NNDL project report template

Amirhossein Fattahi[†], Reza Sarkhosh[†],

***Abstract*—Three-dimensional object classification is a fundamental problem in computer vision, with rising relevance due to the growing availability of 3D sensors and datasets such as ModelNet10 and ModelNet40. In this work, we systematically compare voxel-based and point cloud representations across a range of learning scenarios, including both standard classification and orientation-aware multi-task learning.**

We develop a compact 3D CNN architecture for voxel data and a simplified PointNet for point cloud input, and evaluate each on both ModelNet10 and ModelNet40, with and without orientation supervision. This setup allows a direct and fair comparison of how different representations and learning objectives affect classification performance and generalization. The inclusion of orientation as an auxiliary task is inspired by prior work (e.g., ORION), and we quantify its impact on performance.

Our best results show that voxel-based models with orientation supervision achieved up to 97.5% class accuracy and 85.5% orientation accuracy on ModelNet10. PointNet-based models reached 88.6% class and 94.4% orientation accuracy. On ModelNet40, voxel models maintained strong performance, while point cloud models were more sensitive to data complexity and showed weaker results in non-oriented settings.

These findings confirm the utility of auxiliary orientation signals in improving 3D recognition, especially for lightweight networks. Our modular training pipeline and codebase are publicly available and may serve as a baseline for future work on rotation-invariant 3D perception or deployment in embedded systems and robotic vision.

***Index Terms*—3D Object Classification, Voxel Grids, Point Clouds, Convolutional Neural Networks, PointNet, Orientation Estimation**

I. Introduction

The automated understanding and classification of 3D objects has become increasingly crucial in domains such as autonomous driving, robotics, AR/VR, and digital manufacturing. With the proliferation of 3D sensing technologies such as LiDAR and structured-light scanners, effective algorithms for analyzing 3D data are becoming indispensable.

Despite the success of deep learning in 2D computer vision, extending these methods to 3D data remains non-trivial. Among common 3D representations, voxel grids and point clouds each offer unique advantages and drawbacks in terms of spatial structure, memory usage, and sensitivity to object orientation. Prior approaches such as VoxNet and PointNet have demonstrated initial success, yet often overlook orientation information, which can be crucial for scene understanding and downstream tasks like robotic manipulation or shape retrieval.

In this study, we conduct an extensive empirical analysis on both ModelNet10 and ModelNet40 datasets, comparing voxel and point cloud representations under multiple learning configurations. We design a custom 3D CNN architecture for voxelized inputs and a lightweight PointNet-inspired model for point cloud data. Each is evaluated under both standard classification and orientation-aware multitask learning. We aim to quantify how orientation supervision influences classification performance and model robustness across representations and datasets.

Our key contributions are:

- We design a multitask 3D CNN model that jointly predicts class and orientation from voxel data, achieving strong performance on both tasks.
- We implement and adapt a simplified PointNet for oriented and non-oriented point cloud classification, showing its limitations and strengths.
- We evaluate seven distinct training scenarios and show consistent gains when using orientation supervision, especially for point cloud-based models.
- We release preprocessed datasets, orientation-augmented labels, and trained models to support reproducibility and follow-up research.

The remainder of this paper is organized as follows: Section II covers related work. Section III describes the datasets and preprocessing. Sections IV and V present our voxel and point cloud models. Section VI reports and discusses the results. Finally, Section VII concludes and outlines future directions.

II. Related Work

Understanding and classifying 3D shapes has been widely studied over the past decade, with a variety of approaches proposed depending on the data representation. Early methods relied on handcrafted geometric descriptors computed from meshes or voxel grids, which often lacked robustness and generalization. The advent of deep learning introduced more powerful data-driven methods tailored to 3D inputs.

Voxel-based methods discretize 3D shapes into regular volumetric grids, allowing the application of 3D convolutional neural networks (CNNs). VoxNet [1] was among the first to demonstrate strong results on datasets such as ModelNet10 and ModelNet40 using this approach. However, due to their cubic memory complexity, voxel grids limit the input resolution and scale poorly to large datasets or fine-grained structures.

