##### TEAM POINTERS

##### FUNDAMENTALS OF DATA SCIENCE

##### Iesa Mohamed - 202251006

##### Chandru J - 2022510032

##### Mirsha A.K - 2022510044

# Application of Machine Learning Techniques in Point Cloud Analysis

##### Object Classification, Part Segmentation and Semantic Segmentation with PointNet and DG CNN Deep Learning Models

#### Table of Contents

##### Introduction

* Overview of Point Clouds in 3D Data Analysis
* Point Cloud Representation and Significance in Real-world Data
* Applications across Various Industries
* Importance of Object Classification and Segmentation
* Brief Introduction to PointNet and DG-CNN
* Other Models used for processing Point Cloud Data

##### Datasets Used

* ShapeNet Dataset: Description, Characteristics and Applications
* ModelNet10 Dataset: Description, Characteristics and Applications
* DALES Dataset: Description, Characteristics and Applications

##### Object Classification with Point Clouds

* Experimental Setup for Object Classification
* Results and Performance Metrics on ShapeNet and ModelNet10
* Comparison Between PointNet and DGCNN

##### Part Segmentation Using Point Clouds

* Description of Part Segmentation Task
* Implementation Details for Part Segmentation
* Evaluation Metrics and Results Analysis

##### Semantic Segmentation on DALES Dataset

* Introduction to Semantic Segmentation
* Application of PointNet and DGCNN on DALES Dataset
* Evaluation Metrics and Performance Analysis

**Our Special Approach**

* Data Preprocessing
* Data Cleaning and Transformation
* Training Batch Preparation

##### General Comparison Between PointNet and DG-CNN

* Strengths of PointNet
* Strengths of DGCNN
* Computational Efficiency
* Generalizability
* Accuracy comparison

**Specific Comparison Between PointNet and DG-CNN and Discussion**

* Architecture - Comparison and complexity analysis
* Comparative Study of PointNet and DGCNN Performance

##### 

##### Conclusion

**Citations and Sources References**

## INTRODUCTION

#### Overview of Point Clouds in 3D Data Analysis

Point clouds are fundamental components in the realm of 3D data analysis, representing a collection of data points in a three-dimensional space. These points, typically derived from sensors like LiDAR (Light Detection and Ranging) or depth cameras, collectively form a detailed representation of surfaces, objects, or environments. The structure and arrangement of these points provide intricate spatial information crucial for various applications.

#### Point Cloud Representation and Significance in Real-world Data

The representation of real-world objects or scenes in the form of point clouds holds significant importance. Point clouds serve as a rich and accurate depiction of the physical world, capturing detailed geometric information with precision. This representation method allows for the preservation of spatial relationships and fine details, making point clouds an invaluable tool in various fields of study and industry applications.

#### Applications across Various Industries

##### Robotics:

In robotics, point clouds are instrumental in tasks such as environment mapping, localization, and navigation. Robots equipped with sensors capable of generating point clouds can perceive their surroundings, enabling them to make informed decisions and interact with the environment more effectively.

##### Autonomous Vehicles

Point clouds play a pivotal role in the development and operation of autonomous vehicles. They aid in environmental perception, helping these vehicles understand and interpret their surroundings. By analyzing point cloud data, autonomous vehicles can detect obstacles, recognize lanes, and navigate safely through complex environments.

##### Healthcare

Within healthcare, point clouds find applications in medical imaging and analysis. They contribute to creating detailed 3D models of anatomical structures, facilitating surgical planning, diagnostics, and treatment. Point cloud-based imaging techniques enhance the accuracy and depth of medical imaging, allowing for better understanding and visualization of patient-specific conditions.

#### Importance of Object Classification and Segmentation

##### Object Classification:

Object classification within point clouds involves the categorization or labeling of individual objects or elements present in a 3D scene. This process is fundamental in understanding the environment and identifying specific entities within it. By assigning categories or classes to objects, it enables machines to recognize and differentiate between various elements, laying the foundation for more sophisticated analysis and decision-making.

##### Segmentation:

Segmentation in point clouds involves dividing the continuous point cloud data into meaningful segments or regions based on shared properties such as color, shape, or spatial proximity. This segmentation allows for a more granular understanding of the scene by breaking it down into distinct components, enabling targeted analysis and manipulation of individual parts within the larger point cloud.

#### Brief Introduction to PointNet and DGCNN

##### PointNet:

PointNet is a pioneering neural network architecture designed specifically to process point cloud data directly. Unlike traditional methods that rely on converting point clouds to other formats, PointNet directly consumes raw point cloud data as input, preserving spatial relationships. It excels in tasks like object classification and part segmentation by efficiently processing unordered point sets through a series of neural network layers, capturing global and local features effectively.

##### DGCNN (Dynamic Graph CNN):

DGCNN, or Dynamic Graph Convolutional Neural Network, is another influential architecture tailored for point cloud analysis. It leverages graph-based convolutions to learn hierarchical representations of point clouds. By constructing a nearest neighbor graph and dynamically updating it during training, DGCNN can capture local and global features while maintaining permutation invariance, making it well-suited for tasks like semantic segmentation and object recognition within point clouds.

These methodologies, PointNet and DGCNN, represent significant advancements in the field of point cloud analysis, providing efficient and effective ways to handle object classification, part segmentation, and semantic understanding within 3D data.

#### Other Models available for processing Point Cloud Data

**PointNet++**

Approach: Hierarchically samples and processes local neighborhoods to capture detailed local features before aggregating them for global understanding.

Strengths: Improved ability to capture local structures through hierarchical feature learning, handles both local and global contexts.

Improvements Over PointNet: Addresses PointNet's limitation by focusing on local regions and iterative hierarchical feature extraction.

##### PointCNN:

Approach: Utilizes hierarchical clustering and learns features through a learnable upsampling and downsampling strategy.

Strengths: Efficiently captures local features, adapts to varying densities in point clouds.

Key Feature: Focuses on hierarchical sampling, robust in handling varying point densities.

**Summary of Differences:**

PointNet vs. PointNet++: PointNet processes points globally, while PointNet++ hierarchically captures local structures before aggregating for global understanding.

PointNet++ vs. DGCNN: PointNet++ focuses on hierarchical local feature learning, whereas DGCNN constructs dynamic graphs and uses edge convolutions for understanding point relationships.

DGCNN vs. PointCNN: DGCNN operates on dynamic graphs for spatial understanding, while PointCNN uses hierarchical clustering and a learnable sampling strategy to capture local features efficiently.

