INDOOR OBJECT DETECTION

A THESIS SUBMITTED TO
THE FACULTY OF ARCHITECTURE AND ENGINEERING
OF
EPOKA UNIVERSITY

BY

STELA DOLLAKU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

MARCH, 2024

Approval sheet of the Thesis

This is to certify that we have read this thesis entitled **"INDOOR OBJECT DETECTION"** and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Assoc. Prof. Dr. Arban Uka
Head of Department
Date: 01. 03. 2024

Examining Committee Members:

Prof. Dr. Betim Çiço Computer Engineering) _____

Prof. Dr. Gëzim Karapici (Computer Engineering) _____

Assoc. Prof. Dr. Dimitrios Karras (Computer Engineering) _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Surname: Stela Dollaku

Signature: _____

ABSTRACT

INDOOR OBJECT DETECTION

Dollaku, Stela

M.Sc., Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Dimitrios Karras

The process of identifying and localizing various items, usually in pictures that represent objects found in daily life, is called object detection. Object detection identifies each object as belonging to a specific class and creates a bounding box around it. In this thesis we focus our study in indoor datasets. The purpose of the thesis is to evaluate different methods of object detection in indoor datasets. We also aim to compare these results with each other, in order to try and find the best methods for the selected datasets.

Overall, these results highlight how crucial it is to carefully evaluate model architectures, preprocessing methods, and dataset properties in order to fully utilize deep learning for 3D applications. Subsequent investigations may examine techniques to mitigate class disparities and improve model resilience in a variety of object categories and shapes.

Keywords: *object detection, classification, segmentation, point cloud, indoor dataset*

ABSTRAKT

DEDEKTIMI I OBJEKTEVE NE AMBIENTE TE MBYLLURA

Dollaku, Stela

Master Shkencor, Departamenti i Inxhinierisë Kompjuterike

Udhëheqësi: Assoc. Prof. Dr. Dimitrios Karras

Procesi i identifikimit dhe lokalizimit të objekteve të ndryshme, zakonisht në figura që përfaqësojnë objekte të gjetura në jetën e përditshme, quhet dedektimi i objekteve. Dedektimi i objekteve identifikon çdo objekt si i përkatësuar në një klasë të caktuar dhe krijon një kutië afër tij. Në këtë tezë, ne fokusohemi në studimin tonë në ambiente të mbyllura. Qëllimi i tezës është të vlerësojmë metodat e ndryshme të zbulimit të objekteve në ambiente të mbyllura. Ne gjithashtu synojmë të krahasojmë këto rezultate me njëri-tjetrin, në mënyrë që të përpiqemi të gjejmë metodat më të mira për skedarët e zgjedhur.

Fjalët kyçe: *zbulimi i objekteve, klasifikimi, segmentimi, dataset, vision kompjuterik*

Table of Contents

ABSTRACT iii

ABSTRAKT iv

LIST OF TABLES vii

LIST OF FIGURES viii

CHAPTER 1 1

INTRODUCTION 1

1.1 Thesis Objective 2

1.2 Thesis Outline 2

CHAPTER 2 4

BACKGROUND INFORMATION 4

2.1. Definition of indoor object detection 4

2.2. Image Segmentation and Classification 5

2.3. Issues and challenges in 3D indoor object detection 6

2.4. Data Acquisition Methods 8

CHAPTER 3 10

LITERATURE REVIEW 10

3.1. Methods and techniques for 3D detection of indoor objects 10

3.2. Point-based Methods 11

3.3. Voxel-based Methods 12

3.4. Deep Learning- Based Approach 13

3.5. Hybrid Approaches 14

CHAPTER 4 16
METHODOLGY 16
4.1. Dataset Description 16
4.1.1. ModelNet Dataset 16
4.1.2. ShapeNet Dataset 18
4.2. PointNet 20
4.3. VoxNet 21
4.4. Relation-Shape CNN (RS-CNN) 23
CHAPTER 5 26
EXPERIMENTAL RESULTS 26
5.1. Object Classification Results 26
5.2. Misclassification Matrices 30
5.3. Object Segmentation Results 32
5.4. Results in Difficult Scenes and Occluded Spaces 34
5.5. Conclusion and Discussion 36
References 38
APPENDIX 41

LIST OF TABLES

Table 1.Overview of datasets for point cloud semantic segmentation (where S ← synthetic environment, Oc ← object classification, Ps ← part segmentation, Tm ← thousand models). 20

Table 2. Model Performance PointNet 26

Table 3. Model Performance VoxNet 27

Table 4. Model Performance RS-CNN 29

Table 5. Accuracy values for ModelNet 30

Table 6. Segmentation Results 33

LIST OF FIGURES

Figure 1. Deep learning tasks. a) Object classification b) Part segmentation c) Object detection 6

Figure 2. Overview of Data Acquisition Techniques: Passive and Active Methods....8

Figure 3. Classification of 3D Object Detection Methods 10

Figure 4. Some samples of the CAD models that compose the ModelNet10 dataset 17

Figure 5. Part segmentation examples on the ShapeNet dataset. 19

Figure 6. PointNet Architecture.....20

Figure 7. VoxNet Architecture 22

Figure 8. Relation-Shape CNN's Architecture 24

Figure 9. Qualitative results 35

CHAPTER 1

INTRODUCTION

The most incomprehensible thing about the world is that it is comprehensible.

-Albert Einstein

In the era of digital advancement, our daily lives are closely connected with digital cameras, internet-connected devices, and smartphones. Image and video collections are growing more and more each day; More than 1.1 trillion photos were taken in 2016 according to Info-trends; estimates using still cameras and mobile devices [1]. The same prediction states that by 2020, the amount will have increased to 1.4 trillion. A large number of these photos are posted online or kept in cloud storage services. In 2014 saw the daily submission of more than 1.8 billion photos to the most widely used websites, including Facebook and Instagram [2].

A machine interprets an image as a grid of numbers. We must have some understanding of the substance of this data in order to manage it all properly. Many different image-related tasks benefit from the automated processing of image contents. This entails bringing the so-called semantic gap between the human perception of the same images and the pixel-level information contained in the image files for computer systems. Deep learning and computer vision make an effort to overcome this barrier.

The way that individual neurons fire in response to input and only see a very small portion of the overall input/processed data is the inspiration behind deep learning (DL). It has made significant contributions to computer programming approaches, allowing a machine to carry out tasks that almost perfectly mimic human intellect. Deep learning is widely used in the robotics, medical, and automation industries. According to forecasts, the Computer Vision market is expected to reach $33.3 billion in 2019, which will support the notable rise in the consumer, robotics, and machine vision domains. Because of its findings, which are mostly obtained in applications requiring language processing, object detection and picture classification - it has become the most talked about technology.

The process of identifying and localizing various items, usually in pictures that represent objects found in daily life, is called object detection. Object detection identifies each object as belonging to a specific class and creates a bounding box around it. This enables us to locate the specific things in the scene and determine how they are moving through it. Both indoor and outdoor datasets can be used for it.

In this thesis we focus our study in indoor datasets. The purpose of the thesis is to evaluate different methods of object detection in indoor datasets. We also aim to compare these results with each other, in order to try and find the best methods for the selected datasets.

1.1 Thesis Objective

In this thesis, the main investigations are to advance the understanding and application of object detection. First, it looks into object detection studying the image classification. By exploring different methods, the goal is to enhance the accuracy and efficiency of image classification through robust object detection algorithms.

Second, is to study the methods on the context of image segmentation. The objective is to study different techniques applied to image segmentation. The methods are trained and compared using two distinct datasets—one for image classification and another for segmentation. This comparative analysis aims to elucidate the strengths and limitations of these methods in diverse visual recognition task.

1.2 Thesis Outline

In Chapter 2, object detection in indoor spaces is explained focusing on its importance and challenges faced.

In Chapter 3, object detection literature is studied and different methods are taken in consideration.

In Chapter 4, an overview of two dataset in the thesis is given. Also, explanations of each model's architecture and its adjustment for 3D object detection.

In Chapter 5, the outcomes of each experiment are presented, providing he results of each method for the two datasets. Also, the results mentioned in the chapter are discussed and explained in detail.

CHAPTER 2

BACKGROUND INFORMATION

The second chapter of this thesis provides an introduction to the foundational concepts and developments of 3D object detection. The goal is to give a complete outline of the characteristics and challenges in indoor 3D object detection and their importance.

2.1. Definition of indoor object detection

Indoor object detection refers to the process of identifying and categorizing objects within indoor environments. It involves utilizing computer vision techniques and algorithms to analyze visual data captured from cameras or other sensors to recognize and understand the objects present in indoor scenes.

When utilized for indoor scene classification, conventional approaches do not perform as well as they do for outdoor scene classification [3]. Early attempts at enhancing indoor scene classification included methods like bag-of-visual words that attempted to exploit both local and global spatial date [4]. In 2019, Chen et al. [5] examined scene classification through the integration of traditional scene classification techniques with natural language processing methodologies. They used a convolutional neural network (CNN) module and a scene parser module to segment the scene in order to generate an ordered top 5 prediction for a given image. By putting these split parts into the word embedding module, the classification accuracy of interior scenes and the top 5 predictions both increased. Three super-categories—the home, the mall, and the school—were used to train and test their model. They reasoned that while GPS tracking would be adequate to ascertain a possible agent's overall surroundings, it would not be adequate to ascertain the precise location and area the agent would be in. The limitations of restricted scene diversity may be mitigated by limiting the range of room classes to setting-specific choices, given the intersection of many scene categories. If GPS indicates, for example, that an agent is on a school campus, an indoor room classification model trained on indoor school settings may be used to predict the room category that the agent is in. The goal of indoor object detection is to enable machines or computer systems to perceive and interpret their

surroundings in indoor settings. This detection can involve detecting and classifying objects based on their appearance, shape, texture, or other visual features. The detection process typically involves several steps, including image acquisition, preprocessing, feature extraction, and classification or matching with known object models or categories.