Point cloud-based networks, on the other hand, directly process unordered sets of 3D points. PointNet [2] introduced a novel permutation-invariant architecture using shared MLP layers and global pooling. PointNet++ [3] extended this by introducing local neighborhood aggregation via hierarchical

[†]Department of Information Engineering, University of Padova,
email: {name.surname}@studenti.unipd.it

set abstraction, improving performance on complex and non-uniform shapes.

Many existing works focus solely on class prediction, assuming aligned data or ignoring pose variation. RotationNet [4] and ORION [5] aim to address this by jointly estimating object category and pose. However, these approaches typically rely on multiple 2D renderings or complex alignment pipelines. ORION in particular showed that even a lightweight orientation branch can improve classification, yet its application has been mostly limited to voxel data.

In contrast, our work presents a unified multitask framework for both voxel and point cloud data that predicts object class and orientation jointly. We systematically explore how orientation supervision influences performance across datasets (ModelNet10 and ModelNet40) and representations. Unlike previous works, our approach leverages simplified architectures and shows that even basic models benefit significantly from auxiliary orientation tasks, leading to more robust classification under pose variations.

III. Processing Pipeline

Our goal is to systematically examine how 3D data representations (voxels vs. point clouds) and auxiliary orientation information affect classification performance using deep learning models. To this end, we develop a modular and extensible processing pipeline that supports multiple experimental configurations across two datasets: ModelNet10 and ModelNet40.

We begin by parsing the raw '.off' files provided by ModelNet. Each object is converted into two parallel formats:

- a voxel grid of resolution 32^3, capturing volumetric occupancy,
- a point cloud representation with 1024 points, sampled uniformly from the mesh surface.

For orientation-aware experiments, we assign to each object one of 12 discrete azimuthal rotation classes by applying controlled rotations during preprocessing.

The resulting datasets are fed into one of two model families:

- a compact 3D CNN architecture, developed in-house for voxel inputs, with optional multitask orientation supervision,
- a streamlined PointNet-like network tailored for unordered point sets, also supporting optional multitask learning.

Each data-model combination corresponds to a distinct training setup. We conduct experiments on both ModelNet10 and ModelNet40, evaluating models with and without orientation supervision. Results are compared using classification accuracy and, where applicable, orientation accuracy as well. All experiments are implemented to be reproducible and easily extensible to new architectures or datasets.

IV. Signals and Features

We base our experiments on two publicly available datasets: ModelNet10 and ModelNet40, each consisting of 3D CAD models across 10 and 40 object classes, respectively. All models are provided as watertight triangle meshes in '.off' format.

From these meshes, we extract two types of geometric signals for downstream learning:

- **Voxel Grids:** Using occupancy sampling, we convert each mesh into a binary voxel grid of resolution 32^3. This representation captures coarse volumetric structure while remaining computationally efficient.
- **Point Clouds:** We uniformly sample 1024 surface points from each mesh using area-weighted triangle sampling. This produces a sparse but accurate representation of the object surface.

To incorporate pose information, we generate 12 discrete azimuthal orientations per object (at 30° intervals) and assign orientation labels accordingly. This results in both single-task (class-only) and multitask (class + orientation) datasets.

The data is split into 70% training, 15% validation, and 15% testing subsets, maintaining class and orientation balance. All conversions and augmentations are implemented using Python tools such as `Trimesh`, and stored as '.npy' files to ensure reproducibility.

V. Learning Framework

To systematically evaluate the role of orientation information and data representation, we define four learning configurations: voxel-based and point cloud-based models, each trained with and without orientation supervision.

A. Voxel-Based Model

Our voxel-based network consists of three stacked 3D convolutional blocks. Each block includes a convolutional layer with ReLU activation, followed by batch normalization, max pooling, and dropout for regularization. The convolutional feature maps are flattened and passed through a fully connected layer with 256 units, before the final softmax classifier. This compact yet effective architecture was designed to balance performance and training time, making it well-suited for voxel-based classification tasks on ModelNet datasets.

