Scene Classification using 3D Object Detection and Spatial Relations

Mingshi Chi 1004881096

Supervised by Professor John K. Tsotsos

April 2023

B.A.Sc. Thesis

Division of Engineering Science
UNIVERSITY OF TORONTO

Abstract

Scene classification is an essential task in computer vision that involves categorizing an image into one of several predefined scene categories. Previous works for scene classification which global feature or topological representations or with the immediate representations of objects. One important use case in scene classification is in the context of service robotics that may need knowledge of the objects and their spatial relations in scenes for search and retrieval tasks. Recent popular approaches to scene classification may not capture the spatial relationships between objects in the scene. We propose a method to utilize the 3D spatial relationships between objects to better represent and classify scenes. This approach involves detecting 3D objects in the scene using a Frustum PointNets based approach with a fine-tuned model on the SUN RGB-D dataset and encoding their spatial relationships in a 3D metric space. The encoded spatial relationships are in a sequential order for object-to-object pair relations and are then used to predict the scene category by a Recurrent Neural Network (RNN). We report the results on both 3D object detection and scene classification against current state-of-the-art methods tested on the SUN RGB-D dataset. This scene classification method is additionally deployed on mobile video feeds from a stereo depth camera on a mobile robotic with high accuracy results to verify the effectiveness in real world environments.

Acknowledgements

I would like to express my deepest gratitude to my thesis advisor, Professor John Tsotsos, for his guidance, expertise, and unwavering support throughout my research journey. Their insightful feedback and encouragement have been instrumental in the success of my work. I would also like to thank my fellow students and friends who have offered valuable feedback and support throughout the process.

I would like to extend my heartfelt thanks to Bikhram Bir Dey, who generously gave their time and played a paramount role in the implementation of the wheelchair mobile base and Markus Solbach who, with his expertise, gave insightful advice during the research process. I would like to thank York University and the Active Vision Lab for providing space for robotics development. Finally, I would like to thank my roommates Saiyam and Cameron for their constant feedback and support throughout this process. This thesis would not have been possible without the collective support and encouragement of these individuals, and I am deeply grateful for their contributions.

Contents

List of Figures

List of Tables

1 Introduction

Imagine a common situation: you are about to leave your home for a day at work. You have your lunch packed, jacket on, but you realise your keys are not with you. You check your pockets, and the key dish beside the door. Quickly, you look through all the rooms and surfaces where your keys might be (bedroom counter, living room coffee table, etc) and finally find them on the kitchen table.

Breaking down this scenario into simple steps, first the initial goal was to locate the keys. Second, you knew that the keys are likely in your pocket or in the key dish. You search these places first as they are immediate and obvious first areas to search. Finally, when realising the keys were not there, you search for the keys in the various different rooms and respective surfaces to solve your dilemma. In this context, you understood in which rooms the keys most likely reside such as living room and less likely to appear in a boiler room. We also know the other objects they will most likely appear with such as on top of a table. We know that the table's function is to hold things above it and such things include keys.

This series of tasks may seem very simple to us but has many complexities when broken down. Knowing prior knowledge of scenes the target object is most likely to appear and then categorizing visual scenes quickly and robustly is critical in search and retrieval and deciding how to act in a given context for both humans and robots. Understanding scene perception is an important research topic in both biological and machine vision. Commonly, visual scenes such as a room typically contain a large number of items arranged according to semantics and syntactic regularities. Mostly, humans are able to classify scenes by recognizing the objects and functions of objects within an environment[1]. Scenes contain objects which are spatially distinct entities that can be moved as well as areas that are cannot move such as walls, floors, and doors[2].

Indirect search is a concept that was first introduced by Thomas Garvey in 1976, as documented in his publication [3]. This idea suggests that when searching for something, it can be advantageous to have knowledge about the context in which the object is typically found. By utilizing contextual information, the search space can be pruned, and the search process can become more efficient. In particular, having knowledge about scenes that serve as priors for object occurrences can be highly beneficial in advanced service robotics tasks.

In such applications, knowing about common object occurrences in specific contexts can assist with search and retrieval tasks. For example, if a robot is searching for a specific tool, the robot would know the most likely place a tool may be is in a workshop. It may also be helpful for it to know where such tools are typically stored or used in that environment. Such knowledge can help the robot to navigate and search the workshop more efficiently, ultimately improving its performance. Mobile robots should be able to recognise the type of scene they are in by understanding the meaning of places, objects, and relationships between objects despite possible dynamic changes to the placement of items to be able to perform higher level autonomous tasks. Learning the co-occurrence and spatial relationships between objects in a scene and objects and their scene is a step towards reaching the goal of understanding functions of items and the relationship between functions and scenes.

