.88%. The proposed model has higher accuracy than state-of-the-art classification algorithms. The proposed model integrates contrastive learning, which improves the accuracy percentage. The training of the proposed model was trained using the training parameter “k”. The batch size is 32, the input dataset learning rate is 0.01, and the number of epochs is 30. The 3D LiDAR point cloud dataset is implemented using the Jupyter environment in Python 3.10.9 and used for the PyTorch implementation. The model represents the quantitative results of experiments using various performance metrics.

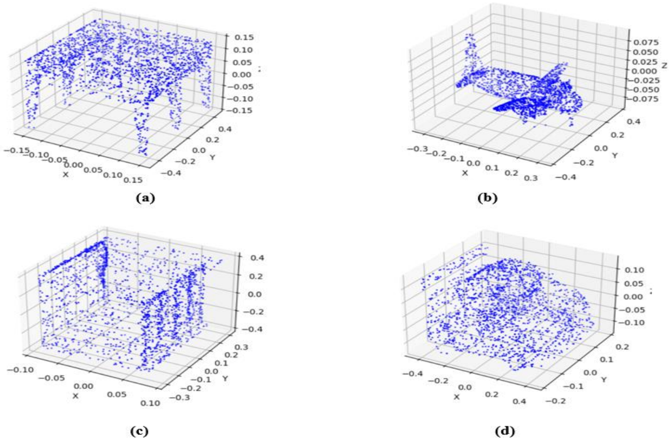

Fig. 2. Original ShapeNet Dataset - Input sample images from 3D point cloud datasets (a) Teapoy.ply (b) airplane.ply (c)Nightstand.ply and (d) car.ply

The above figure shows the Original ShapeNet dataset of Different shapes that represents the 3D point cloud dataset (a) Teapoy.ply shows the Teapoy shape (b) airplane.ply shows the Airplane shape (c) Nightstand.ply shows the Table shape and (d) car.ply represents the car shape.

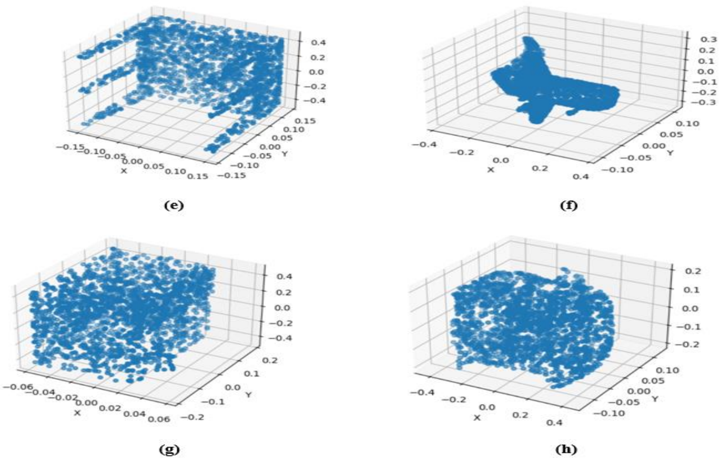

Fig. 3. Output sample images from 3D point cloud datasets (e) Teapoy.ply (f) airplane.ply (g)Nightstand.ply and (h) car.ply. Using the Jupyter environment in Python 3.10.9 and PyTorch, the proposed convolutional neural network model was constructed and trained on dense 3D LiDAR point cloud ShapeNet datasets of various sizes. The dataset is a.Ply file that contains a large amount of training data for the suggested model and is composed of 3D object point cloud data.

5 Conclusion

Contrastive learning and self-supervised learning while combined, have verified to be exceptionally powerful in extracting significant capabilities from 3D point cloud data. These capabilities function a robust basis for downstream tasks, consisting of clustering and fine-tuned outlier detection. The inclusion of on-the-fly augmentations has substantially advanced of the model`s robustness to numerous geometric transformations, consisting of rotation and jittering, normally encountered in real-time scenarios. The implementation of DynamicEdgeConv has been a pivotal advancement, making an allowance for the extraction of neighborhood contextual data via the formation of subgraphs. The parameter 'k' aids withinside the identity of nearest neighbors, contributing to extra informative characteristic from data. First-rate upgrades in clustering overall performance and outlier detection accuracy is 90.88% while as compared to present methods. These effects have the viability of the proposed method for real-time 3D point cloud data evaluation tasks. Future work ought into optimizing augmentation strategies, scaling the method to large datasets, and increasing its utility to fields consisting of autonomous navigation and robotics.