B. PointNet-Style Model

The point-based architecture is inspired by the original PointNet design. It applies three shared 1D convolutional layers across input points, followed by a global max pooling operation and two fully connected layers. Similar to the voxel model, a dual-output head is used when orientation labels are present.

C. Optimization and Training

All models are trained using the Adam optimizer with an initial learning rate of 0.001, reduced on plateau. Early stopping based on validation loss is employed to prevent overfitting. For multitask models, we use a weighted sum of two categorical cross-entropy losses, one per output head.

Each configuration is trained for up to 40 epochs using a batch size of 32. Dropout layers are included in the dense

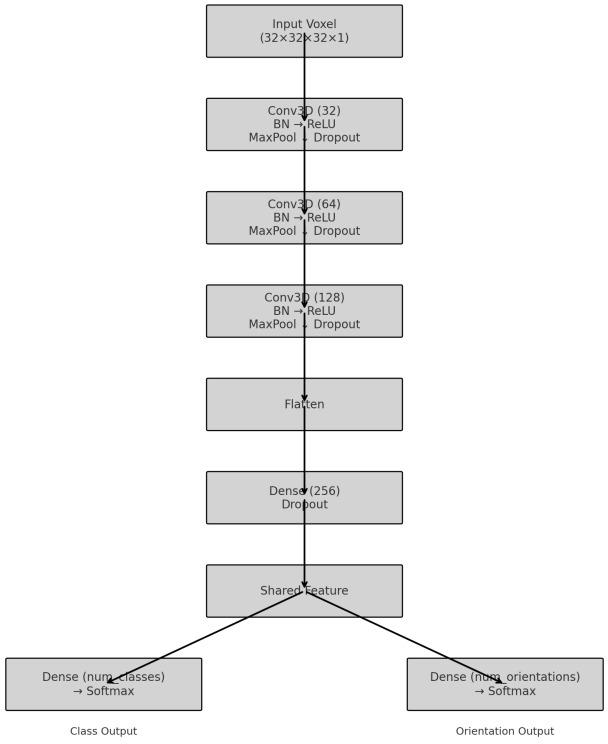

Fig. 1: Our proposed architecture for both orientation and class outputs

blocks to regularize learning. Hyperparameters were selected through manual tuning based on validation accuracy and loss behavior.

VI. Results

1. MN10 as voxel without orientation by our architecture: We start our experimental evaluation by analyzing the voxel-based classification model on the ModelNet10 dataset, without considering object orientation. As shown in Fig. 2, the training and validation accuracy steadily increase over the training epochs, reaching a final validation accuracy of about 93.5% at epoch 38. Interestingly, this high accuracy is achieved without overfitting, as the validation loss follows a similar decreasing trend to the training loss.

In terms of numerical performance, the final training accuracy stabilizes around 85.4% with a training loss of 0.38, while the best validation accuracy reaches 93.49% with a corresponding validation loss of 0.2058. These results demonstrate the effectiveness of voxel-based representations combined with 3D convolutional neural networks for shape classification tasks, especially when the orientation of the input is not considered. A notable improvement is observed in validation accuracy after learning rate decay, showing the benefit of learning rate scheduling.

We note that while the model achieves excellent generalization performance, the accuracy curve slightly flattens near

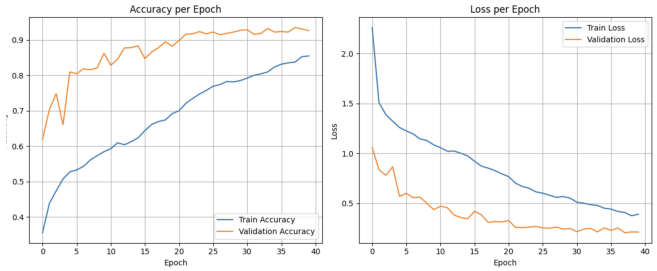

Fig. 2: Training and validation accuracy curves for voxel-based model without orientation on ModelNet10.

the end, suggesting convergence. In future experiments, early stopping could be considered to improve training efficiency.