Although sometimes used interchangeably, an important distinction to be made is between scene classification and scene recognition. Scene classification is often used in robotics and aims to categorize a scene image into predefined scene categories based on prior knowledge from training[4]. Scene recognition aims to identify the exact location or scene depicted in an image[5] and is often used in robotic navigation and mapping. Both are important in robotic applications such as Simultaneous Localization and Mapping (SLAM) algorithms. While exploring a new environment, a robot may use scene classification to identify types of scenes it finds itself in and maps the scene and scene category to its memory while later using scene recognition for localization. I will be discussing scene classification for the remainder of this document.

Currently, many methods in mobile robotic scene classification rely on pixel-based statistics and overall appearance or topology of the environment with depth data being recently used to enhance these methods[6]. These approaches are limited in their understanding of the objects that fill each location and their functions in relation to their environment. When objects are considered in scene classification, they are often an addition to other approaches[6]. Solely object based approaches are uncommon as they are heavily dependant on the reliability of the object detector[7]. Additionally, very few scene classification algorithms have been tested on real world environments using mobile robotics.

Thus, the main objectives of my thesis is to explore the gap of understanding the relationships between objects and their scene in the computer vision and robotics field. To

do this, this thesis work develops and implements a stereo vision based scene classification pipeline on an indoor mobile robot that is able to categorize scenes based on the objects present, their co-occurrence, and spatial context, first proposed by Thomas D Garvey[3], and to utilize depth data to improve classification accuracy. To meet this goal, there are three main sub-objectives:

1. **Classify objects in a scene using RGB-D data.**

   In service robotics, being able to classify scenes without being restricted by viewpoint is crucial for mobility. Changes in camera viewpoint make it difficult to accurately recognize and classify scenes using 2D data alone. The integration of 3D spatial metric data provides a means of maintaining the consistency of object relationships, irrespective of the observer's position. This facilitates more dependable scene recognition and classification by service robots, enabling them to navigate their environments with greater ease. The use of 3D object relations becomes especially critical when service robots are required to explore a given area in search of a specific object, where knowledge of the 3D spatial relationship between objects is necessary for effective path planning and navigation.

   We prefer 3D object detection over 2D object detection as to take advantage of the depth and structural data we have on objects. The utilization of point clouds as a means of representing this information is particularly advantageous, given that it preserves the surface details of the objects in question. This feature is especially useful in tasks associated with object classification, where the retention of fine-grained details is instrumental in achieving high levels of accuracy. In contrast, other data representation methods such as voxels fail to preserve such details, rendering them less effective in these tasks.

   The efficient processing of point clouds poses a significant computational challenge, given their large size. To address this issue, the Frustum PointNets method has been established as a highly effective approach for performing 3D object detection from RGB-D data [8]. PointNets, which are deep neural networks used for processing point clouds, transform such data into feature vectors that facilitate the efficient processing of unstructured data. The versatility of PointNets has been demonstrated in various

applications, including object segmentation and detection [9]. The Frustum PointNet technique employs a two-stage process involving the generation of a set of 3D object proposals from a 2D object detector, followed by the utilization of 2D image and 3D frustum data to determine the object location and box estimation. By reducing the amount of data processing required through the use of frustums, the proposal network is highly efficient. On the SUN RGB-D dataset, Frustum PointNet has achieved a high level of accuracy, with a precision of 75% and box intersection over union (IoU) of 0.7[8]. The SUN RGB-D dataset, which is a widely used benchmark dataset for 3D object detection and depth scene classification research, will be preprocessed and used to train a 3D object classifier in this study [10].

2. **Train a classifier to learn the relationships between objects themselves and between objects and scenes.**

   Following the identification of objects within a given scene, the next step involves the utilization of an appropriate method for representing this data in a manner that facilitates the learning of object relationships. While the visual bag of words vector method has proven to be highly effective for image classification tasks, it is limited in its ability to extract high-level semantic information from images, given that it primarily focuses on local feature vectors. To overcome this limitation, the proposed methodology for this study involves the implementation of a Sequential Object-to-Object Relation scene encoding approach. This approach incorporates several key features, including object classes, object sizes, object positions, and spatial relations between pairs of objects, thereby encompassing a broader range of semantic information. By leveraging these features, it is anticipated that the proposed method will enable more robust and effective learning of object relationships in scenes.