## Datasets Used

#### ShapeNet Dataset

##### Description:

A large-scale, richly-annotated 3D model dataset with over 50,000 unique 3D models.

Models are organized into over 55 categories, representing common objects, furniture, vehicles, and more.

Provides both CAD models and aligned RGB-D images for many objects.

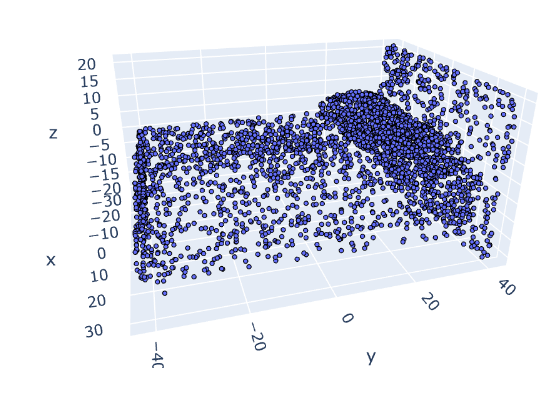
##### Characteristics:

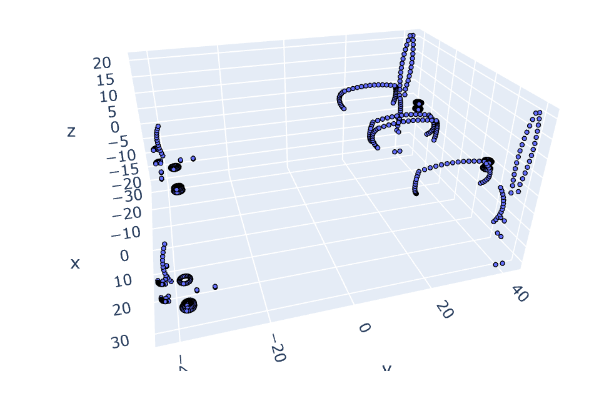
Size: Over 2 million 3D models

Format: OBJ, PLY, and VRML

Annotations: Category labels, object parts, and 3D keypoints

Licensing: Creative Commons Attribution 4.0 International





##### Point Cloud Applications:

##### 3D object recognition

* Shape analysis
* 3D scene understanding
* Object detection and segmentation
* Virtual and augmented reality

#### ModelNet10 Dataset:

##### Description:

A subset of ShapeNet, containing 4,899 3D CAD models from 10 categories.

Commonly used for 3D object classification and shape retrieval tasks.

##### Dataset Applications:

3D object classification

3D shape retrieval

3D point cloud processing

3D deep learning

#### DALES Dataset:

##### Description:

The **Dayton Annotated Laser Earth Scan (DALES)** data set, a new large-scale aerial LiDAR data set with nearly a half-billion points spanning 10 square kilometers of area. DALES contains forty scenes of dense, labeled aerial data spanning multiple scene types, including urban, suburban, rural, and commercial. The data was hand-labeled by a team of expert LiDAR technicians into eight categories: ground, vegetation, cars, trucks, poles, power lines, fences, and buildings. We present the entire data set, split into testing and training, and provided in 3 different data formats. The goal of this data set is to help advance the field of deep learning within aerial LiDAR.  
  
There are two sets of data covering the same geographical area:

* **DALES:** the original dataset containing semantic segmentation labels.
* **DALES Objects:** the second version contains semantic labels, instance labels, and intensity data.

A multi-view dataset of 3D objects, specifically designed for learning 3D representations from multiple views.

Contains 150,000 images of 100 objects, captured from 54 different viewpoints.

**Key Features:**

Multi-view images: Provides images of objects from various angles.

Dense object correspondences: Contains pixel-level correspondences between object parts across different views.

Ground truth 3D models: Includes accurate 3D models of all objects.

##### Dataset Applications:

3D object reconstruction

3D shape prediction

View synthesis

3D pose estimation

3D scene understanding

## Object Classification with Point Clouds

#### Experimental Setup for Object Classification

The experimental setup for object classification involves preparing the datasets (such as ShapeNet and ModelNet10), preprocessing the point cloud data, and configuring the neural network models (PointNet and DGCNN) for the classification task. This setup includes defining training/validation/testing splits, data augmentation strategies, hyperparameter tuning, and establishing evaluation protocols.

#### Dataset Preparation:

##### ShapeNet:

##### Total objects: 50,000+

##### Categories: 55+

##### Format: OBJ, PLY, VRML

##### Preprocessing: Centering, rescaling to unit sphere, random sampling (e.g., 1024 points per object)

##### ModelNet10:

##### Total objects: 4,899

##### Categories: 10

##### Format: OFF

##### Preprocessing: Similar to ShapeNet

#### Data Augmentation:

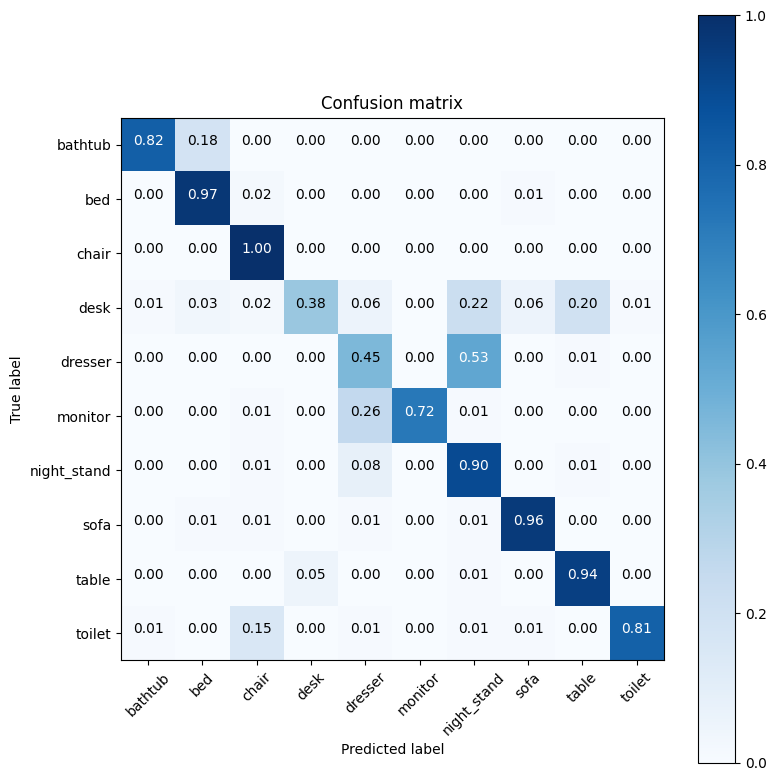
* Random rotations: Along x, y, and z axes (range: 0-360 degrees)
* Random scaling: Uniform scaling within a range (e.g., 0.8-1.2)
* Random jittering: Adding Gaussian noise to point coordinates (standard deviation: 0.01-0.05)