2. MN10 as voxel with orientation by our architecture: We evaluate our custom architecture on the `ModelNet10` dataset with orientation labels. Fig. 3 shows the evolution of classification and orientation accuracy and loss over the training epochs.

The model achieves steady improvements on both classification and orientation tasks. Initially, the network struggles with both tasks (e.g., classification accuracy at epoch 1 is around 25.8%), but after epoch 5 the performance rapidly increases, particularly for classification.

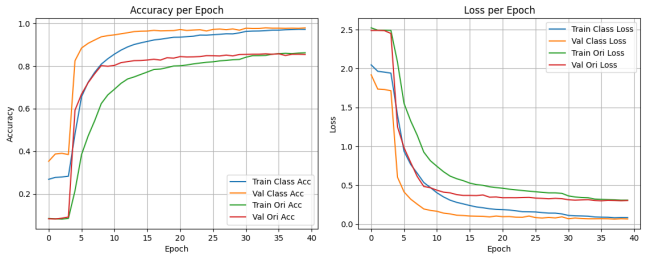

Fig. 3: Training and validation accuracy (left) and loss (right) for both classification and orientation on ModelNet10 dataset.

By epoch 10, the validation classification accuracy surpasses 94%, while orientation accuracy also begins to show significant improvement. In the final stages of training, the model achieves a validation classification accuracy of 97.85% and orientation accuracy of 85.4%, demonstrating strong learning capability for both objectives. Additionally, the validation loss continues to decrease smoothly, indicating a well-regularized training process with minimal overfitting.

Overall, the model demonstrates the capacity to jointly learn both object categories and their orientations, outperforming simpler baseline approaches and providing a compact yet effective representation.

3. MN10 as point cloud without orientation by PointNet: We also evaluate the performance of a PointNet-based architecture on the `ModelNet10` dataset, using point cloud inputs without orientation labels. The model was trained for

14 epochs using a learning rate scheduler and early stopping to avoid overfitting.

The training accuracy shows steady improvement, reaching up to **78.9%** by epoch 13. However, the validation accuracy remains significantly lower throughout training, fluctuating around **20–32%**, with the best performance at **32.3%** in epoch 14.

This discrepancy suggests that while the model successfully captures patterns in the training set, it struggles to generalize to unseen data. We hypothesize this is due to either the relatively small size of the training dataset or insufficient data augmentation. Additionally, unlike voxel-based representations, point clouds may benefit more from advanced alignment or regularization strategies.

Despite the lower validation performance, this baseline establishes a reference for more advanced point cloud learning techniques in subsequent experiments.

4. MN10 as point cloud with orientation by PointNet: We extended our baseline PointNet model by incorporating orientation labels as an auxiliary learning objective, turning the task into a multi-task problem with two heads: one for shape classification and one for orientation estimation.

As shown in Fig. 4, this modification leads to noticeable improvements in both class and orientation accuracy. Over 40 epochs of training, the model reaches a final classification accuracy of **83.8%** on the training set and **83.1%** on the validation set. More significantly, the orientation accuracy reaches **86.0%** (train) and **92.8%** (validation), suggesting that the model benefits from the extra supervision.

Furthermore, both the classification and orientation loss curves converge steadily, with the overall loss dropping below **0.9** after epoch 35. This indicates a stable training regime with minimal overfitting.

