



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 3, March 2025

Point Cloud Processing using Deep Learning

Gauri S Nair

U.G. Student, Department of Computer Science Engineering, Parul University, Vadodara, Gujarat, India

ABSTRACT: This paper explores the use of PointNet, a novel deep learning method, in classifying and segmenting 3D point clouds. Point clouds, which are geometric descriptions of objects in terms of collections of 3D coordinates, have distinct challenges since they are unordered and irregular. PointNet first processes these raw sets of points without going through intermediate voxel or mesh representation. The performance of the network is tested with the ModelNet10 dataset, exhibiting 83.05% accuracy at 20 training iterations. The results validate the effectiveness of PointNet in obtaining global and local features from point clouds, making it a promising contribution to various fields of computer vision, such as autonomous navigation, robotics, and 3D object recognition. Although PointNet is superior in end-to-end learning and geometric invariance, the paper also mentions the development of more sophisticated architectures such as PointNet++ and DGCNN, which overcome some of its drawbacks. This work adds to the expanding body of 3D data understanding by confirming PointNet's status as a baseline model for point cloud analysis.

KEYWORDS: Point Cloud Processing, Deep Learning, PointNet, 3D Object Classification, 3D Segmentation, Computer Vision, Feature Extraction, Semantic Segmentation, Point Cloud

I. INTRODUCTION

A Point cloud is a collection of data points defined in a three-dimensional coordinate system, representing the external surface of objects or spaces. Each point within the cloud has its own set of X, Y, and Z coordinates capturing the precise location in 3D space. In Computer Vision, detecting and recognizing objects in 3D space is more reliable with point clouds due to the additional depth information. Point clouds are commonly generated by 3D scanning technologies such as LiDAR, photogrammetry, or structured light scanning, which capture detailed spatial information about an object's shape and appearance. They are widely used in various fields, including computer graphics, robotics, autonomous driving, and geospatial analysis, for applications like 3D modelling, mapping, object recognition, and environmental analysis. Point cloud is an unstructured data structure that has to be processed by either a Convolutional Neural Network (CNN) or a classification network. Understanding the intricacies of point clouds is crucial, as they form the foundation for advanced deep learning techniques like PointNet, which transforms the processing and analysis of 3D geometric data.

PointNet is a Deep Learning architecture specifically designed for processing Point Clouds. It is used to create a network that directly processes the 3D points without any intermediate representation like 3D meshes or voxels that transform 3D points into another representation that is easier to process for the system. This architecture consists of an input and feature transformation network, a shared multi-layer perceptron (MLP) for extracting local features, and a symmetric function for generating a global feature vector.

This global feature vector is used for tasks such as classification, part segmentation, and semantic segmentation. Part segmentation would label points as belonging to the seat, backrest, legs, etc. PointNet performs semantic segmentation by first learning the global context features of the scene and then combining these features with local point features; segmentation would label points as belonging to categories.

II. LITERATURE SURVEY

The concept of point cloud processing has evolved significantly over the years to address the challenges posed by 3D data. Traditional methods were widely used for registration and object recognition tasks in point clouds [1,2]. These methods relied heavily on handcrafted features and classical machine learning techniques, often requiring manual tuning and domain knowledge.

With the advent of deep learning, researchers began to adapt neural networks to point cloud data. Volumetric methods, which convert point clouds into 3D voxel grids for processing with 3D CNNs, were introduced to address

data irregularity [3]. However, these methods suffered from high memory consumption due to sparsity. Multi-view methods followed, projecting point clouds into multiple 2D views for processing by standard CNNs [5]. These approaches improved efficiency but often lost essential 3D spatial information.

Graph-based techniques emerged as another alternative, where point clouds are represented as graphs and processed using Graph Neural Networks (GNNs) to capture local geometric structures. While more effective in capturing relationships, they required complex graph construction.

A breakthrough came with the introduction of PointNet by Qi et al. [1], which directly processes raw point clouds without voxelization or projections. PointNet employs a symmetric function (max pooling) to ensure permutation invariance and combines local and global feature extraction through shared Multi-Layer Perceptrons (MLPs). This model significantly improved classification and segmentation performance on 3D datasets.

Further improvements were made with models such as PointNet++ [3,4], which introduced hierarchical feature learning by recursively applying PointNet on nested point subsets to capture local features more effectively. These advancements have enabled point cloud processing to play a vital role in domains like autonomous driving, robotics, and medical imaging, where 3D data interpretation is crucial.

III. METHODOLOGY

PointNet for 3D Object Classification: Once the data is gathered a five-step approach can be used.

Input Representation: PointNet takes raw point clouds as input, where a point cloud is a set of points $\{(x_i, y_i, z_i)\}$ representing the 3D coordinates of points in the space. Optionally, each point can have additional features such as color or normal vectors.

Feature Extraction: Shared Multi-Layer Perceptron (MLP): Each point in the point cloud is processed independently using a shared MLP. This MLP maps each point to a higher-dimensional feature space. **Input and Feature Transformations (T-Net):** The input points and features are transformed using learned affine transformations to align them into a canonical space, enhancing the network's invariance to geometric transformations.