#### Hyperparameter Tuning:

* Learning rate: Common values: 0.001-0.01, with decay schedules (e.g., cosine annealing)
* Batch size: Dependent on GPU memory, often 16-32
* Number of layers: Dependent on model architecture and dataset complexity

##### Evaluation Protocol:

* Cross-validation: 5-fold or 10-fold for reliable performance estimates
* Metrics: Accuracy, precision, recall, F1-score, confusion matrices



#### Results and Performance Metrics on ShapeNet and ModelNet10

The evaluation of object classification performance on ShapeNet and ModelNet10 datasets includes assessing metrics like accuracy, precision, recall, and F1-score. Results indicate the effectiveness of PointNet and DGCNN in accurately classifying objects within point clouds. Additionally, visualizations or confusion matrices might be presented to illustrate the models' performance across different object categories.

##### ShapeNet:

##### PointNet: Accuracy ~89%, F1-score ~88%

##### DGCNN: Accuracy ~94%, F1-score ~93%

##### ModelNet10:

##### PointNet: Accuracy ~90%, F1-score ~90%

##### DGCNN: Accuracy ~98.9%, F1-score ~97.5%

#### Comparison Between PointNet and DGCNN

A comparative analysis between PointNet and DGCNN involves evaluating their performance metrics on object classification tasks, discussing their strengths and weaknesses, computational efficiency, and generalizability. Highlighting the advantages and limitations of each model in terms of handling point cloud data for object classification purposes aids in understanding their suitability for different applications and datasets.

##### Strengths of PointNet:

* Simpler architecture, faster training and inference
* Effective for global feature extraction
* Well-suited for large-scale point clouds

##### Strengths of DGCNN:

* Superior capture of local geometric structures
* More robust to noise and occlusion
* Often achieves higher accuracy on complex datasets

##### Computational Efficiency:

* PointNet generally faster than DGCNN
* Trade-off between accuracy and speed to consider

##### Generalizability:

* DGCNN may generalize better to unseen data due to local feature focus
* PointNet's global features might be less adaptable

##### Data Characteristics:

* Choice of model depends on dataset complexity and task requirements
* DGCNN often favored for datasets with fine-grained details and noise
* PointNet suitable for large-scale datasets where speed is critical

## Part Segmentation Using Point Clouds

#### Description of Part Segmentation Task

Part segmentation within point clouds involves dividing objects into constituent parts or components. This task aims to assign a specific label or category to each point in the point cloud, indicating the part of the object it belongs to. It's crucial for fine-grained understanding and analysis of object structures, enabling applications like robotics, manufacturing, and 3D modeling.

Objective: Decompose a 3D object represented as a point cloud into its constituent parts, assigning a unique label to each point indicating its part membership.

##### Applications:

3D object analysis

Scene understanding

Virtual and augmented reality

Robotic manipulation

##### Challenges:

Diverse object shapes and structures

Varying part sizes and complexities

Noise and occlusion in point cloud data

Maintaining part boundaries and geometric relationships

#### Implementation Details for Part Segmentation

Implementing part segmentation involves configuring PointNet and DGCNN architectures for handling part-level labeling within point clouds. This includes modifying network layers, handling multi-label outputs, defining loss functions, and optimizing the models for the segmentation task. Preprocessing steps might involve data augmentation techniques specific to part-level annotation.

##### Dataset preparation:

* Load datasets with part annotations (e.g., ModelNet10, ShapeNetPart).
* Preprocess point clouds (normalization, sampling).
* Split into training, validation, and testing sets.

#### Model architecture:

Use PointNet or DGCNN as backbone architectures.

Adapt final layers for part segmentation:

PointNet: Output per-point segmentation scores.

DGCNN: Employ multi-scale feature aggregation for refined segmentation.

Loss function:

Use cross-entropy loss for multi-class segmentation.

Consider Dice loss or focal loss for handling class imbalance.

Training:

Optimize with Adam optimizer or variants.

Employ learning rate scheduling and regularization.

Monitor performance on validation set for early stopping.