   By using objects as features, the classification algorithm can capture the spatial arrangement of objects and their relationships, which can provide rich information for accurate classification. The Sequential Object-to-Object Relations (SOOR) encoding method, as proposed by Song et al. [11], represents a notable advancement in this area, enabling the capture of spatial relations between objects. Building upon this existing methodology, the present study seeks to extend the SOOR approach to encompass 3D

object-to-object relations, thereby enhancing its capacity to represent spatial relationships in more complex, three-dimensional contexts. By leveraging the SOOR framework to capture these additional dimensions of spatial information, it is anticipated that the proposed methodology will yield more accurate classification results based solely on the objects present and their spatial location, particularly in contexts where object relationships play a critical role in accurate scene interpretation.

3. **Implement the pipeline on a mobile robot base.**

   Validation of the proposed pipeline's robustness, computational efficiency, and overall feasibility in a real-world setting represents a critical aspect of this research endeavor. Specifically, implementation of the pipeline on a mobile robotic platform offers a valuable means of testing the algorithm's capacity to function robustly in diverse point-of-view scenarios, which can be a challenging task in place classification tasks. Given that real-time computation is a crucial consideration in the design of viable algorithms for practical applications, the success of the proposed pipeline on a mobile robotic platform would represent a significant milestone. To this end, the scene classification pipeline will be implemented on a mobile service wheelchair that utilizes a stereo depth camera, which will serve as a valuable experiment for the development of future complex service tasks. Computation on the mobile platform will be facilitated by a single laptop equipped with a GPU, which represents the primary real-time constraint for the algorithm in question. By leveraging this mobile platform to test the proposed methodology, it is anticipated that valuable insights will be gained regarding the algorithm's overall performance, computational efficiency, and real-world feasibility.

2 Literature Review

To aid with this discussion, it is beneficial to provide an overview of how modern scene classification pipelines work work as well as methods of object detection and implementation. This section highlights the prevailing methods of object detection and scene classification and implementations of scene classification. Problems and benefits of each approach will be addressed for an overall analysis on previous designs for each each portion of the desired pipeline. Figure 1 below illustrates a high-level representation of previous scene classification works as well as the areas in which this thesis will contribute to.