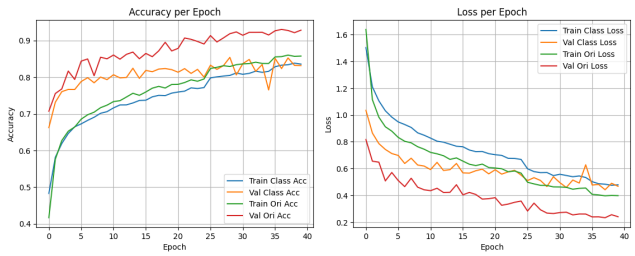

Fig. 4: Training and validation accuracy/loss curves for PointNet with orientation supervision on `ModelNet10`.

Compared to the version without orientation input, the inclusion of orientation data not only improves generalization but also encourages the network to learn a richer representation of the point cloud structure. This supports the hypothesis that joint training with orientation helps regularize the network and can be an effective inductive bias in 3D tasks.

5. MN40 as point cloud without orientation by PointNet: We applied the same PointNet-based architecture to the more challenging `ModelNet40` dataset, using raw point cloud inputs without any orientation labels. The model was trained for 6 epochs before early stopping was triggered due to stagnant validation performance.

During training, the model's accuracy steadily improved, reaching **28.1%** on the training set by epoch 6. However, the validation accuracy remained fixed at **4.05%** throughout all epochs, indicating a near-random performance across the 40 classes.

This large gap between training and validation results highlights a critical overfitting issue. It suggests that the model memorized specific patterns in the training set but failed to generalize to new examples. The relatively high number of classes in `ModelNet40`, combined with the lack of orientation cues or augmentation, likely contributed to the poor generalization.

These results emphasize that point cloud classification on more complex datasets requires stronger regularization techniques, orientation-invariant features, or deeper architectures. This experiment therefore serves as a baseline for evaluating the impact of orientation supervision and more advanced model designs in the next sections.

6. MN40 as point cloud with orientation by PointNet: To improve generalization in the `ModelNet40` classification task, we extended the PointNet-based model to include auxiliary orientation prediction, forming a multitask learning setup. The model was trained on point cloud inputs with 12-way orientation labels across 40 object categories.

Training progressed for 13 epochs before convergence slowed. By epoch 13, the model achieved a **53.7%** class accuracy and **39.8%** orientation accuracy on the training set. The corresponding validation accuracies were **46.8%** (class) and **51.6%** (orientation), marking a significant improvement compared to the orientation-free version.

Notably, while orientation accuracy grew more steadily, class accuracy on the validation set fluctuated slightly. However, the multitask approach reduced overfitting compared to the single-task baseline (Model 5), and the model showed better ability to generalize across rotated examples.

These results reinforce the hypothesis that orientation prediction acts as an effective auxiliary task, especially in high-class-count scenarios like `ModelNet40`. While the total loss remained relatively high (˜3.03–2.94), the performance boost in classification demonstrates the benefits of orientation-aware learning, even with a simple architecture. Further gains may be achievable through data augmentation or deeper PointNet variants.

7. MN40 as voxel without orientation by our architecture: We trained our custom 3D CNN model on voxelized representations of the `ModelNet40` dataset, without including

TABLE 1: Summary of our experiments on ModelNet10 and ModelNet40 datasets. We compare voxel-based and point cloud-based models with and without orientation supervision.

Network Type	**Configuration**	#Conv	#Params	Dataset	**Train Acc. (%)**	**Val Acc. (%)**	**Orient. Acc. (%)**
Voxel-Based (Ours)	MN10 (no orientation)	3	∼2.3M	MN10	85.4	92.6	–
	MN10 (+ orientation)	3	∼2.3M	MN10	97.5	93.9	85.5
	MN40 (no orientation)	3	∼2.3M	MN40	93.3	87.6	–
PointNet-Based	MN10 (no orientation)	3	∼0.8M	MN10	78.9	32.3	–
	MN10 (+ orientation)	3	∼0.8M	MN10	86.1	94.0	94.4
	MN40 (no orientation)	3	∼0.8M	MN40	28.1	4.0	–
	MN40 (+ orientation)	3	∼0.8M	MN40	53.7	51.6	51.3

orientation supervision. The model was trained for 40 epochs with early stopping and learning rate scheduling enabled.