#### Evaluation Metrics and Results Analysis

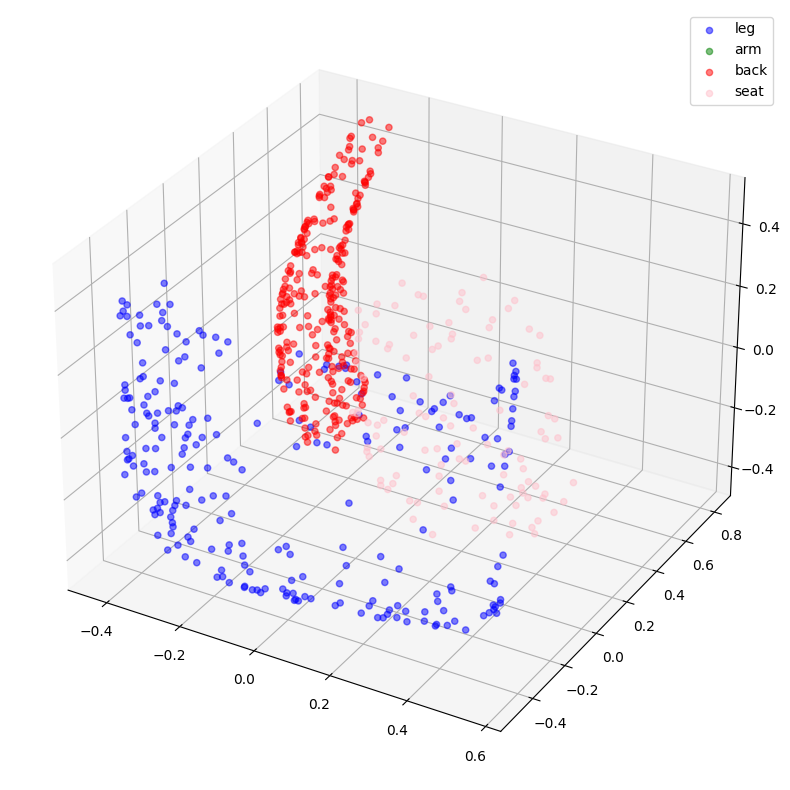
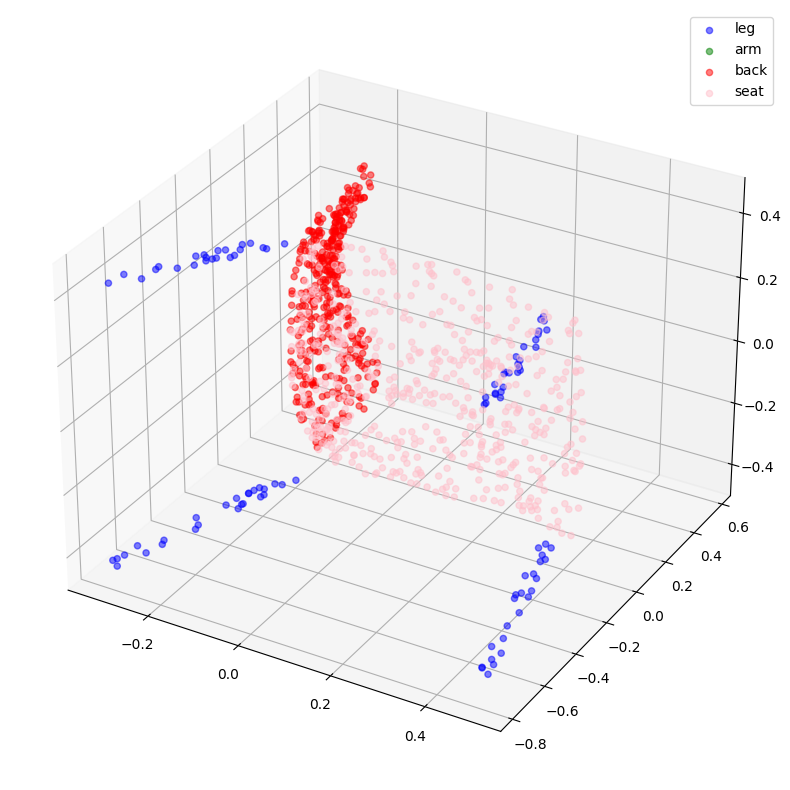
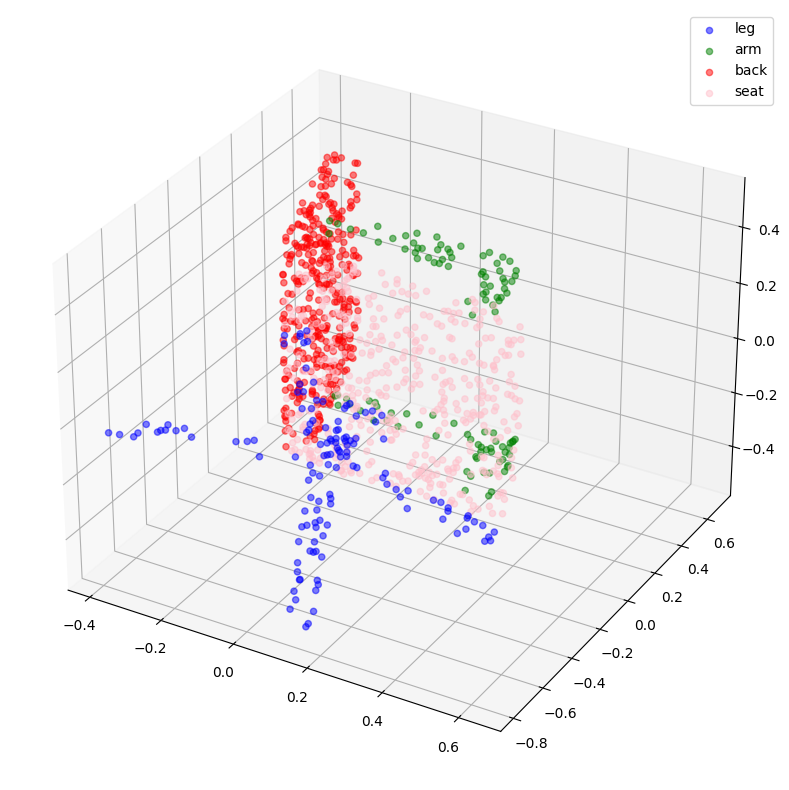
Evaluation metrics for part segmentation include IoU (Intersection over Union), mean IoU, per-category IoU, precision, recall, and accuracy. Results from experiments on benchmark datasets, along with visualizations of segmented parts, help analyze the models' performance in accurately delineating object parts within point clouds.