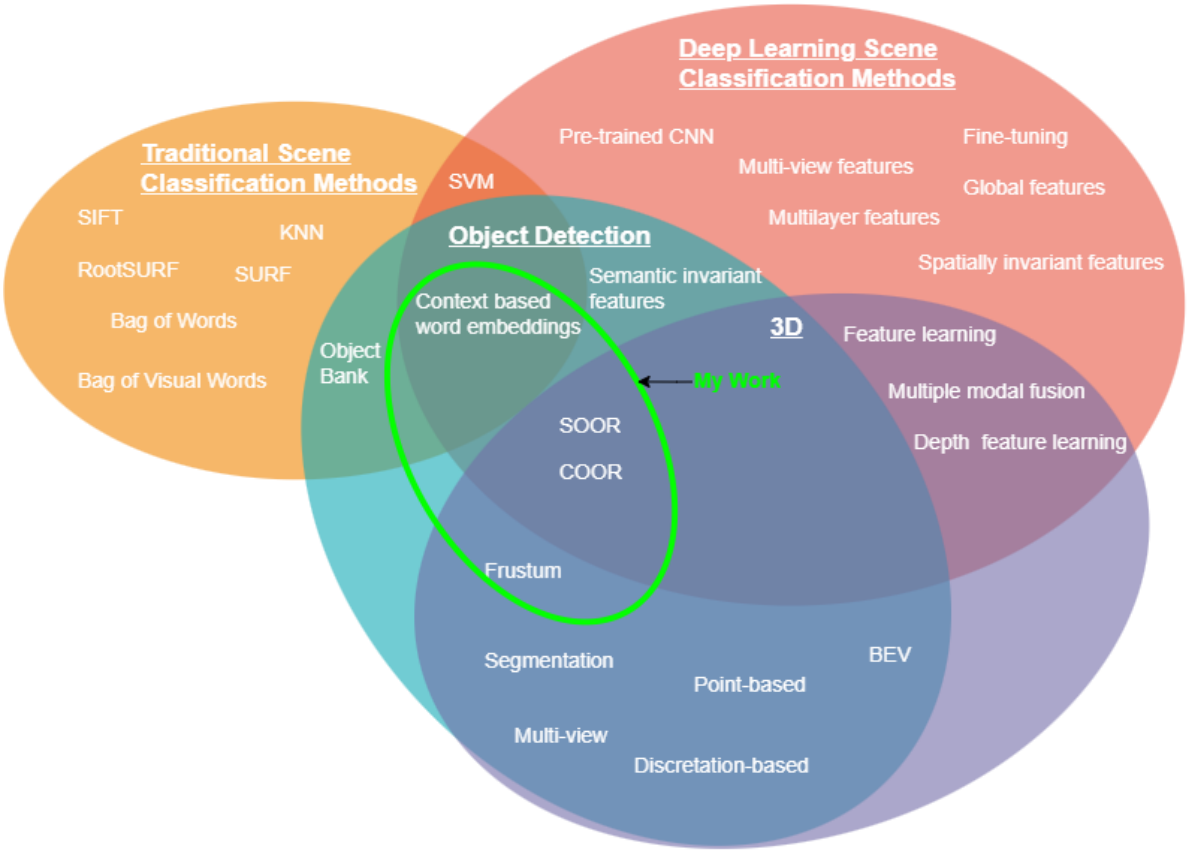


Figure 1: Venn diagram to visualise previous scene classification works and intended area of contribution

2.1 Object Detection

In recent years, object detection and recognition have developed rapidly and have been used in areas such as autonomous driving systems, robot perceptual systems, or detection of irregular events in video surveillance. Following the development of deep learning methods,

object detection has gradually transformed from traditional image processing methods to deep neural networks.

2.1.1 2D Object Detection

Traditional approaches start with extracting features using image processing methods such as SIFT[12], HOG[13], and SURF[14]. A typical architecture of a 2D object classifier is to use a convolutional neural network (CNN) to extract CNN convolution features, train the region proposal network (RPN), and finally train the network to detect the object area. For a one-stage object detection network, category and location information is given through the backbone network without using the RPN network. One-stage networks are faster but accuracies are lower compared to two-stage networks. Some examples of one stage object detection networks include YOLO variations[15][16][17], SSD[18], DSSD[19], Retina-Net[20] etc. These networks have achieved success on benchmark datasets such as KITTI[21] and COCO[22]. Recently, visual transformer models have been proposed as state-of-the-art in computer vision for image recognition tasks. While they can outperform CNNs by almost 4 times when it comes to computational efficiency and accuracy, they require a large amount of data (14 million images)[23].

2.1.2 3D Object Detection

In the context of object detection, 3D detection techniques offer several advantages over their 2D counterparts, such as the ability to capture more accurate information about the environment by incorporating depth, shape, and size information into the detection process [24]. This is in contrast to 2D detection methods, which do not fully leverage such information, resulting in reduced detection accuracy due to a lack of data available for classification decision-making. In light of these advantages, it is prudent to explore the potential benefits of 3D object detection techniques for our intended application on a mobile robotic platform. By leveraging the shape and texture information provided by depth data, 3D point cloud-based detection holds particular promise for improving the robustness and accuracy of object detection in this setting.

Knowing the precise 3D location of objects is of paramount importance in robotics

search and retrieval tasks, particularly in the context of path planning search. In such tasks, the robot needs to efficiently navigate through the environment to locate and retrieve a target object in an expected 3D location.

Dependant on the types of input data, 3D object detection methods often are divided into 2 categories : region proposal-based methods, and single shot methods. Region proposal-based methods have 4 subcategories which are multi-view methods, segmentation-based methods, frustum based methods, and other. Single shot methods have 3 subcategories which are BEV-based methods, discretization-based methods, and point-based methods[25].

1. **Region Proposal-based Methods**

   These methods propose several regions that possibly contain objects and then extract the region's features to determine the category label of the proposal.

   Multi-view based methods. These methods combine proposal features from different view maps such as bird's eye view (BEV) and front view to generate 3D rotated boxes. The computation cost of these methods are often high. Chen et al [26] generated a group of candidate boxes from the BEV map and projected to feature maps of multiple views to combine region-wise features to refine the predicted 3D bounding boxes. Liang et al. used object detection, ground estimation and depth completion to help the network learn better feature representations[27]. However, these methods often use high computational resources and perform at a slower run time since each view needs to pool features for each proposal[25].

   Segmentation-based methods. Leveraging existing semantic segmentation techniques, segmentation-based methods remove most background points to generate a large amount of high-quality proposals on foreground points. This saves computation while keeping detailed features. These methods often achieve higher object recall rates compared to multi-view methods and often work well with complicated scenes with occluded and crowded objects[28][29].