Indoor object detection has various applications, including robotics, home automation, augmented reality, indoor navigation, security systems, and more. By accurately recognizing objects in indoor environments, machines can make informed decisions, interact with the environment, and perform tasks effectively and autonomously

2.2. Image Segmentation and Classification

The process of dividing an image or video into meaningful sections in order to distinguish and identify particular items or areas of interest is called image segmentation. It accomplishes objectives including understanding object boundaries, retrieving detailed data, and facilitating additional analysis.

Traditionally, surface characteristics like normal, curvature and orientation [6] [7] are used for segmentation. Point cloud segmentation has used feature based deep learning methods that divide points in several characteristics. Distinct object pars, also discussed as distinct class categories, also discussed as semantic segmentation could be the aspects.

Parts segmentation gets its name from the fact that every point in the input point cloud is intended to characterize a specific item, and the objective is to allocate each point to a part, as illustrated in figure 1b. The aim of semantic segmentation is to allocate every point to a certain class parameter.

The process of annotating complete photos based on the elements they include is known as image categorization. It describes what is there without mentioning where. Assigning a name or category to an image or video is known as image classification. Accurately assigning an image to a particular, pre-established category or label is the

aim. This is achieved by training a dataset of photos with labels indication their various categories.

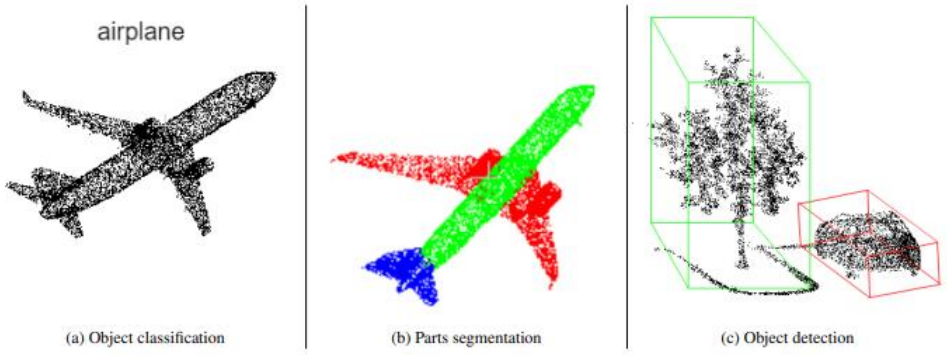


Figure 1. Deep learning tasks. a) Object classification b) Part segmentation c) Object detection

2.3. Issues and challenges in 3D indoor object detection

3D indoor object detection presents several challenges and issues that need to be addressed for accurate and robust recognition. Some of the key challenges include:

Occlusions: In indoor environments, objects are often partially occluded by other objects or obstacles, making it challenging to obtain complete and unobstructed views. Occlusions can significantly affect the visibility of object features, leading to difficulties in accurate detection.

Lighting and Shadows: Indoor lighting conditions can vary, leading to uneven illumination and cast shadows. These variations can cause significant changes in the appearance and texture of objects, making it challenging to extract reliable features for detection.

Cluttered Environments: Indoor scenes often contain a variety of objects and clutter, such as furniture, decorations, and other items. The presence of clutter can

increase the complexity of object detection as it may introduce distractions and visual ambiguities, making it harder to differentiate and identify specific objects.

Variability in Object Appearance: Objects in indoor environments can exhibit significant appearance variations due to factors such as different viewing angles, scale changes, deformations, and variations in texture or color. Handling these appearance variations is crucial for achieving robust and invariant object detection.

Limited Training Data: Obtaining labeled training data for indoor object detection can be challenging and time-consuming. Collecting a diverse and representative dataset that encompasses various indoor scenes, object categories, and variations is crucial for training accurate detection models.

Real-Time Performance: Real-time performance is often a requirement for practical applications of indoor object detection, especially in robotics or interactive systems. Achieving fast and efficient detection algorithms that can process the visual data in real-time is a significant challenge, considering the complexity of 3D object detection.

Generalization across Environments: Objects in indoor environments can vary across different locations, layouts, and contexts. Ensuring the generalization of detection models across different indoor environments is essential to enable robust object detection in unseen or new environments.

Sensor Limitations: The choice of sensors used for capturing the indoor scene data can impact the quality and availability of information for object detection. Sensor limitations such as low resolution, limited field of view, or noise can affect the accuracy and reliability of detection algorithms.

Addressing these challenges requires the development of sophisticated algorithms and techniques that can handle occlusions, variations in appearance, cluttered scenes, and real-time processing. Additionally, the availability of large and diverse datasets, advancements in sensor technologies, and improvements in computational resources contribute to addressing these challenges effectively.

2.4. Data Acquisition Methods

The process of capturing the appearance and the shape of real objects, is accomplished using active and passive methods. Among the passive methods, we mention some of them.

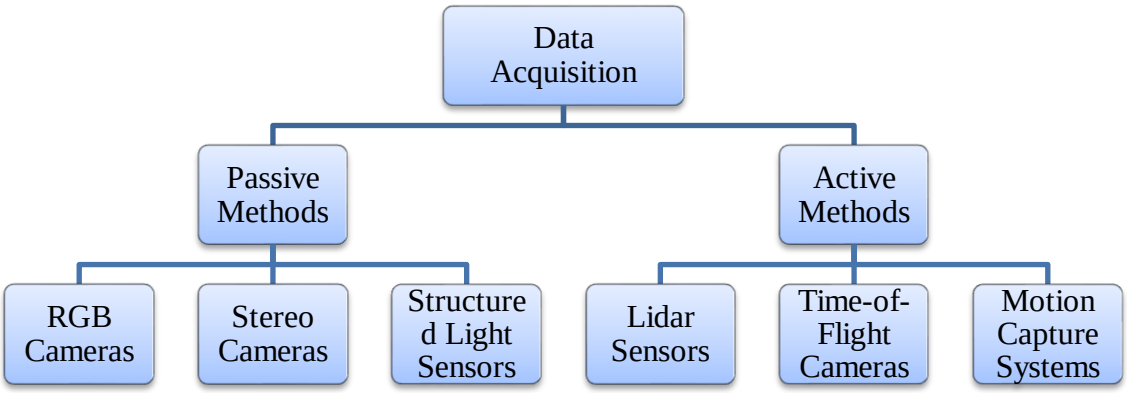


Figure 2. Overview of Data Acquisition Techniques: Passive and Active Methods.

- RGB-D Cameras have been widely adopted, incorporating an RGB camera, a depth sensor and other components like a microphone and a USB port for connection to the computer. The depth sensor, which makes it easier to record 3D point clouds, measures each object's distance from the camera's horizontal optical axis using infrared structured light. Dai et al. (2017) [8] introduces a new method using RGB-D cameras for semantic labeling of three-dimensional point clouds. This method uses both color and geometric features to perform fine-grained semantic segmentation, contributing to better understand indoor scenes and scene reconstruction and robotics.

- Stereo Cameras that come with integrated sensors facilitate and speed up the implementation of spatial analytics. These cameras provide a point cloud, a depth map and a color image of the scene that is in its field of view, computed through intrinsic camera parameters in the disparity map. Leibe et al. (2007) [9] demonstrated the

effectiveness of a car-mounted stereo rig for reasoning about scene depth, detect pedestrians and cars, and tracking them over time

- Sensors for Structured Light can be effectively replace passive stereoscopic sensors in controlled environments (medical, industrial), weakly textured environments (night vision, underwater vision) and weakly textured environments (biometry, anthropometrics). Structured light sensors are made up of one or more cameras and one or more light sources. It analyzes the patter distortion that results from projecting preset patterns onto a target. [10]

Active approaches involve actively illuminating the scene using external light sources, which enables the capture of additional depth information. This illumination can be achieved through various techniques such as LiDAR, time-of-flight (ToF), Motion Capture Systems.

- LIDAR (Light Detection and Ranging) sensors have the ability to emit laser beams and measuring the period it takes for the pulses to return. The sensors can calculate the distance to obstacles in various directions thanks to this time information. Lidar point cloud and picture data have been combined in multiple studies for a range of computer vision applications, including 3D object detection [11], [12], [13], [14]. The author Roland Bruggman et al. (2017) [15] focuses on effectively extracting planes from indoor LiDAR point clouds, suggesting an agglomerative hierarchical clustering algorithm that makes accurate identification of planar regions. This method is applicable to robotics, indoor mapping, augmented reality.

- Time-of-Flight (ToF) cameras create depth image, and each pixel in image represents the distance to a corresponding point in the scene. This method relies on timing the light's journey from the camera to the objects in the picture and back again to get a highly accurate distance reading between the camera and the objects. ToF cameras are suited for dynamic environments and applications that need quick reactions because of their many benefits, one of which is their real-time depth sensing capabilities.