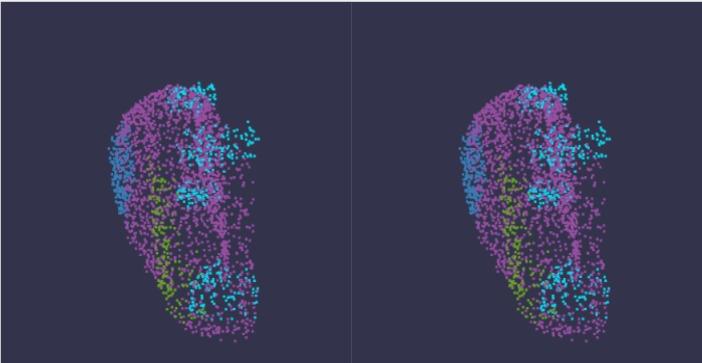
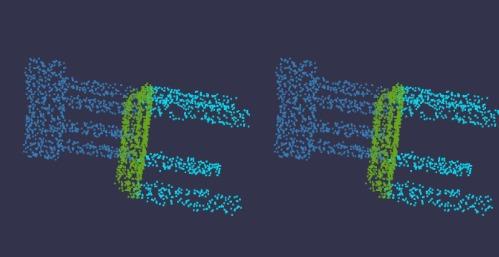
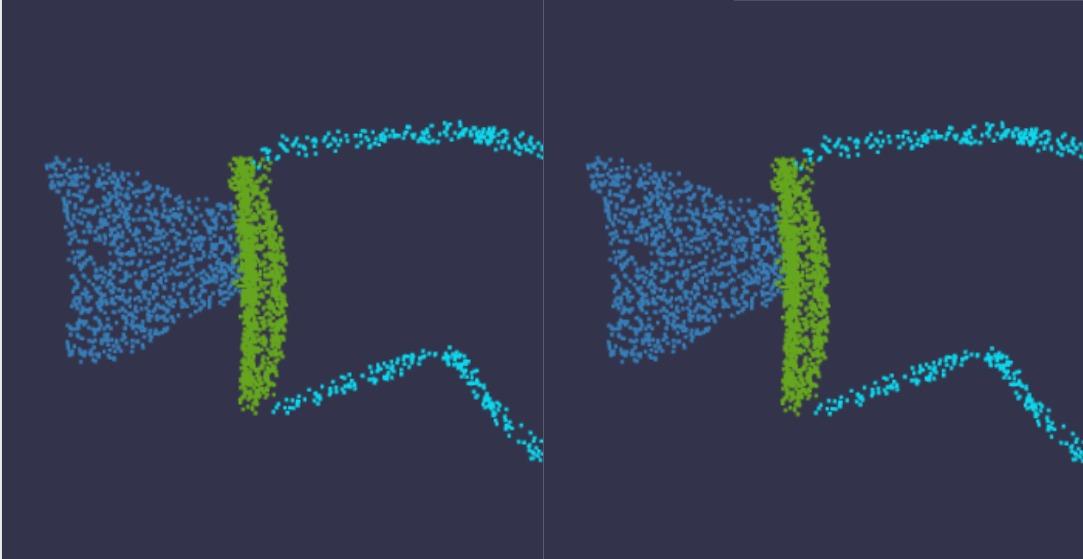
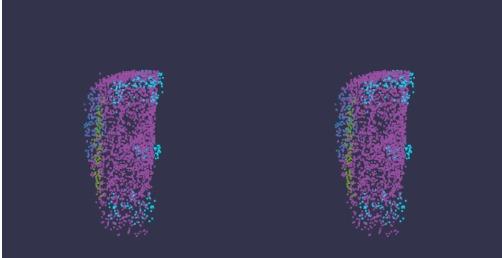
##### Metrics:

**Intersection over Union (IoU):** Measures overlap between predicted and ground truth segments.

**Mean IoU (mIoU):** Average IoU across all part categories.

**Precision and recall:** Assess model's ability to correctly identify true positives and negatives.



**DG-CNN Observations: (Ground Truth vs Prediction)**

##### Key findings:

Both PointNet and DGCNN achieve good segmentation results, but DGCNN often outperforms PointNet due to its focus on local features.

Performance varies across different object categories and dataset complexities.

Visualizations (e.g., colored segmentation masks) provide qualitative insights into model performance.

## Semantic Segmentation on DALES Dataset

#### Introduction to Semantic Segmentation

Semantic segmentation in the context of point clouds involves assigning a specific semantic label to each point, effectively partitioning the 3D space into meaningful segments based on semantic categories. This task is crucial for scene understanding, where each point is classified into object classes or categories, enabling detailed analysis and comprehension of the environment.

Definition: The task of assigning a semantic label (e.g., "car," "building," "tree") to each point in a point cloud, resulting in a dense, pixel-like labeling of the 3D scene.

##### Applications:

* Autonomous driving
* Robotics
* 3D scene understanding
* Urban planning
* Infrastructure management

##### Challenges:

* Large-scale, complex point clouds
* Diverse object categories and shapes
* Varying point densities and noise levels
* Irregular data structure

#### Application of PointNet and DGCNN on DALES Dataset

Utilizing PointNet and DGCNN for semantic segmentation on the DALES dataset involves adapting these architectures to handle the semantic labeling of points within complex real-world scenes captured by sensors like LiDAR. Preprocessing steps may include data augmentation, normalization, and suitable formatting of input and output labels to match the semantic annotation scheme of the dataset.

##### DALES Dataset:

* Large-scale aerial LiDAR dataset with over 500 million hand-labeled points.
* Captures urban scenes with buildings, vegetation, cars, roads, power lines, etc.
* Presents challenges due to density, noise, and unstructured nature of aerial LiDAR data.

##### Model Adaptation:

* Use PointNet or DGCNN as backbone architectures.
* Modify output layers to produce per-point semantic segmentation scores.
* Adapt hyperparameters and training strategies for aerial LiDAR data.

#### Evaluation Metrics and Performance Analysis

Evaluation metrics for semantic segmentation include IoU (Intersection over Union), mean IoU across different semantic categories, pixel accuracy, precision, and recall. Performance analysis involves assessing how accurately PointNet and DGCNN segment different objects or scene elements within the DALES dataset. Visualizations and qualitative assessments of segmented scenes contribute to understanding the models' effectiveness in capturing semantic information within 3D environments.