Global Feature Aggregation: Symmetric Function (Max Pooling): To handle the unordered nature of point clouds and ensure permutation invariance, PointNet applies a symmetric function such as max pooling. This aggregates local features from all points into a single global feature vector, capturing the overall shape and structure of the object.

Classification Network: For object classification, the global feature vector is fed into a fully connected network. This network produces a probability distribution over the possible object categories, allowing the network to classify the entire 3D object.

Segmentation Network: For segmentation tasks, both global and local features are used. The global feature vector is concatenated with the local features of each point. The concatenated features are then processed by additional MLP layers to predict a label for each point in the point cloud. This enables the network to segment the object into different parts (part segmentation) or label each point with a semantic category (semantic segmentation).

IV. EXPERIMENTAL RESULTS

The data was fetched from ModelNet10 - Princeton 3D Object Dataset. There are 4,899 pre-aligned shapes and 10 object categories, split into 3,991 (80%) shapes for training and 908 (20%) shapes for testing. The model was trained and evaluated on the ModelNet10 dataset.

After 20 epochs (iterations), Training on ModelNet took 8 hours to complete with TensorFlow and a T4 GPU (on Google collab).

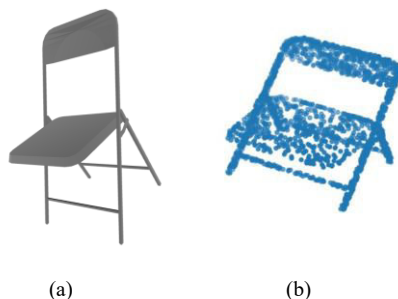


Fig. 1. Image Detection (a) Original image - Chair image from ModelNet10 dataset (b) Chair Point cloud from ModelNet10 dataset

```
model.compile(
    loss='sparse_categorical_crossentropy',
    optimizer=keras.optimizers.Adam(learning_rate=0.001),
    metrics=['sparse_categorical_accuracy'],
)

model.fit(train_dataset, epochs=20, validation_data=validation_dataset)
```

```
*** Epoch 1/20
100/100 [=====] - 527s 5s/step - loss: 3.5535 - sparse_categorical_accuracy: 0.2888 - val_loss: 95197615108398912.0000 - val_sparse_categorical_accuracy: 0.4500
Epoch 2/20
100/100 [=====] - 518s 5s/step - loss: 3.0797 - sparse_categorical_accuracy: 0.3662 - val_loss: 20474150.0000 - val_sparse_categorical_accuracy: 0.3967
Epoch 3/20
100/100 [=====] - 510s 5s/step - loss: 3.0003 - sparse_categorical_accuracy: 0.3803 - val_loss: 91251.5078 - val_sparse_categorical_accuracy: 0.3179
Epoch 4/20
100/100 [=====] - 512s 5s/step - loss: 2.7691 - sparse_categorical_accuracy: 0.4502 - val_loss: 2848136302241456979968.0000 - val_sparse_categorical_accuracy: 0.6744
Epoch 5/20
100/100 [=====] - 506s 5s/step - loss: 2.5828 - sparse_categorical_accuracy: 0.5376 - val_loss: 1515504621756585345024.0000 - val_sparse_categorical_accuracy: 0.6744
Epoch 6/20
100/100 [=====] - 502s 5s/step - loss: 2.5135 - sparse_categorical_accuracy: 0.5542 - val_loss: 2497.8535 - val_sparse_categorical_accuracy: 0.4981
Epoch 7/20
100/100 [=====] - 495s 5s/step - loss: 2.4063 - sparse_categorical_accuracy: 0.5981 - val_loss: 1429902592576828997632.0000 - val_sparse_categorical_accuracy: 0.6744
Epoch 8/20
100/100 [=====] - 507s 5s/step - loss: 2.2233 - sparse_categorical_accuracy: 0.6466 - val_loss: 11723089380527235399680.0000 - val_sparse_categorical_accuracy: 0.6744
Epoch 9/20
100/100 [=====] - 506s 5s/step - loss: 2.2571 - sparse_categorical_accuracy: 0.6526 - val_loss: 7304964665769984.0000 - val_sparse_categorical_accuracy: 0.6744
Epoch 10/20
100/100 [=====] - 530s 5s/step - loss: 2.0602 - sparse_categorical_accuracy: 0.6999 - val_loss: 1328585804452200448.0000 - val_sparse_categorical_accuracy: 0.6744
Epoch 11/20
100/100 [=====] - 531s 5s/step - loss: 2.0084 - sparse_categorical_accuracy: 0.7112 - val_loss: 600341568.0000 - val_sparse_categorical_accuracy: 0.8085
Epoch 12/20
```