CHAPTER 3

LITERATURE REVIEW

The literature review in this thesis offers an examination of existing research in the domain of 3D object detection, specifically focusing on method classification and noteworthy contributions by other researchers. The review gives insights from seminal works by pioneers in the field and each contributing distinctive perspectives to the overarching goal of accurate indoor 3D object detection. Furthermore, it explores recent advancements in hybrid methods, shedding light on novel approaches that combine the strengths of different models

3.1. Methods and techniques for 3D detection of indoor objects

Methods and techniques for 3D detection of indoor objects can be categorized into several approaches. In the context of 3D object detection, feature processing methods can be categorized in four types base in the form of point cloud representation. Here are some commonly used methods:

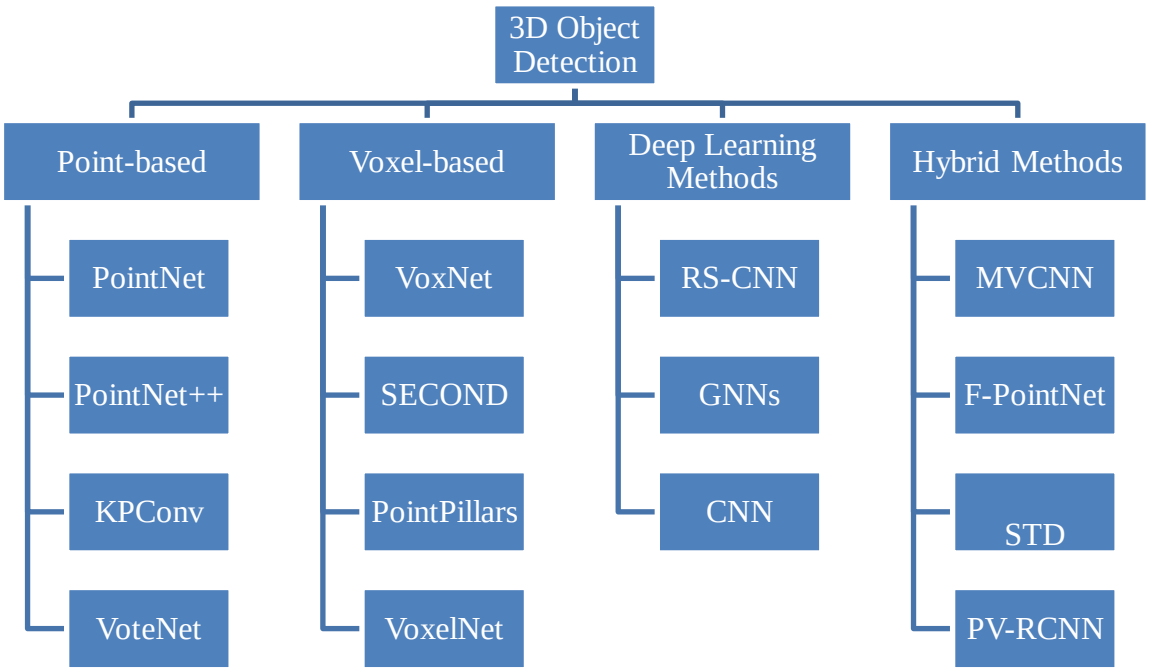


Figure 3. Classification of 3D Object Detection Methods

3.2. Point-based Methods

In recent years, point-based methods have emerged as a groundbreaking approach in computer graphics and computer vision, offering unique solutions for processing unordered point cloud data. From object classification to 3D object detection, their versatility and efficiency have garnered widespread attention, propelling the field of point-based deep learning to new heights. Among these methods, PointNet and PointNet++ stand out as pioneering architectures that have reshaped how 3D data is analyzed and understood.

a. **PointNet**: The architecture is specifically designed for point cloud data. PointNet models process individual points independently. PointNet-based models have achieved outstanding results in tasks including semantic segmentation, classification and part segmentation. Qi et al. (2017) [16] presented a method that represents unordered point clouds using deep learning framework in a set of 3D point. This technique offers means of obtaining both global and local features. The work done by Charles R. Qi, Wei Liu (2018) [17] extended PointNet towards the problem of 3D object detection from RGB-D data. They introduced Frustum PointNets to utilized 2D object propsals from RGB image and projected them into 3D frustums. This technique achieved the most advanced performance in 3D object detection using RGB-D data.

b. **PointNet++:** is an improved version of PointNet designed specifically for processing unordered point cloud data. PointNet++ utilizes hierarchical feature extraction to capture global and local context in point clouds. It is implemented in other methods such as PointRCNN, STD and others. Researches have used PointNet++ in various tasks, from classification and segmentation to object detection. This methos is still ongoing research and development. Charls R. Qi, Li Yi, Hao Su (2017) [18] used PointNet++ in a set of local point neighborhoods and local attributes to capture wider context while capturing fine-grained information. Improves performance was detected in tasks such as scene understanding and segmentation.

3.3. Voxel-based Methods

The 3D equivalent of 2D image pixel is called a voxel. The technique of transforming a continuous geometric entity into a collection of discrete voxels that most closely resemble the thing is known as voxelization. The field of 3D object detection has greatly improved thank to voxel-based techniques, which provide reliable and effective ways to analyze point cloud data. Voxel-based methods provide a consistent and volumetric representation of the scene by transforming complex and unstructured point cloud data into regular 3D grids of voxels.

a. **VoxNet**: The VoxNet technique is a 3D object classification technique that takes in 3D voxelized representation of objects. Firstly, it was proposed by Daniel Maturana and Sebastion Scherer in their work [19] in 2015. Scherer and Maturana evaluated the method on several benchmark datasets, demonstrating its effectiveness in real-time object recognition tasks. The experiments showed that VoxNet achieved competitive accuracy while being computationally efficient, making it suitable for real-time applications in resource-constrained environments. The motivation behind VoxNet stemmed from the need for efficient and real-time object detection in 3D point cloud data, particularly for applications in robotics, augmented reality, and autonomous systems. While traditional methods for object detection were primarily focused on 2D image data, the rise of 3D sensors and advancements in 3D scanning technologies demanded new approaches that could handle 3D data directly. VoxNet was the first to use the voxeliation technique to transform unstructured point clouds into normal voxels. They then used 3D CNN to predict the semantic labels of the occupied voxels by standard procedures. This technique addressed the issue of unstructured point clouds, but it was limited by the sparsity and high computing complexity of 3D CNN, which resulted in low voxel arrangement efficiency.

b. **SECOND**: a seminal work by Yan et al. (2018) [20], has made significant contributions to the field of voxel-based object detection. This work implements two sparse operators using GPU-based hash tables and develops a sparse convolutional network for extracting 3D voxel features. Building upon the foundation of SECOND, researchers have explored various avenues for improvement. Yan et al.

(2018) inspired the development of two-stage detectors that leverage the sparse convolutional network as a crucial component. Furthermore, the Transformer architecture, known for its success in natural language processing, has been introduced into voxel-based detection enabling novel approaches to feature extraction and object detection. The impact of SECOND and its associated advancements is evident in the extensive adoption of its network architecture in voxel-based detectors. Consequently, SECOND has emerged as the de facto backbone network for voxel-based object detection. The research community continues to explore avenues for advancing the sparse operators, extending the capabilities of the SECOND framework, and integrating.

3.4. Deep Learning- Based Approach

Deep Learning-based approaches are methods for solving complicated issues that rely on artificial neural networks—more especially, deep neural networks. Neural networks having several layers, or "deep architectures," are used in deep learning, a branch of machine learning, to learn and represent data in a hierarchical fashion.

a. **3D Convolutional Neural Networks (CNNs):** 3D CNNs extend traditional 2D CNN architectures to handle volumetric data, such as voxel grids or 3D point clouds. These networks employ 3D convolutional operations to learn hierarchical representations of objects in 3D space. They capture local and global features, enabling accurate object detection. Proposed by D. Maturana and S. Scherer (2015) [21].

b. **Relation-Shape CNN (RS-CNN)** is an influential technique in the field of 3D shape analysis and understanding. Proposed by Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas in their seminal paper "RS-CNN: Point Cloud Based 3D Object Detection with Relation-Shape Convolution" (2019) [22], RS-CNN addresses the challenging task of 3D object detection using point cloud data. By introducing the idea of Relation-Shape Convolution and utilizing deep learning, the method allows the network to capture both local geometric features and the relationships between various points in a point cloud. The efficiency of the technique has greatly advanced 3D form analysis and encouraged more study into the use of point

cloud data for a range of applications, including augmented reality, robotics, and autonomous driving. As a result, RS-CNN stands as a critical milestone in the development of sophisticated and robust techniques for 3D shape understanding and recognition.

c. **Generative Models**: 3D object representations can be generated and modeled using generative models, such as Variational Autoencoders (VAEs) or Generative Adversarial Networks (GANs). These models learn to generate synthetic 3D object data, which can be used to augment the training data or generate realistic 3D object samples for recognition tasks. H. Wu et al. propose a Variational Autoencoder (VAE) framework for generating and modeling 3D object representations. This generative model learns to generate synthetic 3D object data, which can be used to augment training data or generate realistic samples for recognition tasks

3.5. Hybrid Approaches

Hybrid approaches refer to methods that combine multiple techniques or models from different domains to tackle complex problems. These approaches leverage the strengths of each component to achieve better overall performance and robustness. Hybrid methods often aim to overcome limitations or challenges those individual techniques may face when applied in isolation.

a. **Multi-View CNN (MVCNN)**: This approach is initially designed for 2D shape recognition, but can be adapted for object detection tasks. It considers multiple views of an object and apply convolutional neural networks on each view, combining information for a holistic representation

b. **F-PointNet**: F-PointNet is an extension of PointNet that incorporates frustum-based 3D object detection. It combines point-wise processing with the analysis of frustums (bounding volume in the 3D space), providing a hybrid approach.

c. **Point-Voxel Feature Set Abstraction (PV-RCNN):** An addition to the RCNN family, PV-RCNN combines voxel- and point-wise features. For point-wise features, it makes use of a set abstraction module similar to PointNet++, and for comprehensive 3D object detection, it leverages voxel-wise features.

d. **Single-Stage 3D Object Detection (STD):** A single-stage 3D object recognition system called STD makes use of point cloud and voxel representations. To enable accurate and efficient object detection, it integrates volumetric and point-wise data.

CHAPTER 4

METHODOLGY

In this section, we provide an overview of the dataset used for the implementation of three models considered in this paper. We present detailed explanations of each model's architecture and its adjustment for 3D object detection. Additionally, we discuss the specific parameters and configurations used for each model.

4.1. Dataset Description

The dataset utilized in this study is crucial for evaluating the performance of the proposed 3D object detection models. We use two different datasets; ModelNet10 and ShapeNet. The first dataset we use image classification models to train the methods, while the second dataset is used for segmentation.

4.1.1. ModelNet Dataset

We use the ModelNet10 dataset, the smaller 10 class version of the ModelNet40. ModelNet10 is widely used, particularly when it comes to 3D shape analysis and deep learning, especially for 3D object classification tasks. The dataset is a collection of 3D CAD models from each object category that were found online by searching for each term associated with object category. The ModelNet-10 dataset has a collection of over 5000 models classified into 10 categories: toilet, table, desk, sofa, night stand, monitor, dresser, chair, bed and bathtub; and divided into training and test sets. Machine learning models are taught on the training set, and their performance is assessed on the testing set. The split is designed to ensure that each category is represented proportionally in both sets, maintaining a balanced distribution of objects across the dataset.