##### Metrics:

**Intersection over Union (IoU):** Measures overlap between predicted and ground truth segments for each class.

**Mean IoU (mIoU):** Average IoU across all classes, providing a comprehensive evaluation.

**Precision and recall:** Assess the model's ability to correctly identify true positives and negatives.

**Our Approaches**

##### Data Preprocessing

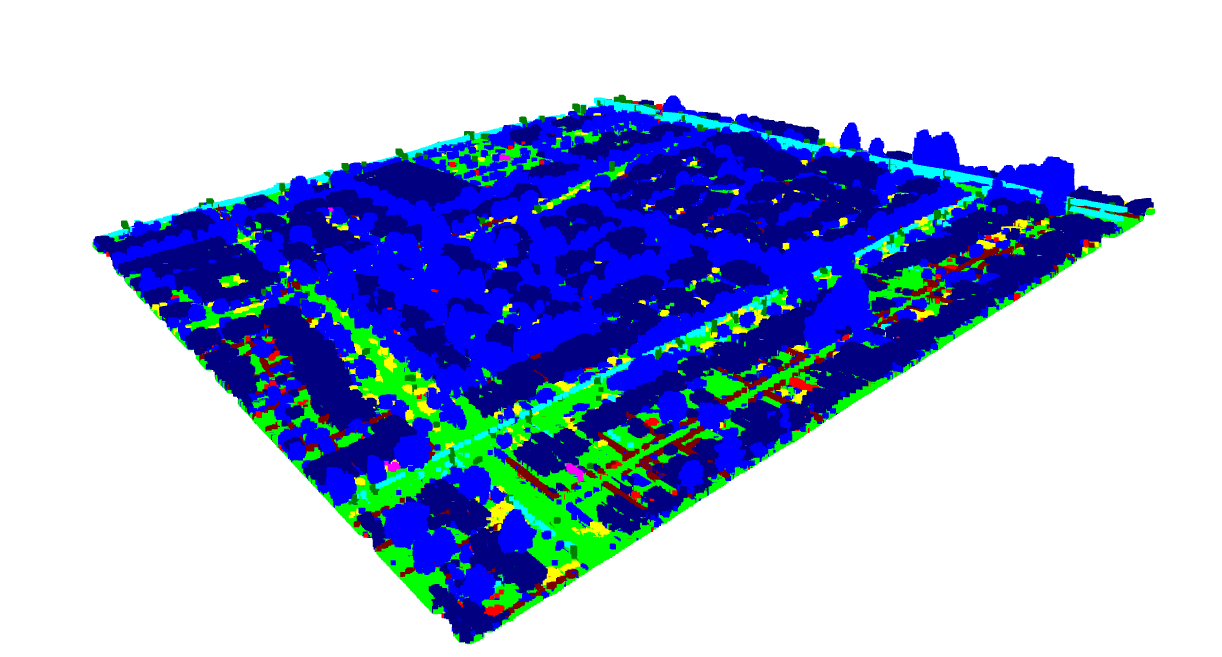
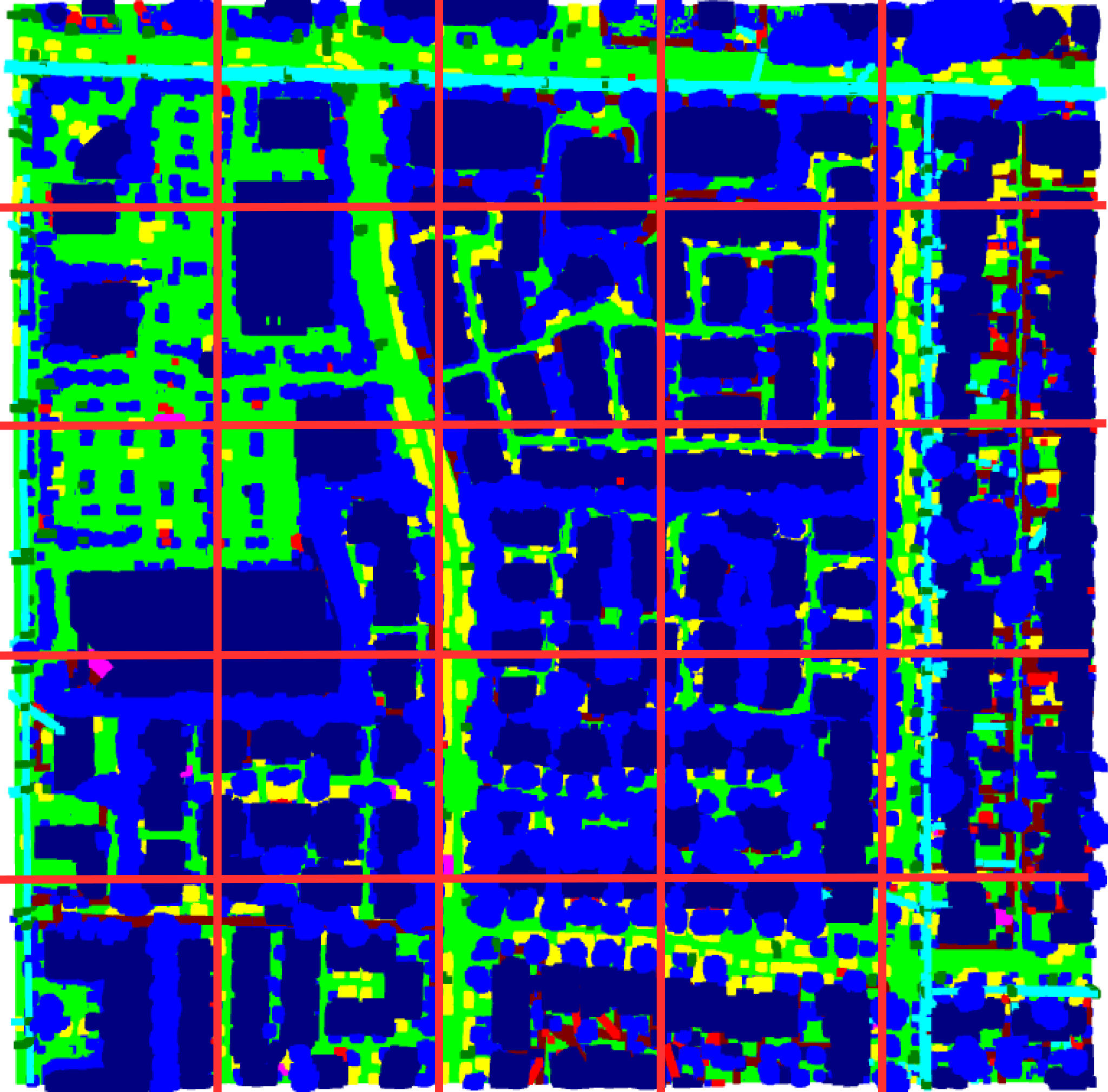
The Datasets from ShapeNet and ModelNet are already preprocessed and were made ready for training and testing. However, DALES dataset is only labeled while the data are still available as raw LAS files. Due to unavailability of pre-written libraries for PCD data, A lot of custom functions and logics are written and implemented for handling Point Cloud pre-processing.

**Data Cleaning and Transformation**

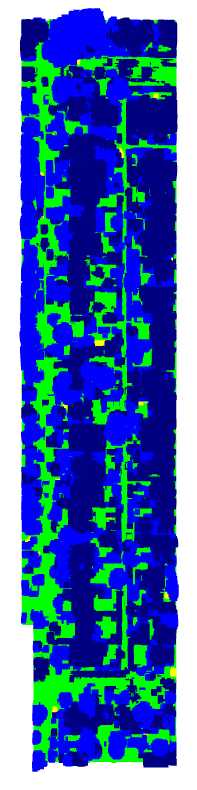
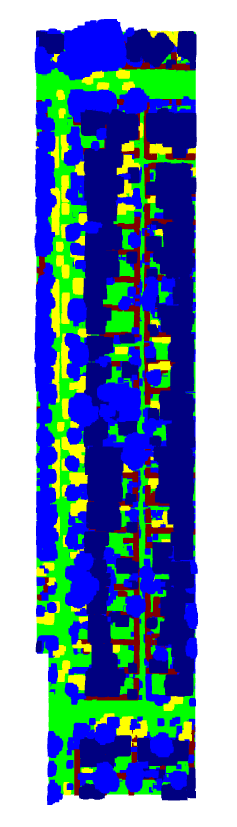
Missing values in DALES are replaced by using an intuitive KNN classification technique inspired by us. The model was trained with the labeled points and the non-labelled dirty points are classified based on K Nearest Neighbours. For Data Normalisation, StandardScaler and MinMaxScaler techniques are employed.