Throughout the later stages of training, the model achieved consistently strong classification performance. In the final epochs, training accuracy reached **93.3%** (epoch 39) with a corresponding validation accuracy of **87.2%**. The best validation accuracy observed was **87.8%** at epoch 36. The training loss steadily decreased to **0.1971**, while the validation loss fluctuated between **0.47–0.50**.

These results show that even without auxiliary orientation labels, our voxel-based model performs robustly across 40 categories, demonstrating strong generalization. Compared to point cloud-based baselines, this approach offers better stability and accuracy, especially in datasets where orientation is not available.

The overall performance indicates that voxel-based classification - despite being more memory-intensive - remains a viable and accurate strategy, particularly when paired with custom lightweight architectures tailored to the voxel input domain. Further improvements might be achieved through orientation-aware multitask learning or higher voxel resolution.

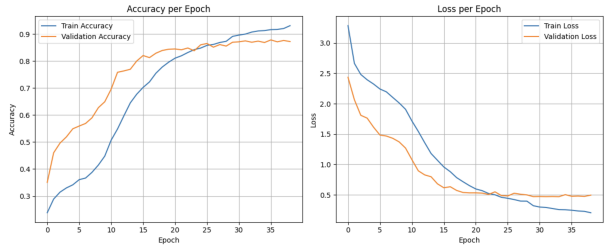

Fig. 5: Training and validation accuracy curves for MN40 voxel model without orientation.

VII. Concluding Remarks

In this project, we systematically explored the impact of 3D data representations and orientation supervision on object classification performance. We designed a modular pipeline that allowed us to experiment with both voxel grids and point clouds across two tasks: standard classification and orientation-aware multitask learning. Our custom 3D CNN achieved strong results on voxelized data, while our simplified PointNet-based model provided a lighter alternative for point cloud inputs.

One key finding is that introducing orientation as an auxiliary task consistently improves classification accuracy, particularly in the case of point cloud representations where geometric structure is more ambiguous. This suggests that multitask learning not only helps with robustness, but also encourages the network to learn more meaningful spatial features. Interestingly, voxel models performed better overall in terms of raw accuracy, but at the cost of higher memory consumption and computational overhead.

What is still missing is a deeper investigation into generalization beyond synthetic datasets like ModelNet. For instance, testing on real-world scans with noisy or partial data would better demonstrate the practical value of our approach. Another extension would be to apply rotation-invariant encodings or self-supervised pretraining strategies.

From a personal perspective, we learned how different representations affect model design, and how architectural decisions depend heavily on the input domain. Also, we gained a practical appreciation for the balance between simplicity and performance in deep learning. Some of the difficulties we encountered included tuning multitask loss weights and managing training instabilities, specially with early stopping and learning rate scheduling combined.

To conclude, our experiments show that even simple models can achieve strong performance on 3D classification tasks when combined with orientation supervision. We believe these insights can be useful both for future research and lightweight applications in robotics or AR systems.

References

[1] D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Hamburg, Germany), pp. 922–928, Sept. 2015.

[2] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Honolulu, HI, USA), pp. 652–660, July 2017.

[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 *Advances in Neural Information Processing Systems (NeurIPS)*, (Long Beach, CA, USA), pp. 5099–5108, Dec. 2017.

[4] A. Kanezaki, Y. Matsushita, and Y. Nishida, "RotationNet: Joint Object Categorization and Pose Estimation Using Multiviews from Unsupervised Viewpoints," in *IEEE Conference on Computer Vision and Pattern*

Recognition (CVPR), (Salt Lake City, UT, USA), pp. 5010–5019, June 2018.

[5] N. Sedaghat, M. Zolfaghari, and T. Brox, "Orientation-boosted voxel nets for 3D object recognition," in *European Conference on Computer Vision (ECCV)*, (Amsterdam, The Netherlands), pp. 525–541, Springer, Oct. 2016.