The dataset is organized into folders; within each folder contains files that represent the 3D geometry of the object, typically stored in a standardized format such as '. off' or '. ply'. Each object in the dataset is represented as a 3D mesh or point

cloud. A 3D object's geometry is encoded in terms of a combination of edges, vertices, and faces using a mesh representation.

Each file contains the 3D shape data that represents the points and triangles composing the mesh. The formant appears as follows:

$$\begin{matrix} x1 & y1 & z1 \\ x2 & y2 & z2 \\ x3 & y3 & z3 \end{matrix}$$

$$\begin{matrix} f & v1 & v2 & v3 \\ f & v1 & v2 & v3 \\ f & v1 & v2 & v3 \end{matrix}$$

Each line represents a 3D point with its coordinates(x,y,z). The lines following the point cordinates represent traingles (faces) defined by vertex indices. This structure allows for the precise representation of the object's geometry through the coordinates of its vertices and the definition of its faces using these vertices, essential for various gemoetric processing tasks and 3D shape analysis.

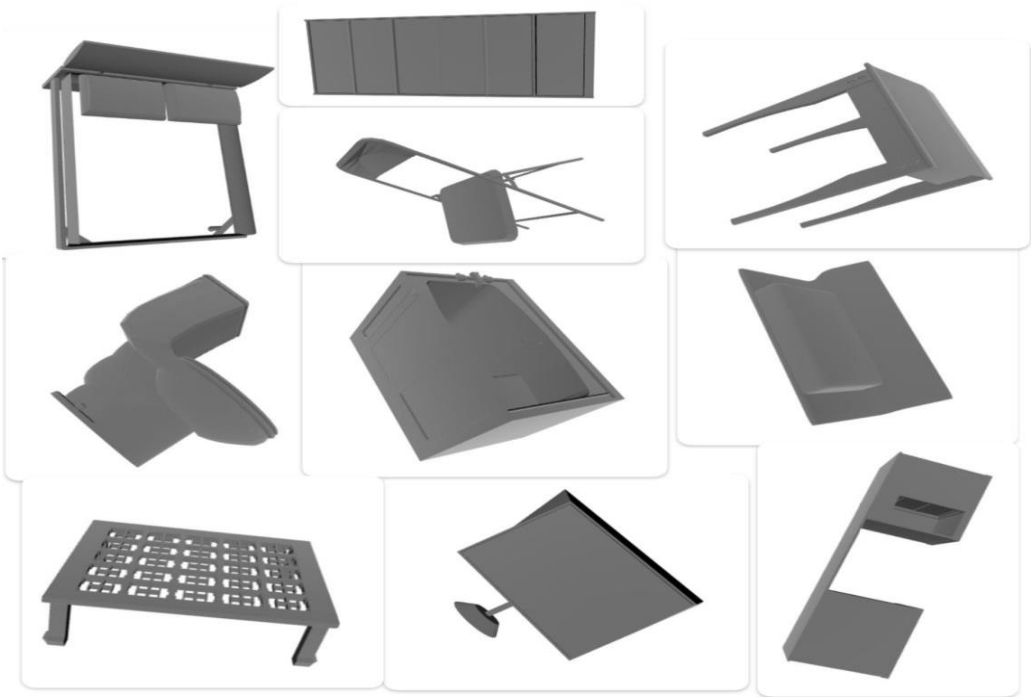


Figure 4. Some samples of the CAD models that compose the ModelNet10 dataset

Training and test splits are also included by the dataset's authors. We chose this dataset since it has enough samples to rapidly train our architecture. We must first extract the object point clouds from the ModelNet10 collection using a real 3D sensor in order to use the CAD models for our objectives. We found the CAD items in the middle of a tessellated sphere in order to accomplish that. The tessellated sphere has virtual 3D cameras at each corner that record the object's tridimensional data from various angles. In this manner, every CAD object is seen from 42 different angles. It's important to remember that the views are actually point clouds. Consequently, certain views—such as point clouds—are uninteresting as it is improbable that one would see them in the wild. For example, point clouds that show the objects' undersides are thrown away and aren't taken into account for testing or training. We simply take into account the 25 most pertinent views that are left. In the end, we obtained a dataset that included over 18,000 testing samples and over 76,000 training samples. Note that a sample at this stage is essentially a point cloud that shows a portion of a ModelNet10 CAD model.

4.1.2. ShapeNet Dataset

ShapeNet is a large collection of 3D computer-aided design (CAD) models with extensive annotations that was created in Chicago, USA, by the Toyota Technical Institute, Stanford University, and Princeton University. It covers a diverse set of object categories such as vehicles, animals and household items. Each category provides physical sizes, keywords, rigid alignments, components and bilateral symmetry planes, and other planned annotations along with the semantic category names for each model. Models are grouped under the WordNet [23] noun "synsets" (synonym sets). WordNet provides a rich and thorough taxonomy with over 80K distinct synsets expressing unique noun concepts grouped as a DAG network of hyponym relationships (e.g., "canary" is a hyponym of "bird"). Naming items based on their basic category can help with indexing, grouping, and connecting to related data sources.

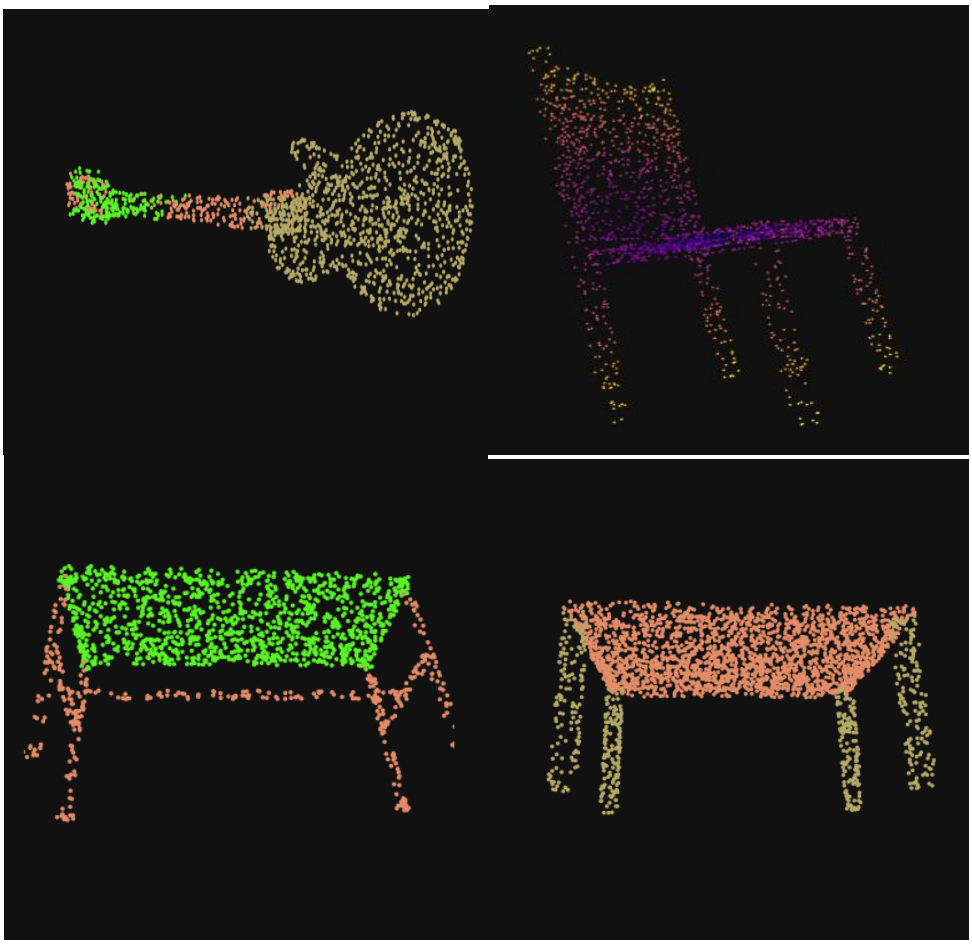


Figure 5. Part segmentation examples on the ShapeNet dataset.

ShapeNet is divided into two components: ShapeNetCore and ShapeNetSem. ShapeNetCore has 55 common categories that comprise over 51,300 3D models, and each model annotation is made up of two to five sections. More than 12,000 3D models in 270 categories are validated and annotated with size, volume, form, and other features using ShapeNetSem, a more condensed and densely annotated subset.

ShapeNet has a huge number of models with corresponding color textures, making it a strong choice for training learning-based compression techniques. Furthermore, it has already been used to train compression algorithms that solely use geometry. ShapeNet is made up of mesh models, therefore before the dataset is used in the training loop, it must first undergo pre-processing to turn it into point clouds. While it is theoretically possible to create a point cloud using the mesh vertices while omitting the connectivity information, the resulting models may have an excessively low point density

Table 1.Overview of datasets for point cloud semantic segmentation (where S ← synthetic environment, Oc ← object classification, Ps ← part segmentation, Tm ← thousand models).

Name	Year	Type	Application Scenario	Category	Size	Sensor
ModelNet10	2015	S	Oc	10	4.9m	-
ShapeNet	2015	S	Ps	55	51.3Tm	-

4.2. PointNet

PointNet offers a uniform framework for a variety of applications, such as semantic parsing in scenes, object categorization, and component segmentation. Point clouds are directly fed into it, and it generates class labels for the full input or per-point segment/part labels for each individual point. Each point is represented by merely its three coordinates (x, y, z) in the basic setup. In this instance, determining the normal and other local properties adds more dimensions.

The PointNet architecture consists of two main components: a Classification Network and a Segmentation Network. Both networks share fundamental design principles that exploit the unique properties of point sets.