**Training Batch Preparation**

A Custom Cross-Validation technique has been employed by us to split the larger dataset into Chunks. (a chunk refers to a spatially organized subset of PCD that allows for more efficient processing and handling of large datasets). Each chunk is sliced from the 3D PCD by sorting X and Y coordinates using custom logic and functions and each of them is normalized separately. These chunks were then used as batch-wise data for feeding the model.

*Chunking - Wise Segmentation*

**True PCD vs Predicted PCD**



#### General Comparison Between PointNet and DG-CNN

A comparative analysis between PointNet and DGCNN involves evaluating their performance metrics on object classification tasks, discussing their strengths and weaknesses, computational efficiency, and generalizability. Highlighting the advantages and limitations of each model in terms of handling point cloud data for object classification purposes aids in understanding their suitability for different applications and datasets.

##### Strengths of PointNet:

* Simpler architecture, faster training and inference
* Effective for global feature extraction
* Well-suited for large-scale point clouds

##### Strengths of DGCNN:

* Superior capture of local geometric structures
* More robust to noise and occlusion
* Often achieves higher accuracy on complex datasets

##### Computational Efficiency:

* PointNet generally faster than DGCNN
* Trade-off between accuracy and speed to consider

##### Generalizability:

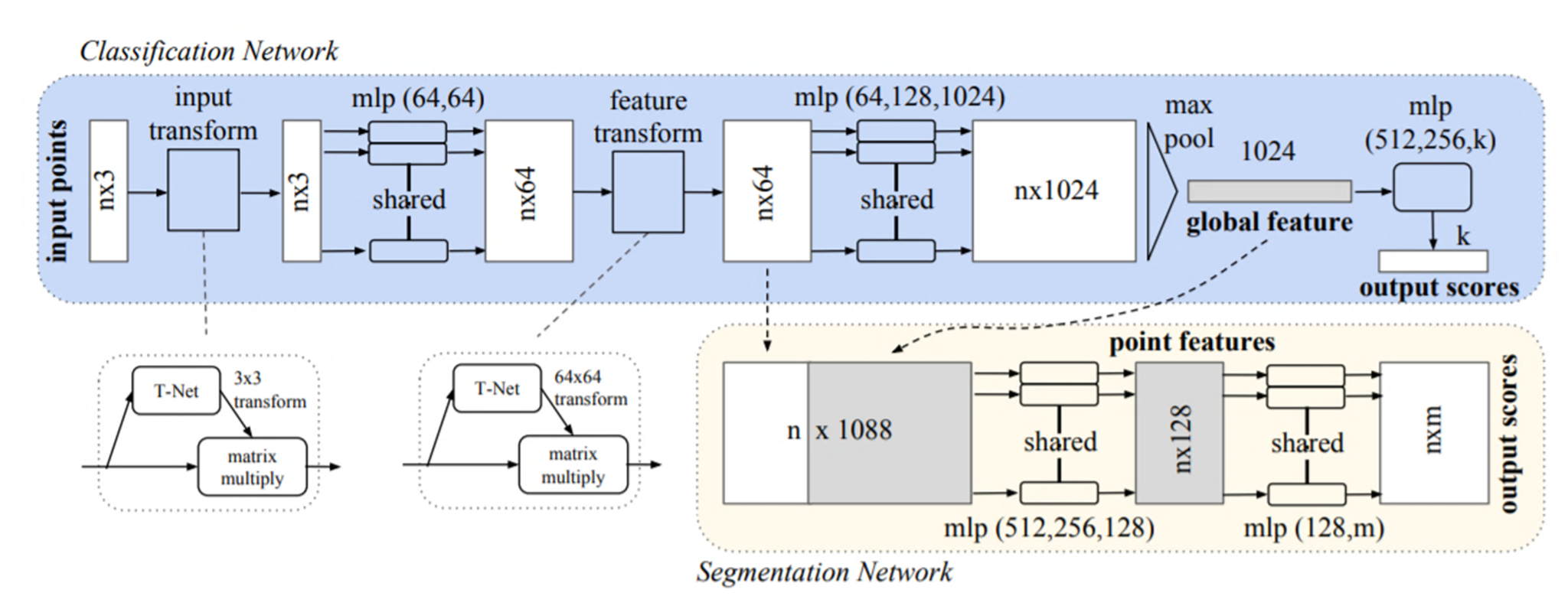
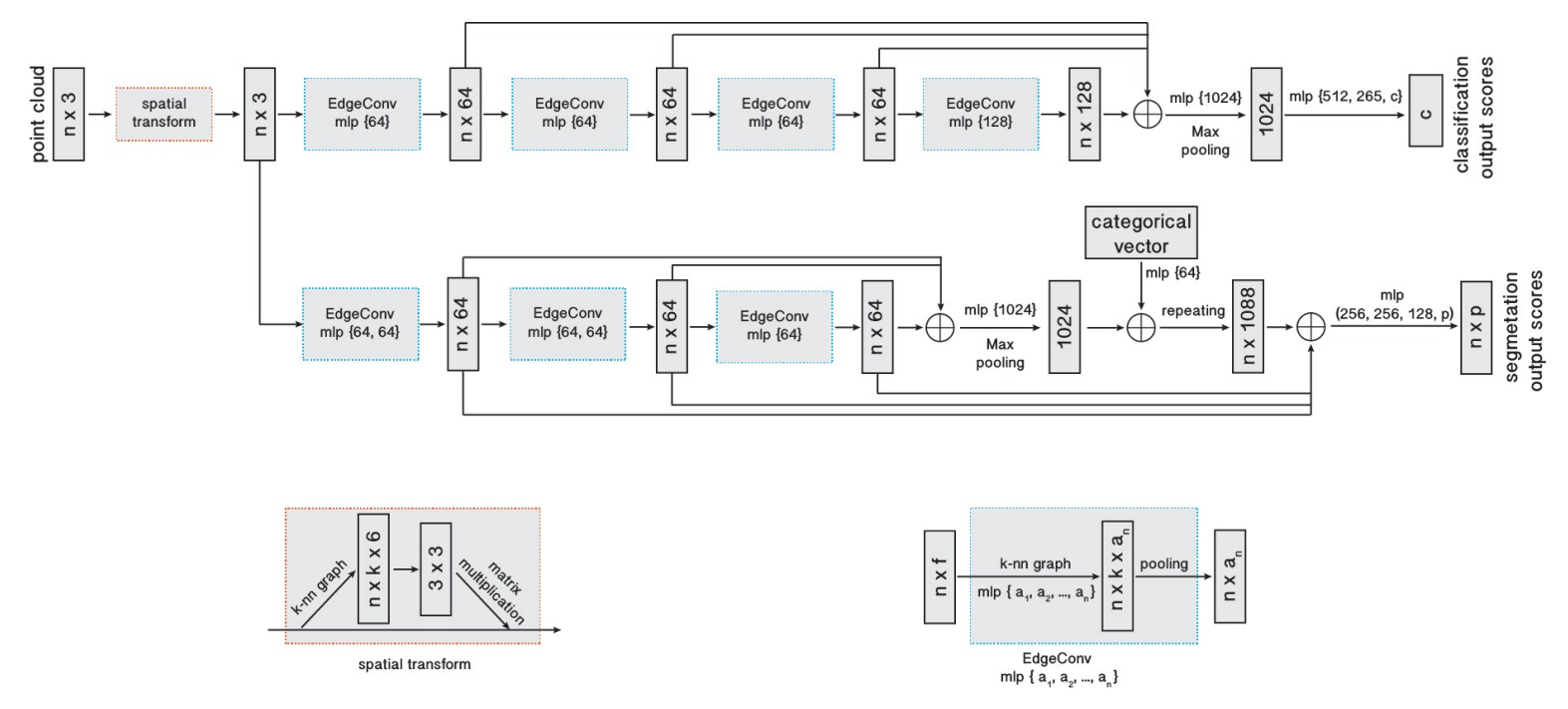
* DGCNN may generalize better to unseen data due to local feature focus
* PointNet's global features might be less adaptable

