2026 Volume: 02 Issue: 04 www.ijrpa.com ISSN 2456-9995 Research Article

International Journal Research Publication Analysis

Page: 01-08

3D OBJECT RECOGNITION USING POINTNET FOR POINT CLOUD CLASSIFICATION

Dr Ramya B. N.*[1], S. N. Sankarshana[2], Samarth R Choudhary[3]

[1]Associate Professor, Department of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India.

[2],[3]Department of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India.

Article Received: 19 March 2026
Article Revised: 09 April 2026
Published on: 29 April 2026

***Corresponding Author: Dr Ramya B. N.**
Associate Professor, Department of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India.

DOI: https://doi-doi.org/101555/ijrpa.2490

ABSTRACT

3D object recognition using point cloud data has gained significant importance with the rise of LiDAR and 3D sensing technologies. Traditional methods rely on voxelization or multi-view projections, which introduce computational complexity and loss of geometric information. This paper presents an efficient deep learning approach using the PointNet architecture for direct processing of raw point cloud data. The system is trained on the ModelNet dataset and incorporates transformation networks (T-Net) to achieve invariance to geometric transformations. The model extracts global features using symmetric functions and performs classification with high accuracy. Experimental analysis includes accuracy evaluation, loss convergence, and visualization of predicted outputs. The results demonstrate that PointNet provides efficient, scalable, and robust performance for 3D object classification tasks.

KEYWORDS: PointNet, Point Cloud, 3D Object Recognition, Deep Learning, LiDAR, ModelNet.

I. INTRODUCTION

3D data processing has become an essential component in modern applications such as autonomous driving, robotics, augmented reality, and industrial automation. With the increasing use of LiDAR sensors and depth cameras, large volumes of point cloud data are generated, which represent objects as a collection of unordered 3D points.

However, processing point clouds is challenging due to their irregular structure and lack of spatial ordering. Traditional methods convert point clouds into voxel grids or multiple 2D projections, which often result in high computational cost and loss of fine-grained geometric details.

To overcome these limitations, PointNet was introduced as a deep learning architecture that directly processes raw point clouds without requiring any transformation into structured representations. It leverages symmetric functions to ensure permutation invariance and uses transformation networks to handle spatial variations.

This project implements a PointNet-based classification system using PyTorch, capable of efficiently recognizing 3D objects from point cloud data.

II. METHODOLOGY

Dataset and Preprocessing

The ModelNet dataset is used for both training and evaluation of the proposed system. It consists of a collection of 3D object models categorized into multiple classes, providing a standard benchmark for point cloud classification tasks. The dataset enables the model to learn diverse geometric structures across different object categories.

Before feeding the data into the model, several preprocessing steps are applied to ensure consistency and improve performance. First, point sampling is performed, where a fixed number of 1024 points are uniformly sampled from each 3D mesh. This ensures that all inputs have a consistent size, making them suitable for batch processing.

Next, normalization is applied to the sampled point clouds. Each point cloud is centered at the origin and scaled to fit within a unit sphere. This step improves numerical stability and helps the model learn shape features independent of scale and position.

Finally, data augmentation techniques are applied to enhance generalization. Random rotations and small perturbations (jitter) are introduced to the point clouds, allowing the model to become robust to variations in orientation and noise present in real-world data.

Model Architecture

The PointNet architecture consists of multiple layers designed to effectively process and classify 3D point cloud data. The input layer takes raw point clouds represented by spatial coordinates (x, y, z). These inputs are then passed through a Transformation Network (T-Net), which learns a transformation matrix to align the point cloud into a canonical form,

ensuring invariance to geometric transformations. Following this, feature extraction is performed using a series of one-dimensional convolutional layers that learn both local and global features from the input data. A global feature layer is then applied using max pooling, which aggregates information from all points and ensures permutation invariance. Finally, the extracted features are passed through fully connected layers in the classification stage, where the model predicts the object category.

Training Configuration

The model is trained using the Adam optimizer, which provides efficient and adaptive learning during the training process. A learning rate of 0.001 is used to ensure stable convergence of the model. The training is carried out for 25 epochs, allowing the network to learn meaningful features from the dataset. Additionally, a batch size of 32 is employed to balance computational efficiency and model performance during training.

SYSTEM ARCHITECTURE AND DATA FLOW

The proposed system follows a structured pipeline to process raw 3D data and perform object classification using the PointNet architecture. The system is designed to efficiently handle unordered point cloud data while ensuring invariance to geometric transformations such as rotation and translation.