   Frustum-based methods. These methods have emerged as an efficient and accurate approach to 3D object detection by exploiting the matured 2D object detectors. These methods generate a 2D candidate region of objects and extract a 3D frustum

proposal for each candidate. Zhao et al incorporated the PointSIFT module into the network to capture orientation information of point clouds which achieved strong robustness to shape scaling and achieved success on both indoor and outdoor datasets[30]. Frustum based methods take advantage of matured 2D object detectors and can generate accurate 3D box predictions given a sparse point-cloud. The 2D proposals through the use of viewpoint frustums reduces computation cost[25].

2. **Single-shot Methods**

Single-shot methods directly predict class probabilities and using a single-stage network, regress 3D bounding boxes. This method runs at a high speed as it does not need region proposal generation and post processing.

BEV-based methods. Taking BEV representation as the input, Yang et al. [31] split the point cloud of a scene with equally spaced cells and encoded reflectance and then used a fully convolution network to estimate locations and heading angles of objects. This method runs fast at 28.6 fps. Yang et al. [31] improved this method later by using geometric and semantic priors from high-definition maps to improve robustness. They used coordinates of ground points to influence translation variance caused by slopes of the road for their BEV projection. BEV methods often do not generalize well for different point cloud densities. With a normalization map, the generalization improves[25].

Discretization-based methods. After converting a point cloud to a regular discrete representation, these methods apply CNN to predict categories and 3D boxes of objects. Li et al. [32] converted a point cloud into a 2D point map. Then, using a 2D FCN, they predicted the bounding boxes and confidences of objects. An extension to this is to discretize into a 4D tensor and extend 2D FCN detection into the 3D domain. While improving accuracy by 20%, the computation cost is high[25]. Zhou et al. [33] used a voxel-based discretization method and although the performance is strong, this method is extremely slow at 2 fps due to sparsity of voxels and 3D convolutions. Partial spatial information is inevitably lost in down-scaling feature maps. SA-SSD proposed by He et al. [34] addresses this by using structure information for improving localization accuracy in autonomous driving scenarios. Although this method performs well on the

KITTI BEV detection benchmark at a 74% for hard classes of cars, it was not evaluated on any indoor scenes or objects.

Point-based methods. These methods take raw point clouds as data by using a fusion sampling strategy for Distance-FPS and Feature-FPS and removes feature propagation layers[25].

The Frustum PointNets approach is well-suited for achieving our desired results as it benefits from the use of established 2D object detectors during the detection phase and incorporates 3D object detection, which utilizes depth, shape, and size information of objects. Furthermore, 3D object detection is advantageous in semantic object-based scene classification as it enables the exploitation of spatial relationships. Although multi-viewed methods are the most effective among 3D object detection methods, they tend to perform slower and require more computational resources. Conversely, single-shot methods are faster than region proposal methods; however, they face challenges with generalization to sparse point clouds and indoor object performance.

2.2 Scene Classification

There are two types of scene classification: traditional methods and deep learning methods. Traditional scene classification techniques use feature detection, feature description, and classification. Common feature detectors used are SIFT, FAST (Features from Accelerated Segment Test), SURF, ORB, MSER (Maximally Stable External Region), and BRISK (Binary Robust Invariant Scalable Keypoints)[1]. Deep learning methods, specifically CNN, automatically extract the features from the image with no specified overhead of manually extracting features. The "Places" dataset was developed by authors who created a scene classification model based on deep CNN features. The dataset consists of diverse and sense images for scene classification[2].

Common datasets for scene classification include those shown in Table 1.

Table 1: Popular datasets for scene classification. Scene15[35], MIT67[36], ImageNet[37], Places205[38], Places365-C[39], NYUD2[40], SUN RGB-D[10]

Type	Dataset	Images	Classes	Labels
RBG	Scene15	4,488	15	Indoor + Outdoor
	MIT67	15,620	67	Indoor
	ImageNet	14 million	21,841	Objects
	Places205	1,076,580	205	Indoor + Outdoor
	Places365-C	8 million	365	Indoor + Outdoor
RGB-D	NYUD2	1,449	10	Indoor
	SUN RGB-D	10,355	19	Indoor

Three common challenges that come with scene classification are large intraclass variation, semantic ambiguity, and computational efficiency. Intraclass variation arises from the diversity of objects, backgrounds, and human activities, as well as variations in image conditions such as changes in viewpoint, illumination, occlusion, clutter, and blur. Semantic ambiguity occurs when different scene categories share common objects, textures, or backgrounds, making it difficult to accurately classify a scene. Computational efficiency is also an important consideration when implementing a scene recognition system on constrained resources. These challenges are critical factors in developing robust and efficient scene classification algorithms [7].