References

[1] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic Graph CNN for Learning on Point Clouds," arXiv:1801.07829 [cs], Jun. 2019, Available: https://arxiv.org/abs/1801.07829.

[2] "Papers with Code - Exploiting Local Geometry for Feature and Graph Construction for Better 3D Point Cloud Processing with Graph Neural Networks," paperswithcode.com. https://paperswithcode.com/paper/exploiting-local-geometry-for-feature-and-graph-construction accessed Sep. 15, 2023).

[3] "Papers with Code - Point Transformer," paperswithcode.com. https://paperswithcode.com/paper/point-transformer-1 (accessed Sep. 15, 2023).

[4] "Papers with Code - Exploiting Local Geometry for Feature and Graph Construction for Better 3D Point Cloud Processing with Graph Neural Networks," paperswithcode.com. https://paperswithcode.com/paper/exploiting-local-geometry-for-feature-

[5] M. Pepe, V. S. Alfio, D. Costantino, and D. Scaringi, "Data for 3D reconstruction and point cloud classification using machine learning in cultural heritage environment," Data in Brief, vol. 42, p. 108250, Jun. 2022, doi:https://doi.org/10.1016/j.dib.2022.108250.

[6] M. Pepe, V. S. Alfio, D. Costantino, and D. Scaringi, "Data for 3D reconstruction and point cloud classification using machine learning in cultural heritage environment," Data in Brief, vol. 42, p. 108250, Jun. 2022, doi: https://doi.org/10.1016/j.dib.2022.108250.

[7] X. Wu, X. Liu, J. Wang, Z. Lai, J. Zhou, and X. Liu, "Point cloud classification based on transformer," Computers and Electrical Engineering, vol. 104, p. 108413, Dec. 2022, doi: https://doi.org/10.1016/j.compeleceng.2022.108413.

[8] E. Widyaningrum, Q. Bai, M. K. Fajari, and R. C. Lindenbergh, "Airborne Laser Scanning Point Cloud Classification Using the DGCNN Deep Learning Method," Remote Sensing, vol. 13, no. 5, p. 859, Feb. 2021, doi: https://doi.org/10.3390/rs13050859.

[9] X. Li, C. Wen, Q. Cao, Y. Du, and Y. Fang, “Retracted: A novel semi-supervised method for airborne LiDAR point cloud classification,” vol. 180, pp. 117–129, Oct. 2021, doi: https://doi.org/10.1016/j.isprsjprs.2021.08.010.
[10] https://shapenet.org/ - ShapeNet Dataset for 3D point cloud dataset.
[11] Sağlam and N. A. Baykan, Segmentation-Based 3D Point Cloud Classification on a Large-Scale and Indoor Semantic Segmentation Dataset. Springer Science and Business Media Deutschland GmbH, 2021. Accessed: Sep. 15, 2023. [Online]. Available: https://gcris.ktun.edu.tr /handle/ 20.500.13091/ 1210.
[12] H. Zhang et al., “Deep learning-based 3D point cloud classification: A systematic survey and outlook,” Displays, vol. 79, p. 102456, Sep. 2023, doi: https://doi.org/10.1016/j.displa.2023.102456.
[13] M. Weinmann, B. Jutzi, C. Mallet, and M. Weinmann, “Geometric Features And Their Relevance For 3d Point Cloud Classification,” ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. IV–1W1, pp. 157–164, May 2017, doi: https://doi.org/10.5194/isprs-annals-IV-1-W1-157-2017.
[14] PL. Chithra and A. C. Tamilmathi, “3D LiDAR point cloud image codec based on Tensor,” The Imaging Science Journal, vol. 68, no. 1, pp. 1–10, Jan. 2020, doi: https://doi.org/10.1080/13682199 . 2020.1719747.
[15] A.C. Tamilmathi and P. L. Chithra, “Deep Learned Quantization-Based Codec for 3D Airborne LiDAR Point Cloud Images,” Frontiers in Robotics and AI, vol. 8, May 2021, doi: https://doi.org/10.3389/frobt.2021.606770.
[16] C.Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. Guibas, “Volumetric and Multi-View CNNs for Object Classification on 3D Data.” Accessed: Aug. 09, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2016/papers/Qi_Volumetric_and_Multi-View_CVPR_2016_paper.pdf.