(a)

```
*** Epoch 13/20
100/100 [=====] - 533s 5s/step - loss: 1.9288 - sparse_categorical_accuracy: 0.7469 - val_loss: 538174623552897024.0000 - val_sparse_categorical_accuracy: 0.7459
Epoch 14/20
100/100 [=====] - 511s 5s/step - loss: 1.9074 - sparse_categorical_accuracy: 0.7528 - val_loss: 51.4937 - val_sparse_categorical_accuracy: 0.8173
Epoch 15/20
100/100 [=====] - 502s 5s/step - loss: 1.7950 - sparse_categorical_accuracy: 0.7773 - val_loss: 274272911360.0000 - val_sparse_categorical_accuracy: 0.7459
Epoch 16/20
100/100 [=====] - 511s 5s/step - loss: 1.8193 - sparse_categorical_accuracy: 0.7788 - val_loss: 4634486.5000 - val_sparse_categorical_accuracy: 0.8385
Epoch 17/20
100/100 [=====] - 513s 5s/step - loss: 1.7435 - sparse_categorical_accuracy: 0.7945 - val_loss: 17.2702 - val_sparse_categorical_accuracy: 0.7109
Epoch 18/20
100/100 [=====] - 510s 5s/step - loss: 1.7029 - sparse_categorical_accuracy: 0.8098 - val_loss: 3375419621376.0000 - val_sparse_categorical_accuracy: 0.8611
Epoch 19/20
100/100 [=====] - 517s 5s/step - loss: 1.7331 - sparse_categorical_accuracy: 0.8033 - val_loss: 2042125746176.0000 - val_sparse_categorical_accuracy: 0.8423
Epoch 20/20
100/100 [=====] - 536s 5s/step - loss: 1.6725 - sparse_categorical_accuracy: 0.8073 - val_loss: 1760152.3750 - val_sparse_categorical_accuracy: 0.8936
9/100 [=====>] - ETA: 7:08 - loss: 1.5887 - sparse_categorical_accuracy: 0.8368
```

(b)

Fig. 2 Training of 20 epochs (a,b) PointNet for 20 epochs

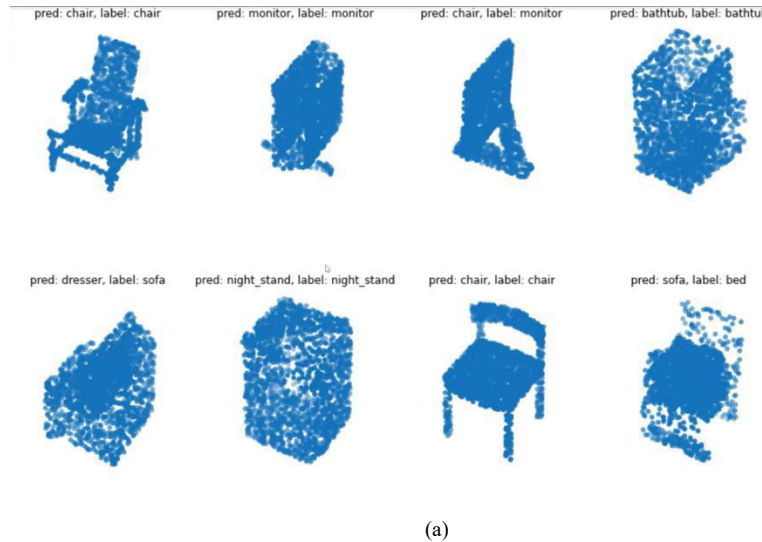


Fig. 3 Predictions after training the dataset (a) Predicted and Labelled

Table: Point Net accuracy Post 20 Epochs		
Model	Training Time (Hours)	Accuracy (%)
PointNet	8	83.05

(a)

Fig. 4 Analytical Table (a)

V. CONCLUSION

The current study highlights the effectiveness of PointNet in handling point sets in 3D spaces for tasks such as 3D classification and segmentation. It has been shown that PointNet is very effective and requires minimal training time. However, the literature survey indicates that there are alternative models with better training time and accuracy available compared to PointNet. As a way forward for future research PointNet++ could be evaluated for further reference.

REFERENCES

- [1] 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 Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017.
- [2] 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. (CVPR), 2016.
- [3] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2017.
- [4] M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature: A quantum mechanical approach to shape analysis," in Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops), pp. 1626–1633, 2011.
- [5] S. M. Shelke, I. S. Pathak, A. P. Sangai, D. V. Lunge, K. A. Shahale, and H. R. Vyawahare, "A Review Paper on Computer Vision," Int. J. Comput. Appl., vol. XX, no. XX, pp. XX–XX, Year.
- [6] D. Belton and D. D. Lichti, "Classification and segmentation of terrestrial laser scanner point clouds using local variance information," Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., vol. XXXVI-5, pp. 44–49, 2006.
- [7] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," arXiv preprint arXiv:1312.6203, 2013.
- [8] B. Singh, "PointNet for 3D Object Classification," [Online]. Available: <https://www.kaggle.com/code/balraj98/pointnet-for-3d-object-classification-pytorch/input> [Accessed: Mar. 20, 2025].

- [9] A. Torralba, P. Isola, and W. T. Freeman, Foundations of Computer Vision, 2024.
- [10] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 1912–1920, 2015.
- [11] C. M. Bishop and H. Bishop, Deep Learning – Foundations and Concepts, Springer Nature, 2023.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details