2.2.1 Traditional Methods

Traditional approaches often use SIFT and SURF to extract low-level features that include shape, color, and textures[41]. RootSIFT was proposed to create the Bag of Visual Words (BoVW) which is combined with attention methods for scene classification[1]. BoVW is an extension of the Bag of Words (BoW) representation of features that is a very common concept of Natural Language Processing in which the multiplicity is represented while grammar and word order are disregarded. An improved BoVW method was created by Lazebnik et al. [42] called Spatial Pyramid in which the image was partitioned into sub-regions then the histogram of local features was computed. Methods such as combining local and global features were used but it was found that while global spatial properties can be used to classify outdoor scenes, there is a need of high-level information for indoor scene classification[43]. Here, a method called the Object Bank was proposed and objects were found using pre-trained

detectors[44]. Often, an SVM is used for classifying traditional methods. Other classifiers such as linear and K-Nearest-Neighbours have been used previously as well[1].

Generally, outdoor scenes have a common layout where scenes have similar features such as grass at the bottom and trees at the top of the scene for a forest. Thus, outdoor scenes can be classified with adequate accuracy using local or global features. Indoor scenes however have many similar features between two classes due to occlusion, similar objects, and changes in illumination. Use of traditional methods for indoor scene classification fail to achieve good results due to the lack of semantic information[1].

2.2.2 Deep Learning Methods

Due to the difficulties of most traditional methods to perform well on indoor spaces, deep learning based approaches are mostly used today for scene classification. Despite over several decades of development in scene classification, most methods still have not been able to perform at a level that is sufficient in real-world applications. Common problems that deep learning aim to solve are large intra-class variation, semantic ambiguity, and computational efficiency[7].

The main **CNN framework** for deep learning methods are generally divided into pre-trained CNN models[45][46], fine-tuned CNN models[47][48][49], and specific CNN models[50][51][52].

Pre-trained CNN models overcome the issue of training data being scarce in certain applications and under-fitting of models during training. Training CNNs on large-scale datasets makes them learn enriched visual representations. However the effectiveness of pre-trained models depends on the similarity between the source and target domains. These models as feature extractors can be object-centric or scene-centric. Object-centric CNNs contain object descriptors and are often represented as a bag of semantics and are generally robust against size and scale but depend on the scaling of the dataset it was pre-trained on. Scene-centric CNNs often perform better than object-centric CNNs since scene-centric methods make use of more details of a scene such as semantic regions and topology of the scene[7]. However, Zhou et al. [53] showed that scene-centric CNNs may perform as object detectors without being explicitly trained on object datasets.

Fine-tuning the pre-trained CNNs using a target scene dataset improves performance by reducing the amount of possible domain shifts between two datasets[54]. A common

fine-tuning methods is freezing layers where the frozen CONV layers are not updated during fine-tuning and the modified CNN is fine-tuned by training on the new dataset. Smaller datasets can be augmented to make fine-tuning more effective and robust. However, there exists a problem via augmentation to fine-tuning that with too small patches as CNN inputs, and the final classification accuracy is worse[7].

Another group of deep models are specifically designed for scene classification. They are developed to extract effective scene representations from the input by introducing new network architectures. These include, but are not limited to, Dictionary-Learning CNN called DL-CNN which replace FC layers with dictionary learning layers that update parameters through back-propagation in an end-to-end manner [50], Global Average Pooling CNN called GAP-CNN which combines the original GAP layer and the 1x1 convolution operation to form a class activation map that can focus on class-specific regions[51], and Contextual Features in Appearance called CFA where CONV feature maps are inputs of LSTM layers that are used to describe spatial contextual dependencies[52].

Deep learning based scene representation determines what the model learns about scenes for effective results. This has been a focused area and as such many proposals have been created. These representations include: global CNN features, spatially invariant features, semantic features, multi-layer features, and multi-view features.

Global CNN feature based methods directly predict the probabilities of scene categories from the whole scene image[2][39][50]. The performance is greatly affected by the content of the input image and backgrounds may introduce noise to features[7]. These methods do not incorporate any semantic knowledge about the scene and are based on the overall scene appearance. Ayub and Wagner [55] achieved high accuracies on the SUN RGB-D dataset with a global approach. The approach involves extracting features from the data using a Convolutional Neural Network (CNN) and then applying a clustering algorithm to group similar data points into clusters. The centroid of each cluster is then used to represent a concept, and a support vector machine (SVM) is trained on these concepts to classify new scenes.