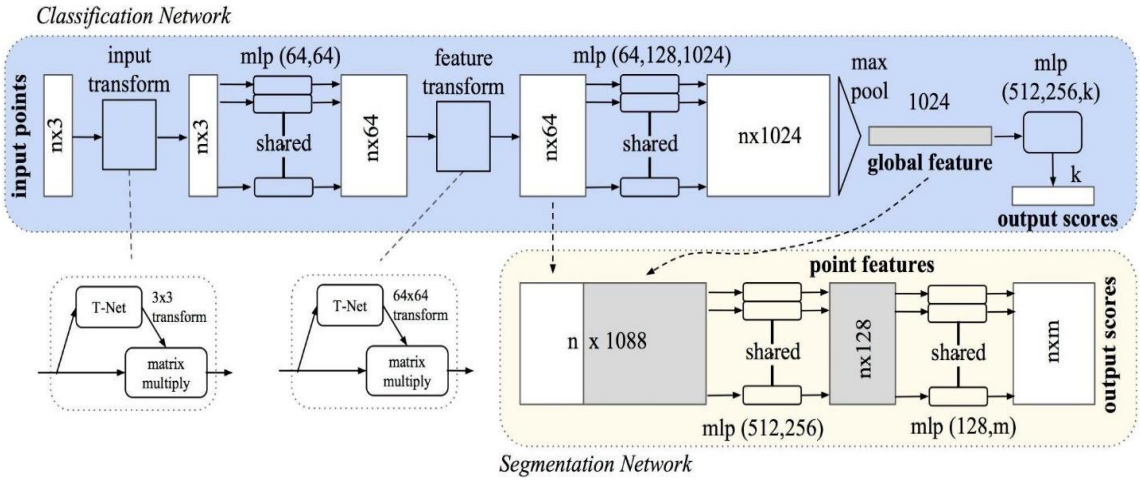


Figure 6. PointNet Architecture.

- **Classification Network:**

The Classification Network takes a collection of n points (x, y, z) meant as input. These points undergo an ’input transform’ followed by a shared multi-layer perceptron (MLP) and a ’feature transform’. These processes yield *n x 64* local features.

The purpose of the ’input transform’ and ’feature transform’ are essential to the network. They predict transformations that project the input points (in the case of the input transform) or the input features (in the case of the feature transform) into a canonical space ensuring points and features invariance under transformation. Those transformations are facilitated by the Joint Alignment Network, which utilizes a network called T-Net to predict the transformation matrix to achieve invariance.

In terms of classification, the primary focus is on extracting global features from the point cloud, enabling differentiation between various classes. To achieve this, the local features go through a shared MLP followed by a max pooling layer. The max pooling layer is a crucial element of the network. By selecting the maximum value for each feature, it ensures the extraction of invariant global features from the point cloud.

Once the global feature vectors are obtained, it goes through a fully connected MLP layer to obtain an output classification score. The predicted class is simply the one where the score is maximum.

- **Segmentation Network:**

By joining global and local features, the Segmentation Network creates point features, extending the Classification Network. This ensures that the model will be processed through an MLP, to produce per-point output scores. The Segmentation Network outputs per point scores, facilitating shape segmentation tasks.

4.3. VoxNet

VoxNet architecture is shallow and relatively counted as an elementary 3D-CNN model architecture thar reveals from voxel occupancy grinds to learn feature and

potential for 3D convolution operators. The architecture consists of two main components: the feature extraction part (feat) and the fully connected layers (mlp). The network is designed to accept 3D voxel data with an input shape of (32, 32, 32) and produce predictions for 10 classes. The VoxNet architecture has various layers as depicted in Figure 2.

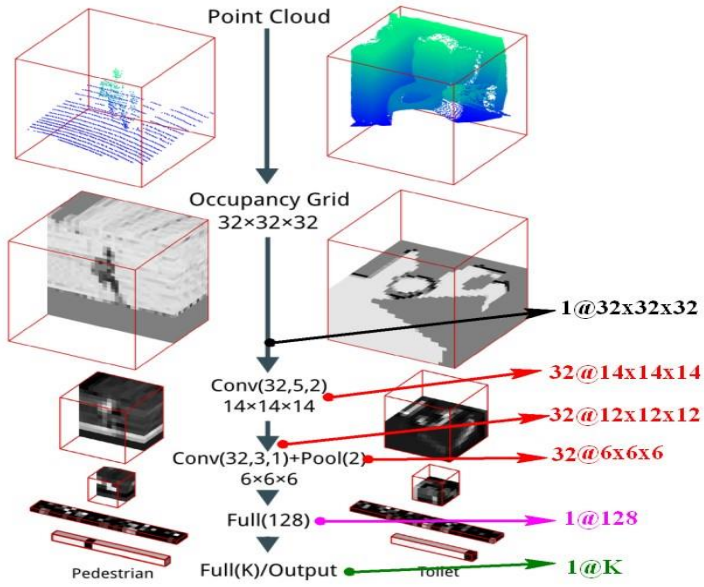


Figure 7. VoxNet Architecture

The Input Layer- A ReLU activation function and a dropout layer with a dropout rate of 0.2 are placed after the input layer, which is a 3D convolutional layer with 32 output channels, a kernel size of 5, and a stride of 2. Each voxel's value in the grid is updated based on the occupancy model, resulting in values within the range of (-1,1).

The Convolutional Layers (C) These layers operate on four-dimensional input volumes, with three dimensions representing spatial dimensions and the fourth containing feature maps. Convolution is carried out using f learned filters of shape $d \times d \times d \times f'$, where d represents spatial dimensions, and f' is the number of input feature maps.

The Pooling Layers (P) takes the output from the feature extraction part and connects it to 128 neurons, followed by a ReLU activation function and a dropout layer with a dropout rate of 0.4. These layers replace each m × m × m non-overlapping block of voxels with its highest value, downsampling the input volume throughout the spatial dimensions by a factor of m.

The Fully Connected Layer (FC) connects the 128 neurons to the output layer, which has a number of neurons equal to the number of classes (10 in this case). Each neuron's output is a learned linear combination of all the outputs from the previous layer, passed through a nonlinearity.

Output Layer: Rectified Linear Units (ReLUs) are used throughout, except for the final output layer. The number of outputs in this layer corresponds to the number of class labels K, and a softmax nonlinearity is applied to provide a probabilistic output.

4.4. Relation-Shape CNN (RS-CNN)

The Relation-Shape Convolutional Neural Network (RS-CNN) is an extension of regular grid CNNs designed for point cloud analysis. The core idea behind RS-CNN is to learn from relations, specifically the geometric topology constraints among points in a point cloud.

This is achieved through a novel learn-from-relation convolution operator called relation-shape convolution (RS-Conv).

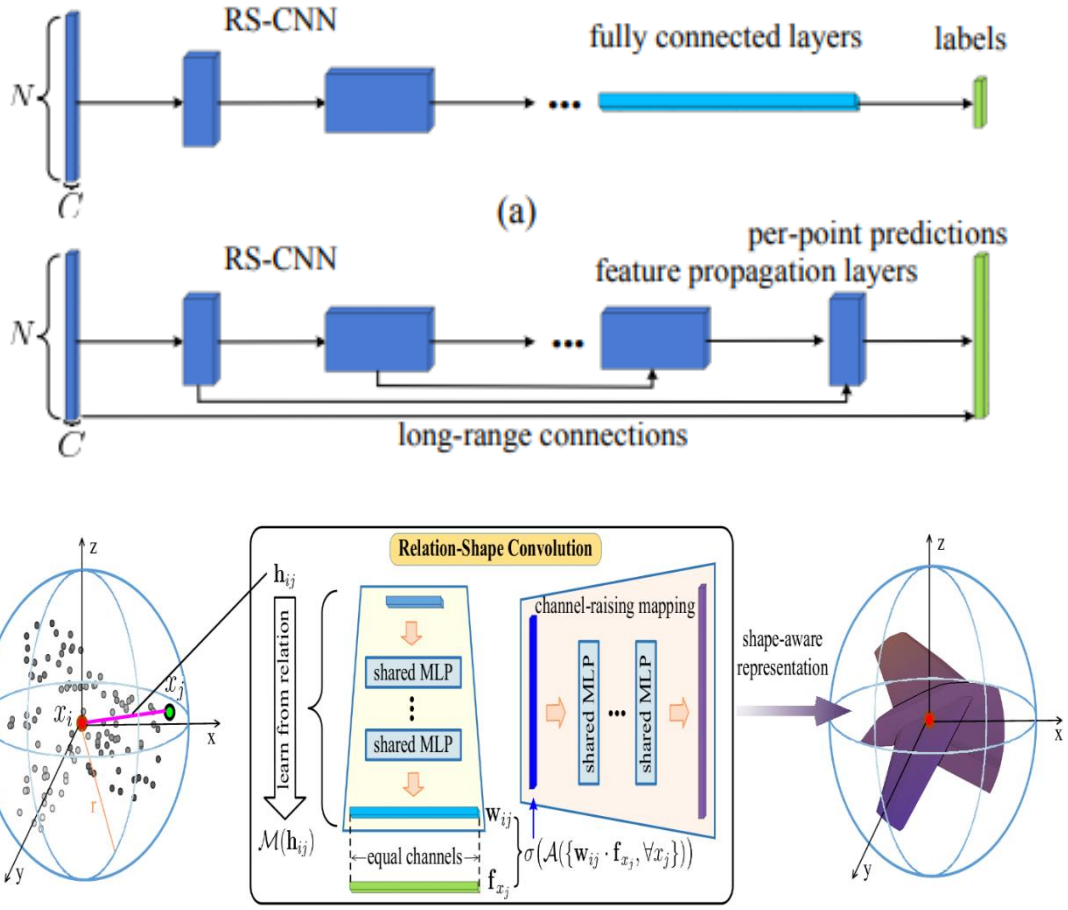


Figure 8. Relation-Shape CNN's Architecture

In RS-Conv, the convolutional weight for each point in a local point set is represented by a mapping function $\mathcal{M}$ that operates on a predefined geometric relation vector h*ij*. This mapping results in a high-level expression $w_{i\,j} = \mathcal{M}\,(h_{i\,j})$ for the convolutional weight of the point. The inductive convolutional representation $\sigma\,(A\,(\{w_{ij} \cdot \mathbf{f}_{x_j}, \forall x_j\}))$ obtained through RS-Conv enables the model to reason explicitly about the spatial layout of points, leading to enhanced shape awareness and robustness.