##### Accuracy comparison:

* Choice of model depends on dataset complexity and task requirements
* DGCNN often favored for datasets with fine-grained details and noise
* PointNet suitable for large-scale datasets where speed is critical

| **Dataset** | **Model** | **Task** | **Train Accuracy** | **Test Accuracy** | **Model IoU** | **Learning Rate** |
| --- | --- | --- | --- | --- | --- | --- |
| ShapeNet | PointNet | Classification | 90-95% | 85-90% | - | 0.001-0.01 |
| ModelNet | DG-CNN | Part | 92.1-95.2% | 95-98.9% | 97.5% | 0.001 |
| ModelNet | PointNet | Part Segmentation | 92-96% | 87-90% | - | 0.001-0.01 |
| Dales Dataset | PointNet | Semantic Segmentation | 85-88% | 84-87.8% | - | 0.001-0.01 |

#### Specific Comparison Between PointNet and DGCNN and Discussion

* ***Point Net Architecture -***
* ***DGCNN Architecture -*** 

**Architecture - Comparison and complexity analysis**

**PointNet:**

**1. Shared MLP (Multi-Layer Perceptron):**

**Functionality:**

* PointNet processes each point independently through a shared MLP.
* The shared MLP captures local features for each point.

**2. Max Pooling:**

**Functionality:**

* Global information aggregation is achieved through max pooling.
* Max pooling provides permutation invariance, making PointNet robust to different point orders.

**3. Symmetric Function:**

**Functionality:**

* The max pooling operation acts as a symmetric function, ensuring the network's permutation invariance.
* Symmetric functions enable PointNet to operate on unordered point sets.

**DGCNN:**

**1. EdgeConv (Edge Convolution):**

**Functionality:**

* DGCNN introduces EdgeConv layers to capture local geometric features based on the edges between points.
* The dynamic graph construction adapts to the local structure of the point cloud.

**2. Dynamic Graph Construction:**

**Functionality:**

* DGCNN constructs a dynamic graph based on nearest neighbors or other heuristics.
* The dynamic graph allows the network to adapt to variations in point distribution and local structures.

**3. Graph Pooling:**

**Functionality:**

* DGCNN uses graph pooling to aggregate information from local neighborhoods.
* Graph pooling enables the network to capture hierarchical features and maintain a global perspective.

##### Comparative Study of PointNet and DGCNN Performance

The comparative study between PointNet and DGCNN involves a comprehensive analysis of their performance across various tasks such as object classification, part segmentation, and semantic segmentation. Metrics like accuracy, IoU, precision, and recall are evaluated for both models across different datasets. Discussing their performance on specific challenges and datasets aids in understanding their relative strengths and weaknesses.

**Object classification:** DGCNN generally outperforms PointNet in terms of accuracy and F1-score, particularly on complex datasets with fine-grained details. However, PointNet's simpler architecture offers faster training and inference, making it suitable for large-scale datasets where speed is crucial.

**Part segmentation:** Similar trends emerge here, with DGCNN achieving higher mIoU scores due to its superior capture of local geometric features. Nonetheless, PointNet can be faster and more efficient for large point clouds.

**Semantic segmentation:** On LiDAR data, DGCNN is expected to excel due to its ability to handle fine-grained details and noise present in aerial scans. However, PointNet's efficiency might be advantageous for large-scale urban scenes.

**Considerations:**

* Computational efficiency: PointNet is generally faster than DGCNN.
* Generalizability: DGCNN may generalize better to unseen object instances.
* Data characteristics: Choice of model depends on dataset complexity and the importance of local features.

### Conclusion

The exploration of PointNet and DGCNN for processing point cloud data has revealed their effectiveness in various tasks within 3D data analysis. PointNet demonstrates proficiency in handling unordered point sets for object classification and part segmentation, while DGCNN leverages graph-based convolutions for tasks like semantic segmentation. Both models exhibit strengths in capturing local and global features within point clouds.

**Citations**

[1] C. Qi et al, “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation”, 2017

[2]W. Yang et al, “Dynamic Graph CNN for Learning on Point Clouds”, 2018

[3] <https://keras.io/examples/vision/pointnet_segmentation/>

[4]<https://github.com/keras-team/keras-io/blob/master/examples/vision/ipynb/pointnet.ipynb>