More recently, ResNet101-RNN[56] and OMNIVORE [57] have achieved highest accuracy in scene detection on the SUN RGB-D dataset. Both these approaches extract features on a global level using a CNN and uses attention mechanisms to determine points of interest for

improved feature weighting.

Mosella-Montoro et al. [58] proposes a new method for indoor scene classification using both 2D and 3D information with a multi-neighbourhood graph convolutional neural network to fuse the two sources of information. The authors first extract 2D and 3D features separately from the RGB and depth data, respectively. These features are then passed through two separate graph convolutional networks (GCNs) to model the relationships between feature points in the scene. The resulting node features are concatenated to create a joint feature representation that incorporates both 2D and 3D information. To further improve the feature representation, a multi-neighbourhood graph convolution operation that considers different neighbourhood sizes and scales is used. This allows the GCN to capture both fine-grained and coarse-grained relationships between feature regions. The joint feature representation is then passed through a classifier to predict the scene category. This approach achieved a 58.6% accuracy on the SUN RGB-D dataset for scene classification.

Spatially invariant feature based methods alleviate problems against geometric variations caused by sequential operations in a standard CNN. The process often follows this order: local patch extraction, local feature extraction, codebook generation for different regions of the image, and finally spatially invariant feature generation from the codebook. Sliding window approaches requires fixed aspect ratios which are not suitable for arbitrary objects with varying sizes[7].

Semantic feature based methods. Object based approaches allow for information on whether or not instances of salient regions are present in the scene which reduces redundant computation cost of the entire image. Different methods include selective search[49], Multi-scale Combinatorial Grouping (MCG)[59], and object detection networks such as the ones mentioned in section 2.1. Semantic feature based methods rely on the performance of object detection to extract features. Object outliers can cause problems in training and thus, many methods use an SVM to prune outliers and redundant regions[7].

Chen et al.[60] proposed a 2D scene classification method using context based word embeddings to represent scenes trained on the Places365 dataset. Two CNN models were used: one for scene classification to compute the initial top-5 predictions, and one for scene parsing to compute scene contents from foreground and background. The word vector module then computes the vector similarity between objects present in the scene and the top 5

predicted labels. This model achieved better accuracy than ResNet50 CNN and showed that objects play a vital role in scene classification.

Song et al. [11] proposed two methods of representing scenes that incorporate intermediate spatial relationships between objects present. The first method is co-occurring frequency of object-to-object relation called COOR and the second method is sequential representation of object-to-object relation called SOOR. COOR emphasises object presence and its frequency of occurrence in a specific scene and each spatial relation is represented as a triplet of ⟨object, relation, object⟩. This method evaluated, performs better than typical object-object co-occurrence methods. SOOR, on the other hand, is generated by the objects, their attributes, and relations in the sequential order of: Attribute(i), Object(i), Relation(V), Relation(D), Attribute(j), Object(j). Using the SOOR method achieved an accuracy of 55.5% on the SUN RGB-D dataset. Combining COOR and SOOR representations in scene classification yielded better results than each of the representations alone.

Multiple-view feature based methods integrates multiple features generated from complementary CNN models (features generated from networks trained on different datasets)[7].

3D scene classification adds information to the available features to learn. Depth information is invariant to lighting and color variations. It includes geometrical and shape cues which is useful in scene representation [7]. Depth information of RGB-D images can improve performance of CNN models compared to 2D images[61]. Some CNNs are designed for depth-specific learning. There are also methods [55][62] that fuse modalities which tends to do better on the NYUD2[40] dataset than other methods. However, best performing overall on both the NYUD2 and SUN RGB-D datasets was TRecgNet[63]. It is important to note that the SUN RGB-D dataset contains images from the NYUD2 set. The TRecgNet method avoids redundancy of concatenating features when combining modalities. It does so by performing a global average pooling to reduce feature dimensions after concatenation.

Huang et al.[64] proposed a method of scene classification using 3D point cloud data which out performs BEV methods by utilizing the relationship between objects to each other and objects to scene as well as using other cues such as 3D geometry of the scene and color. The multi-task method of deep learning that was implemented to collect many examples per class and also perform per-point semantic labeling of the point cloud. This method proved to perform well when using only geometry information compared to other methods such as

ResNet14[65]. Importantly, they showed that sparse 3D data is sufficient to classify indoor scenes with good accuracy[64]. They trained and tested on the ScanNet dataset[66].