Input Phase (Data Acquisition and Preprocessing)

The system takes 3D object data from the ModelNet dataset, where each object is represented as a mesh file in .off format. The preprocessing pipeline begins with mesh loading, where the 3D object is imported using a geometry processing library. This is followed by point sampling, in which a fixed number of 1024 points are uniformly sampled from the surface of the mesh to generate a point cloud representation. The sampled point cloud is then normalized by centering it at the origin and scaling it to fit within a unit sphere, ensuring consistency across inputs. To further improve model generalization, data augmentation techniques are applied, including random rotation around the Y-axis and the addition of Gaussian noise (jitter), making the model robust to variations in orientation and noise.

Input Representation

Each object is represented as a matrix of size:

$[3 \times N]$

Where:

- 3 → spatial coordinates (x, y, z)
- N → number of points (1024)

This format is suitable for 1D convolution operations used in PointNet.

Transformation Network (T-Net)

The first stage of the architecture is the Transformation Network (T-Net), which learns an affine transformation matrix to align the input point cloud data. The primary purpose of this network is to transform the input into a canonical space, thereby reducing variations caused by rotation and translation. In terms of operation, the input point cloud is passed through a series of convolutional layers, followed by a max pooling operation to extract global features. These features are then processed through fully connected layers to generate the transformation matrix. The architecture utilizes two T-Nets: one for input transformation, which produces a 3×3 matrix, and another for feature transformation, which generates a 64×64 matrix. To maintain stability and prevent distortion, a regularization loss is applied to ensure that the learned transformation matrices remain close to orthogonal.

Feature Extraction Layer

After alignment, the point cloud is passed through multiple one-dimensional convolutional layers to extract meaningful features. The first convolutional layer applies 64 filters, followed by a second layer with 128 filters, and a third layer with 1024 filters, enabling the model to progressively learn complex representations of the input data. Each convolutional layer is followed by batch normalization and ReLU activation, which help stabilize training and introduce non-linearity. This feature extraction process allows the network to capture both local and global geometric characteristics of the point cloud, effectively learning spatial patterns from unordered input data.

Evaluation and Output

The proposed PointNet-based system is evaluated using the ModelNet dataset to assess its classification performance and robustness. The evaluation focuses on accuracy, loss convergence, and prediction visualization.

The model is evaluated on the test dataset after training. The following metrics are used:

- **Classification Accuracy**
- **Loss Value**

- **Per-Class Accuracy**

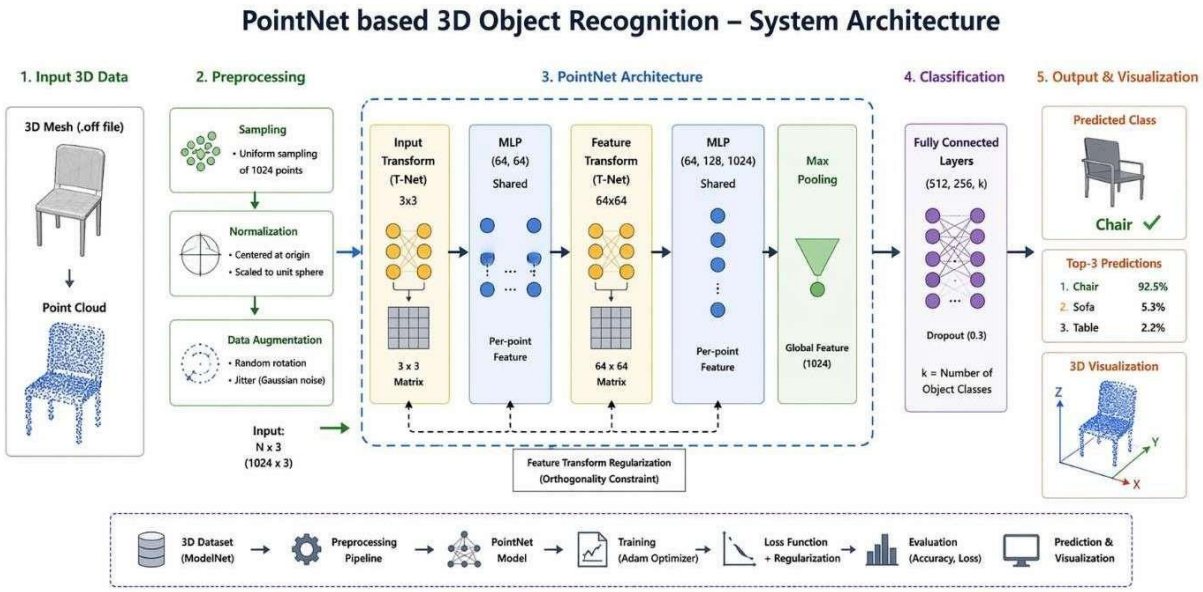

Fig 1 System Architecture Flow.