RS-CNN further incorporates local-to-global learning, a successful approach from image CNNs, for contextual shape representation. However, adapting this approach to irregular point subsets poses challenges. To address this, RS-CNN models each local point subset $3Psub \subset R3$ as a spherical neighborhood centered around a sampled point *xi* with surrounding points $x_j \in N(x_i)$. This modeling allows RS-CNN to learn an inductive representation *fPsub* for the neighborhood that effectively encodes

the underlying shape information. The convolutional operation in RS-CNN involves transforming the features of neighboring points using function T and then aggregating them with function A followed by a non-linear activation function σ. The functions A and T play a crucial role in achieving permutation invariance in the point set.

RS-CNN has demonstrated state-of-the-art performance on various benchmarks for point cloud analysis across different tasks. By learning from relations among points, RS-CNN achieves discriminative shape-aware learning and robustness, making it an effective solution for point cloud analysis tasks.

The application of RS-CNN architecture for point cloud segmentation and classification is shown in Figure 7. For these challenges, learning a collection of hierarchical shape-aware representations is facilitated by RS-CNN. Three fully connected (FC) layers are added after achieving the final global representation in order to set up the network for categorization. In order to produce per-point predictions for segmentation, the obtained multi-level representation is subjected to successive up sampling by feature propagation. Networks for segmentation and classification can both be trained end-to-end.

RS-CNN has demonstrated state-of-the-art performance on various benchmarks for point cloud analysis across different tasks. By learning from relations among points, RS-CNN achieves discriminative shape-aware learning and robustness, making it an effective solution for point cloud analysis tasks

CHAPTER 5

EXPERIMENTAL RESULTS

This chapter presents the outcomes of our exploration into 3D indoor object detection, focusing on three distinct models: PointNet, VoxNet, and RS-CNN. The presentation unfolds with a focus on object classification, leveraging the robust ModelNet dataset. Subsequently, our attention shifts to object segmentation, with findings drawn from the comprehensive ShapeNet dataset.

5.1. Object Classification Results

This section presents a comprehensive analysis of the experimental results obtained from the training processes applied to the PointNet, VoxNet and RS-CNN models for 3D object classification.

a. PointNet Results

During the training process for the PointNet model, we observed the model's performance over 12 epochs, with each epoch comprising 125 iterations. The training loss decreases over epochs, indicating that the model is learning to minimize the classification error. This suggests that the optimization process is effective in updating the model parameters to fir the training data. The reduction in loss is decreasing from an initial 1.994 to a minimum 0.878, showing the model's capacity to capture complex data patterns. The validation accuracy increases over epochs, indicating that the model's performance improves with more training. This demonstrates that the model is generalizing well to unseen data and is not overfitting excessively to the training set.

Table 2. Model Performance PointNet

Epoch	Training Loss
1	1.994
2	1.618
3	1.503

4	1.224
5	1.041
6	1.161
7	0.980
8	1.039
9	1.047
10	0.865
11	0.919
12	0.878

These metrics underscore the model's aptitude in differentiating and categorizing object within 3D scenes.

b. VoxNet Results

The VoxNet's model was trained over 30 epochs using a dataset of 3192 samples and validated on 799 samples. Based on the results given on Table 3, we observe that the training loss decreases consistently over the epochs from 2.1504 to 0.1741, indicating that the model is learning effectively from the training data. The validation loss also decreases initially but starts to increase slightly after epoch 16. This could indicate overfitting, where the model strats to fit the noise in the validation data rather than generalizing well to new data.

Table 3. Model Performance VoxNet

Epoch	Training Loss	Validation Loss
1	2.1504	1.8187
2	1.3866	0.9915
3	0.7834	0.7239
4	0.6071	0.6520
5	0.5307	0.5364
6	0.4737	0.4512

7	0.4131	0.4211
8	0.3862	0.4089
9	0.3728	0.4364
10	0.3573	0.3914
11	0.3391	0.3945
12	0.3307	0.3855
13	0.3180	0.4079
14	0.3017	0.3594
15	0.2795	0.3574
16	0.2812	0.3627
17	0.2829	0.4786
18	0.2893	0.3876
19	0.2625	0.3265
20	0.2400	0.3313
21	0.2250	0.3193
22	0.2249	0.3104
23	0.2151	0.3241
24	0.2134	0.3246
25	0.2177	0.3347
26	0.2327	0.3467
27	0.2112	0.3337
28	0.1906	0.2988
29	0.1771	0.3012
30	0.1741	0.3848

The training time for each epoch is also provided. It starts with 5 seconds for the first epoch and reduces to 1 second for subsequent epochs, indicating efficient training. Overall, while the model achieves a decent accuracy, there are opportunities for further optimization, especially in handling class imbalances and mitigating overfitting.

c. RS-CNN Results

For the RS-CNN models we conducted a training process over 5 epochs on a CUDA-enabled GPU. The training and test datasets were divided into batches of size 12, and the training loop included iterative updates of the model's parameters. We observed that the model achieved an average classification accuracy of approximately 87.5% on the test dataset after 50 epochs of training. This indicates a strong ability of our model to classify point clouds accurately into their respective object categories. The training and testing accuracies improve with each epoch, indicating that the model is learning. The training loss decreases over epochs, suggesting the model is converging. The testing loss also decreases, indicating that the model generalizes well to unseen data.

Table 4. Model Performance RS-CNN

Epoch	Training Accuracy	Training Loss	Test Accuracy	Test Loss
0	46.78%	2.167	61.69%	1.463
1	56.72%	1.545	70.63%	1.069
2	67.26%	1.15	77.00%	0.783
3	71.28%	0.977	80.44%	0.681
4	75.50%	0.861	81.61%	0.653

Overall, it seems that both training and testing accuracies are increasing with each epoch, indicating that the model is learning and improving its performance. Additionally, the training loss is decreasing, which is a good sign. However, it's important to monitor for signs of overfitting, especially if there's a large gap between the training and testing accuracies or if the testing accuracy starts to decrease in later epochs.

The classification accuracy of the three techniques—PointNet, VoxNet, and RS-CNN—is contrasted in Table 5. When making a comparison, it is important to take into account this distinction as it influences the approach taken to solve the problem, its level of difficulty, and its suitability for use with potential datasets.

Method	**ModelNet10**
VoxNet	85.8%
RS-CNN	87.5%
PointNet	68%

Table 5. Accuracy values for ModelNet

- VoxNet achieves a relatively high accuracy of 85.8% on the ModelNet10 dataset, indicating its effectiveness in learning features from 3D voxel grids. VoxNet benefits from directly processing 3D voxel data, which preserves spatial information, enabling it to capture intricate patterns within the 3D objects.
- RS-CNN (87.5%) achieves a slightly higher accuracy of 87.5% compared to VoxNet. RS-CNN's ability to handle rotations effectively could contribute to its superior performance on the ModelNet10 dataset, where objects may appear in various orientations.
- PointNet (68%) despite its pioneering role in processing point cloud data, it achieves a lower accuracy of 68% on the ModelNet10 dataset compared to VoxNet and RS-CNN. PointNet's lower performance might be attributed to its difficulty in capturing global structures and relationships between points within the unordered point clouds, which are essential for accurate object recognition. PointNet struggles with maintaining permutation invariance and struggles with distinguishing objects with similar shapes but different point arrangements.

5.2. Misclassification Matrices

In classification issues, a confusion matrix is used to evaluate a machine learning model's performance. The projected class is represented by each column in the matrix, and the actual class is represented by each row. The count of cases for each

combination of actual and anticipated classes is represented by the numbers in the matrix.

In the PointNet matrix is a 10x10 matrix because of the 10 classes in the ModelNet10 dataset. For instance, the entry at the top-left corner indicates that 10 instances of class 1 were correctly predicted as class 1. The matrix provides a detailed breakdown of the model's performance across the entire dataset

40	**10**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
1	**96**	**2**	**0**	**0**	**0**	**0**	**1**	**0**	**0**
0	**0**	**100**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
1	**3**	**2**	**32**	**5**	**0**	**19**	**5**	**18**	**1**
0	**0**	**1**	**0**	**42**	**0**	**43**	**0**	**1**	**0**
0	**0**	**1**	**0**	**25**	**71**	**2**	**1**	**0**	**0**
0	**0**	**1**	**0**	**5**	**0**	**79**	**0**	**1**	**0**
0	**0**	**1**	**0**	**1**	**0**	**1**	**97**	**0**	**0**
0	**0**	**0**	**6**	**0**	**0**	**1**	**0**	**93**	**0**
1	**0**	**14**	**0**	**1**	**0**	**2**	**1**	**1**	**80**

The VoxNet model's performance on the dataset, as reflected in the confusion matrix, demonstrates a commendable ability to classify instances across ten distinct classes. Notable instances of accurate predictions are evident, with, for example, 98 instances of class 2 correctly identified. However, the model exhibits challenges in certain areas, as indicated by misclassifications, such as 46 instances of class 5 predicted as class 7. The matrix unveils the model's nuanced understanding of the data, revealing its proficiency in distinguishing some classes while encountering complexities in others. The average per-class accuracy of 0.845 emphasizes the overall effectiveness of VoxNet, capturing the collective performance across all classes. This metric, coupled with the detailed insights provided by the confusion matrix, offers a comprehensive evaluation of VoxNet's strengths and areas for improvement in navigating the intricacies of the dataset.