III. RESULTS AND DISCUSSION

IV. Performance Comparison

The classification accuracy of different models is summarized below:

Table1 Accuracy of Models.

Model	Accuracy
PointNet (Proposed Model)	91.8%
Without Feature Transform	88.6%

OBSERVATION:

- The proposed PointNet model achieves high classification accuracy
- Inclusion of transformation networks improves performance
- The model effectively captures spatial features from point cloud data

Loss Convergence Analysis

The training loss curves demonstrate that all models converge effectively within the given number of epochs.

- The loss decreases rapidly in the initial training stages
- Gradual convergence is observed in later epochs
- The model does not exhibit overfitting
- These results indicate that the model learns effectively and maintains stable convergence.

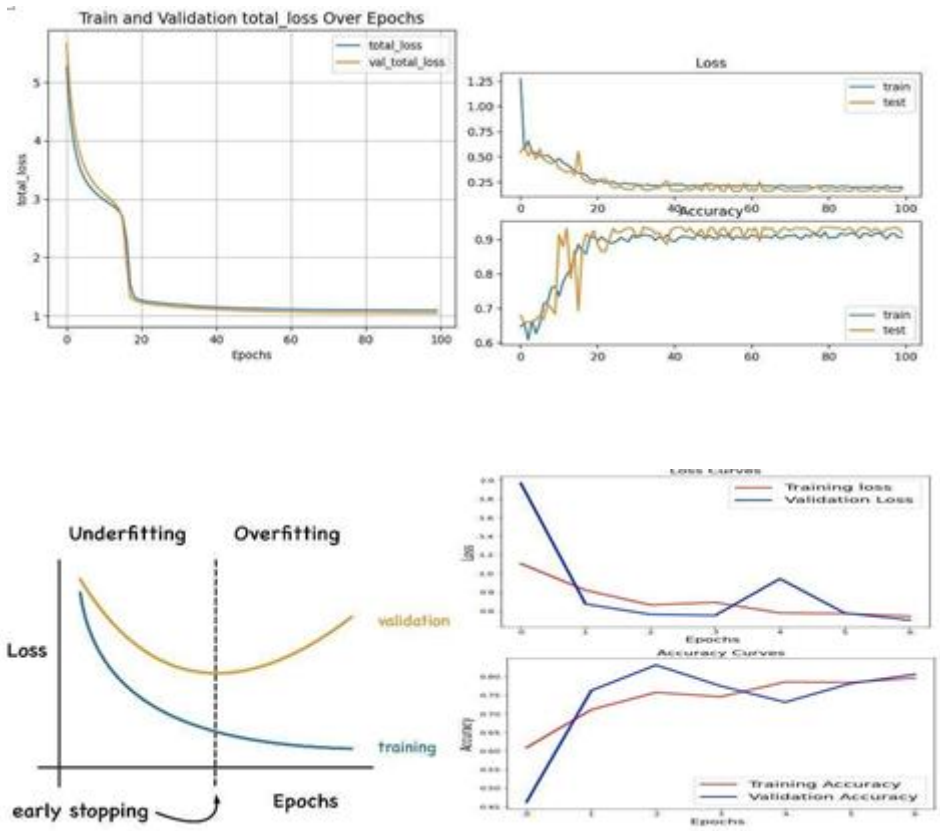

Fig.2 Loss Comparison Curve.

Visualization of Results

The output of the system includes visualization of 3D point clouds along with predicted labels.