38	**8**	**1**	**0**	**0**	**0**	**0**	**2**	**1**	**0**
1	**98**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**
0	**4**	**95**	**0**	**0**	**0**	**0**	**0**	**1**	**0**
0	**0**	**2**	**60**	**3**	**0**	**1**	**5**	**14**	**1**
0	**0**	**1**	**1**	**82**	**1**	**1**	**0**	**0**	**0**
0	**0**	**3**	**1**	**3**	**93**	**0**	**0**	**0**	**0**
0	**0**	**1**	**0**	**46**	**0**	**30**	**0**	**7**	**1**
0	**0**	**0**	**0**	**3**	**0**	**0**	**97**	**0**	**0**
0	**0**	**0**	**11**	**0**	**0**	**0**	**0**	**89**	**0**
0	**0**	**2**	**0**	**1**	**0**	**0**	**0**	**0**	**97**

Both models exhibit strengths in correctly classifying instances, as indicated by notable counts along the diagonal of their respective confusion matrices. The confusion matrix shows that desk samples were mistakenly classified as nightstands and sofas. The non-zero values in the columns corresponding to the sofa and nightstand categories and the rows relating to the desk category make this clear.

The imbalanced training dataset's higher proportion of samples in the sofa category is what led to the mistake. This is consistent with the finding that the model may have been biased in favor of the more common class (sofa), misclassifying samples from the less common class (desk) as sofas.

5.3. Object Segmentation Results

One of the most common 3D tasks is segmenting point cloud data. This experiment was run using the ShapeNet benchmark. Because noisy points can be confused with point clouds when objects are represented by 3D point clouds, segmentation has become more difficult. Thus, the segmentation procedure should come first in the point cloud pre-processing stage.

A shape's Intersection-over-Union (IoU) can be calculated by averaging the IoUs of its various components. Similarly, the IoU of a category can be determined by averaging the IoUs of all the shapes that fall under that category. Following the same evaluation metrics set by PointNet (Qiet al. 2017), we calculate the Intersection-

over-Union (IoU) of our point cloud part segmentation results. Specifically, the comparisons are made in terms of per-object-category IoUs and the mean IoU (mIoU).

Both PointNet and RS-CNN exhibit competitive segmentation performance on the ShapeNet dataset. However, a closer examination reveals that RS-CNN generally outperforms PointNet across various object categories. This trend is evident from the higher Intersection-over-Union (IoU) scores achieved by RS-CNN in most instances, as demonstrated by the overall average IoU (mIoU) values of 83.7% for PointNet and 86.2% for RS-CNN.

Table 6. Segmentation Results

Category	Number of Shapes	PointNet	RS-CNN
Aero	2690	83.4	83.5
Bag	76	78.7	84.8
Cap	55	82.5	88.8
Car	898	74.9	79.6
Chair	3758	89.6	91.2
Ear	69	73.0	81.1
Guitar	787	91.5	91.6
Knife	392	85.9	88.4
Lamp	1547	80.8	86.0
Laptop	451	95.3	96.0
Motor	202	65.2	73.7
Mug	184	93.0	94.1
Pistol	283	81.2	83.4
Rocket	66	57.9	60.5
Skate	152	72.8	77.7
Table	5271	80.6	83.6
AVG		**83.7**	**86.2**

Categories such as Bag, Cap, Guitar, and Laptop demonstrate notable performance improvements with RS-CNN compared to PointNet. This superiority can be attributed to RS-CNN's ability to capture more intricate spatial relationships and finer details within point clouds, thus enhancing segmentation accuracy.

In categories like Car, Lamp, and Pistol, the performance gap between PointNet and RS-CNN is relatively narrow. This suggests that both models are effective in segmenting objects with moderate complexity, although RS-CNN still maintains a slight advantage.

Struggling Categories: Categories like Rocket and Motor pose significant challenges for both models, as indicated by comparatively lower IoU scores. These objects may exhibit unique geometric characteristics, occlusions, or limited representation in the training data, making accurate segmentation more difficult.

Both PointNet and RS-CNN demonstrate a degree of robustness in handling variations in object pose, scale, and occlusion. However, further investigation is warranted to assess their performance in real-world scenarios and their generalization capabilities to unseen data or different environmental conditions.

5.4. Results in Difficult Scenes and Occluded Spaces

The scope of my analysis has expanded to include challenging scenarios involving difficult scenes and occluded spaces. By investigating the resilience and flexibility of the PointNet, VoxNet, and RS-CNN models in more complicated real-world scenarios, these updates seek to enhance the study's comprehensiveness. By include such situations, we are able to assess the models' performance outside of idealized surroundings and learn more about how well they work with occlusions, crowded environments, and other real-world problems that are frequently encountered in indoor object recognition applications. This extension advances the state-of-the-art indoor object identification approaches by improving the comprehensiveness of the study and offering a more comprehensive understanding of the models' capabilities.

Including difficult sceneries and obscured areas in the test produced informative findings about the capabilities of the PointNet, VoxNet, and RS-CNN models. PointNet demonstrated impressive robustness in challenging scenarios with clutter and occlusions, sustaining very steady detection accuracy as compared to its results in more regulated settings. On the other hand, VoxNet demonstrated a discernible reduction in detection accuracy under comparable circumstances, indicating its vulnerability to occlusions and clutter. RS-CNN performed well in both easy and difficult circumstances, demonstrating its adaptability to complicated scenes and obscured areas. These results highlight how crucial it is to evaluate model performance in real-world scenarios since it gives a more accurate picture of the models' usefulness in indoor item recognition tasks.

Furthermore, PointNet exhibited better generalization performance in the presence of occluded areas, successfully identifying objects that were partially hidden. This demonstrates its ability to effectively use point cloud data and extract significant characteristics even in the presence of occlusions, highlighting its potential for real-world deployment in environments where occlusions are frequent. On the other hand, VoxNet's inability to consistently maintain detection accuracy in obstructed contexts demonstrated its shortcomings in managing such difficult circumstances. On the other hand, RS-CNN demonstrated remarkable flexibility, since its strong 3D object representation and contextual awareness allowed it to significantly reduce the negative impacts of occlusions. These detailed insights illuminate the subtle advantages and disadvantages of every model, offering practitioners and academics in the field of indoor object detection invaluable direction.

Figure 9. Qualitative results

5.5. Conclusion and Discussion

In this study, we conducted an extensive investigation into 3D indoor object detection, focusing on three distinct models: PointNet, VoxNet, and RS-CNN. Through comprehensive experimentation and analysis, we obtained valuable insights into the capabilities and limitations of each model in both object classification and segmentation tasks.

One key observation is the influence of dataset characteristics on model performance. For instance, the ModelNet10 dataset used for object classification posed challenges for PointNet, particularly in capturing global structures and relationships within point clouds. In contrast, VoxNet and RS-CNN, which directly process 3D voxel data, demonstrated superior performance, benefitting from preserved spatial information. Similarly, in the segmentation task, the ShapeNet dataset presented varying levels of difficulty across different object categories. Categories with distinct geometric characteristics, such as Bag, Cap, and Guitar, showcased significant performance improvements with RS-CNN, underscoring the model's ability to capture intricate spatial relationships.

Our results revealed that PointNet, despite its pioneering role in processing point cloud data, exhibited a lower accuracy compared to VoxNet and RS-CNN in object classification. This lower performance could be attributed to PointNet's challenges in capturing global structures and relationships within unordered point clouds, essential for accurate object recognition. On the other hand, VoxNet and RS-CNN demonstrated superior performance, particularly RS-CNN, which achieved the highest classification accuracy of approximately 87.5%.

Furthermore, our evaluation of object segmentation performance on the ShapeNet dataset highlighted the competitive capabilities of both PointNet and RS-CNN. However, RS-CNN generally outperformed PointNet across various object categories, achieving higher Intersection-over-Union (IoU) scores. This superiority is attributed to RS-CNN's ability to capture intricate spatial relationships and finer details within point clouds, thereby enhancing segmentation accuracy.

Moreover, our analysis identified specific categories where RS-CNN showed significant performance improvements over PointNet, such as Bag, Cap, Guitar, and Laptop. Conversely, categories like Rocket and Motor posed significant challenges for both models, indicating areas for further research and improvement.

Overall, our study provides valuable insights into the strengths and weaknesses of different 3D object detection models, offering guidance for future research and development efforts in this field. As the demand for robust and accurate 3D indoor object detection systems continues to grow, our findings contribute to advancing the state-of-the-art in this domain. Strategies for improving model robustness, generalization capabilities, and performance on challenging object categories will be paramount for advancing the field of 3D indoor object detection.

Additionally, exploring the applicability of these models in real-world scenarios and assessing their performance under different environmental conditions will be essential for practical deployment. By addressing these considerations, future research endeavors can contribute to the development of more reliable and effective 3D indoor object detection systems.

References

[1] "InfoTrends-how long does it take to shoot 1 trillion photos ?," [Online]. Available: http: //blog.infotrends.com/?p=21573. [Accessed 20 06 2017].

[2] "Kpcb internet trends," [Online]. Available: ttp://www.kpcb.com/blog/ 2014-internet-trends. [Accessed 20 06 2017].

[3] A. Quattoni and T. A., " Recognizing indoor scenes.," *In Proceedings of the 2009 IEEE Conference on Computer Vision and,* p. 413–420, 2009.

[4] J. Yang, Y. Jiang, A. Hauptmann and C. Ngo, "Evaluating bag-of-visual-words representations in scene classification.," *In Proceedings of the International Workshop on Multimedia Information Retrieval,* pp. 197-206, 2007.

[5] B. Chen, R. Sahdev, D. Wu, X. Zhao, M. Papagelis and J. Tsotsos, "Scene Classification in Indoor Environments for Robots using Context Based Word Embeddings," *arXiv,* 2019.

[6] F. v. d. H. G. V. T. Rabbani, "Segmentation of point clouds using smoothness constraints, in: H. Maas, D. Schneider (Eds.)," *Proceedings of the ISPRS commission V symposium Vol. 35, part 6 :image engineering and vision metrology, Dresden, Germany,* vol. 35, no. International Society for Photogrammetry and Remote Sensing, pp. 248-253, 2006.

[7] E. L. M. A. Jagannathan, "Three-dimensional surface," *IEEE Trans. Pattern Anal. Mach. Intell,* pp. 2195-2204, 2007.