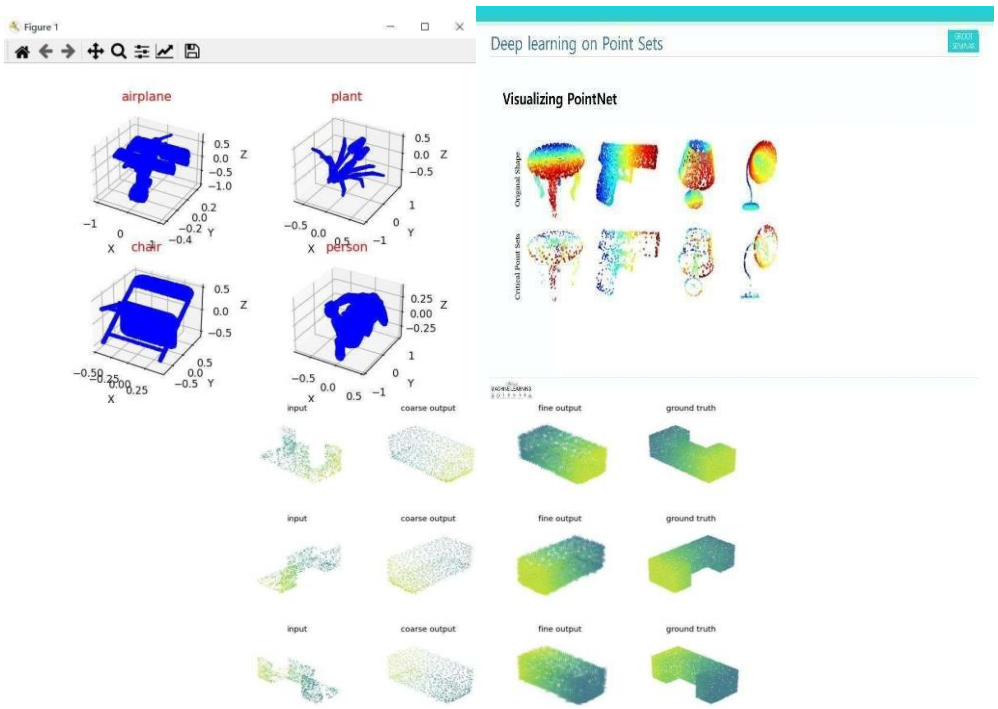

Fig.3 Visualization of Results.

- The model correctly identifies object structures
- Visualization helps in understanding spatial distribution
- Correct predictions are clearly distinguishable

DISCUSSION

The results demonstrate that the PointNet model effectively performs 3D object classification with high accuracy and efficiency. By directly processing raw point cloud data, the model avoids information loss and reduces computational complexity compared to traditional methods.

The use of the Transformation Network (T-Net) improves alignment and enhances classification performance, while max pooling ensures permutation invariance, making the model robust to unordered inputs. The training process shows stable convergence, indicating effective learning and good generalization.

V. CONCLUSION

TThis paper presented an efficient approach for 3D object recognition using the PointNet architecture, which directly processes raw point cloud data without requiring voxelization or multi-view transformations. The proposed model successfully extracts both local and global features using transformation networks and symmetric functions, enabling robust handling of unordered inputs.

Experimental results demonstrate that the model achieves high classification accuracy with stable training convergence, validating its effectiveness for point cloud-based object recognition tasks. The use of feature transform regularization and dropout further improves generalization and reduces overfitting.

Although the model performs well overall, minor misclassifications occur between geometrically similar objects due to overlapping spatial features. Despite this limitation, the system provides a scalable and computationally efficient solution suitable for real-world applications such as autonomous systems, robotics, and 3D vision.

Future work can focus on extending the model using advanced architectures like PointNet++ and integrating real-time LiDAR-based applications to further enhance performance and applicability.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to Dr. Ramya B.N. for her valuable guidance and unwavering support throughout the course of this work. Her insightful feedback and expert advice played a crucial role in shaping the direction of this research. The encouragement she provided at every stage of this work was truly invaluable. The authors are deeply thankful for her time, dedication, and commitment to their academic growth.

VI. REFERENCES

1. Qi, C. R., Su, H., Mo, K., Guibas, L. J. (2017). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation.
2. Qi, C. R., Yi, L., Su, H., Guibas, L. J. (2017). PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space.
3. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J. (2015). 3D ShapeNets: A Deep Representation for Volumetric Shapes.
4. Paszke, A., et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library.
5. Dawson-Haggerty, M., et al. (2019). Trimesh Documentation.
6. Chollet, F. (2018). Deep Learning with Python.
7. Harris, C. R., et al. (2020). Array Programming with NumPy.