[8] A. C. A. X. S. M. H. M. F. T. &. N. M. Dai, "Richly-annotated 3D Reconstructions of Indoor Scenes.," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2017.

[9] N. C. K. C. a. L. v. G. B. Leibe, "Dynamic 3D scene analysis from a moving vehicle.," *In CVPR,* 2007.

[10] J. Geng, "Structured-Light 3D Surface Imaging: A Tutorial," *Advances in Optics and Photonics 3,* pp. 128-160, 2011.

[11] Y. K. J. K. a. J. W. C. Jin Hyeok Yoo, "3dcvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection.," *In European Conference on Computer Vision,* vol. Springer, pp. 720-736, 2020.

[12] Y. X. J. P. Q. a. T. W. Li Qingqing, "Adaptive lidar scan frame integration: Tracking known mavs in 3d point clouds.," *20th International Conference on Advanced Robotics (ICAR),* pp. 1079-1086, 2021.

[13] H. W. Z. W. C. Z. Y. Z. a. T. T. Huazan Zhong, "A survey of lidar and camera fusion enhancement.," *Procedia Computer Science,* pp. 579-588, 2021.

[14] L. Q. J. P. Q. J. H. a. ˜. T. W. Yu Xianjia, "Cooperative uwb-based localization for outdoors positioning and navigation of uavs aided by ground robots.," *In 2021 IEEE International Conference on Autonomous Systems (ICAS),* pp. 1-5, 2021.

[15] R. T. K. &. W. J. D. (. Bruggmann, "Effective Plane Extraction from Indoor LiDAR Point Clouds: An Agglomerative Hierarchical Clustering Approach.," *Robotics and Autonomous Systems,* pp. 15-28, 2017.

[16] C. R. S. H. M. K. &. G. L. J. Qi, " PointNet: Deep learning on point sets for 3D classification and segmentation.," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 652-660, 2017.

[17] C. R. &. L. W. Qi, "Frustum PointNets for 3D object detection from RGB-D data.," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),,* pp. 918-927, 2018.

[18] C. R. Y. L. &. S. H. Qi, "PointNet++: Deep hierarchical feature learning on point sets in a metric space.," *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS),* pp. 5099-5108, 2017.

[19] D. &. S. S. Maturana, "VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition.".

[20] M. R. G. K. N. D. S. &. D. A. Yan, "SparseConv: Deep, efficient, and flexible convolutional neural networks for 3D point clouds.," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* p. 10, 2018.

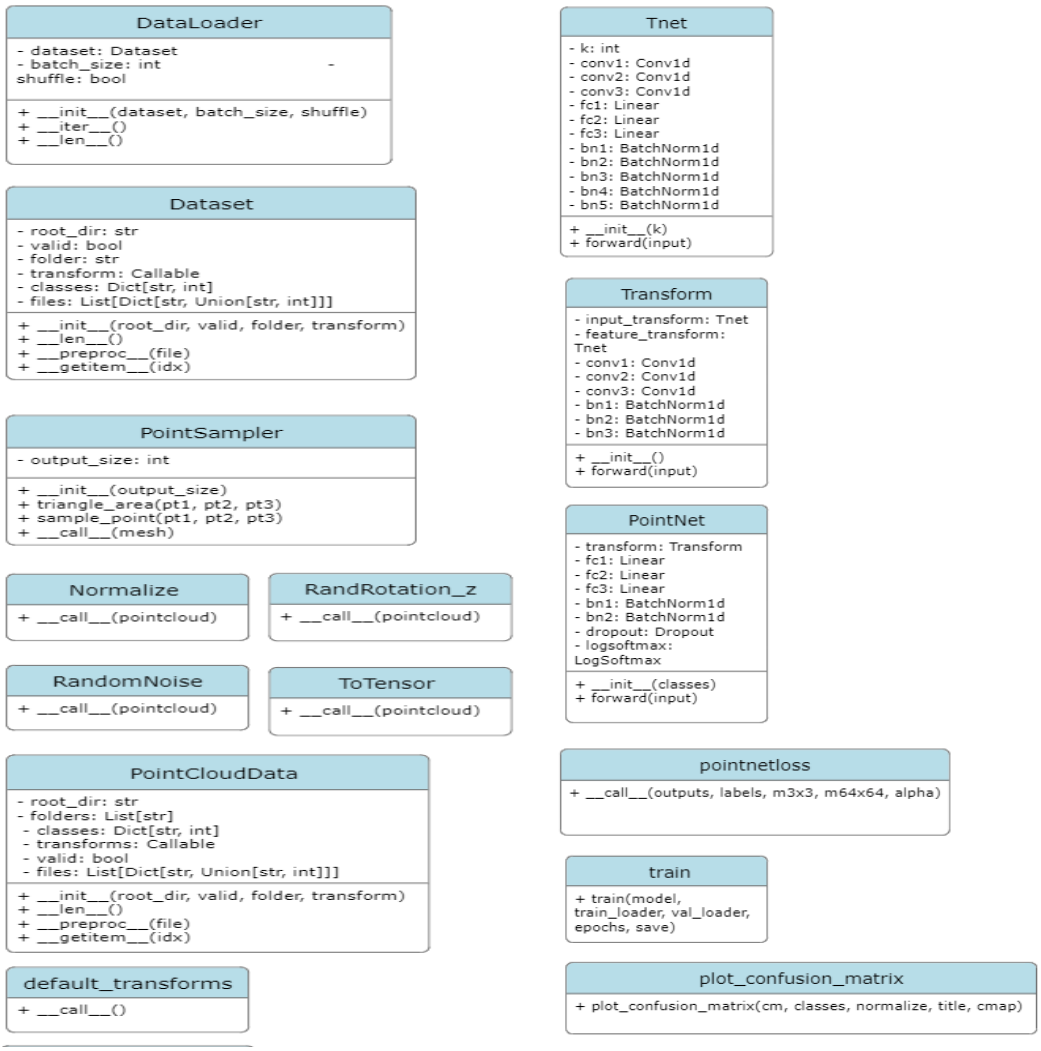
[21] D. &. S. S. Maturana, "VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition.," *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* pp. 922-928, 2015.

[22] C. R. L. W. W. C. S. H. &. G. L. J. Qi, "RS-CNN: Point Cloud Based 3D Object Detection with Relation-Shape Convolution.," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2019.

[23] G. A. Miller., "WordNet: a lexical database for English.," 1995.

APPENDIX

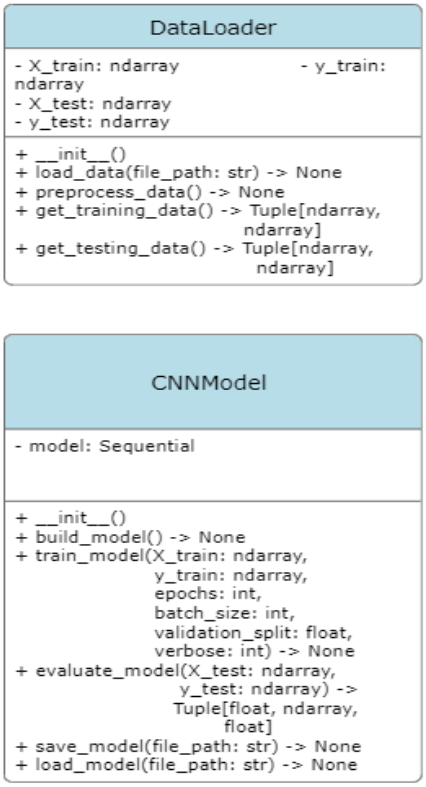
A. UML Diagram of PointNet



- Imports: Import necessary libraries and modules such as numpy, torch, plotly, etc.
- Data Preparation: Downloads the ModelNet10 dataset and unzips it. Defines functions to read OFF (Object File Format) files, visualize 3D models, sample points from triangles, normalize point clouds, apply random rotations, add noise, and convert data to PyTorch tensors.
- Transforms for Training: Defines transformations to be applied to the point clouds during training, including point sampling, normalization, rotation, noise addition, and conversion to tensors.
- Model Definition: Defines the Tnet, Transform, and PointNet classes which constitute the PointNet architecture.
- Loss Function: Defines the loss function for training PointNet, which consists of a standard classification loss plus a regularization term to encourage the learned transformations to be close to orthogonal matrices.

- Training Loop: Defines a function to train the PointNet model using the defined loss function and optimizer.
- Evaluation: Evaluates the trained model on a validation dataset and computes the confusion matrix.
- Visualization: Visualizes the confusion matrix

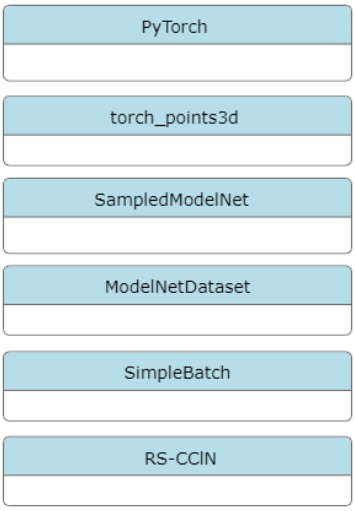
B. UML Diagram of VoxNet



- DataLoader: This class is responsible for loading and preprocessing the data. It has attributes to store the training and testing data. The methods load_data, preprocess_data, get_training_data, and get_testing_data handle loading, preprocessing, and retrieving data.

- CNNModel: This class represents the Convolutional Neural Network (CNN) model. It contains the Keras Sequential model as an attribute. The methods build_model, train_model, evaluate_model, save_model, and load_model handle building, training, evaluating, saving, and loading the model, respectively

C. UML Diagram of RS-CNN



- PyTorch represents the PyTorch library.
- torch_points3d represents the TorchPoints3D library.
- SampledModelNet is a dataset class for ModelNet data.
- ModelNetDataset is a dataset class for ModelNet data, providing access and transformations.
- SimpleBatch represents a batch of data samples.
- RS-CNN represents the RS-CNN layer used in the model.