Some state-of-the-art results for scene classification on the SUN RGB-D dataset are as follows in Table 2

Table 2: Comparison of state-of-the-art scene classification results

SOOR (2020)[11]	Cbcl (2020)[55]	GFN (2021)[58]	ResNet101-RNN (2022)[56]	OMNIVORE (2022)[57]
55.5	57.8	56.1	60.1	67.2

A 3D object-centric approach is desired for scene-classification for our pipeline. Using such an approach can leverage spatial and depth information between objects and scenes. This method best mimics the way humans interpret scenes as humans are able to classify scenes by recognizing the objects and functions of objects within an environment[1]. Scene representations such as using context-based word embeddings[60] or SOOR and COOR [11] would enable learning spatial relationships between objects and scenes alongside object co-occurrence.

The ability to classify scenes based on their spatial relationships is particularly important in the context of service robotics, where mobility is a key requirement. Since the viewpoint of the robot's camera can change, the 2D pixel relationships between objects in the scene will also shift. This makes it challenging to accurately recognize and classify the scene using 2D data alone. However, by utilizing 3D spatial metric data, the relationship between objects remains consistent regardless of the viewer's viewpoint. For instance, the distance between a chair and a table does not change in the 3D world even if the robot moves from one side of the room to the other. In contrast, the 2D pixel relationship between the two objects will differ significantly. By incorporating 3D spatial information into scene classification algorithms, service robots can more accurately and reliably recognize and understand the scenes they encounter.

2.3 Mobile Base Implementation

Although extensive studies have been performed on scene recognition, very few studies have implemented their methods in new test environments on a mobile robot.

Chen et al. [60] evaluated their 2D scene recognition pipeline on a robot operating in the real world on 3 environments: school, home, and shopping mall. Each environment contained second level places such as classroom, office, ComputerLab, cafeteria for school. The pipeline did well on school and home but their CNN scene parsing model could not find objects that distinguished shopping mall scenes enough such as shoes, watches, hats.

Fazl-Ersi and Tsotsos [67] performed experiments of their Histogram of Oriented Uniform Patterns methods for 2D scene classification using two mobile robots of different heights under three different light conditions: Cloudy, Night, and Sunny. Their method for scene classification generalized well in different lighting conditions and perspective changes and out performed other state of the art scene classification methods at the time.

Liu et al. [68] implemented a generative probabilistic hierarchical model for indoor scene classification where low-level visual features are associated to objects and contextual relations are used to associate objects to scenes using a 3D range sensor to increase detection accuracy. This method was tested using real data captured by a mobile robot navigating in an office and home environments. The authors provide a detailed description of the robot behavior and decisions made in each type of room as it navigates. Overall, the robot is influenced by fast findings of objects and only makes inferences when objects are detected as it is an object-centric approach. There were also problems that occurred in scenarios with rare object combinations.

3 Methods

This thesis work will focus on utilizing a 3D object-based representation of scenes to form prior scene contexts for robotic search applications. The proposed approach for 3D scene classification is to use objects and their relation to other objects as scene features. This approach involves detecting and extracting object instances from the 3D scenes, and encoding their spatial relationships as features for classification. The Sequentially Encoding Object-to-Object Relations method proposed by Song et al. [11] allows for spatial relations to be captured between objects. This method will be extended to 3D object-to-object relations from the original 2D pixel-based object-to-object relations. To detect 3D objects and their 3D bounding box, a Frustum Point-Net [8] method was chosen for it's efficiency in computation and proven ability to generalize and perform well in indoor settings. Scene classification and 3D object detection will be trained using the SUN RGB-D dataset with over 10300 depth images[69]. Finally, this pipeline will be tested on a mobile robotic wheelchair using a stereo depth camera.

3.1 Scene Classification

The use of Sequential Object-to-Object Relations (SOOR) encoding in scene recognition has several advantages over other object relation techniques. One key advantage is that SOOR encodes the spatial relationships between objects in a scene, which allows for a more accurate representation of the scene's structure. This sequential encoding is achieved by capturing the relative positions of objects in the scene and using this information to construct a object-to-object representation of the scene. Compared to other scene area relation techniques, such as Graph Convolutional Networks (GCNs)[58], SOOR encodings utilize object information and has been shown to achieve similar accuracy in recognition tasks with only using 2D image data. This is because GCNs are limited by their fixed distance graph structure and may not be able to capture other features of objects in a scene such as size and other attributes. SOOR encoding is capable of representing not only co-occurring object relationships but also spatial relationships, such as extended directional relations, distance, area and size, which may not be captured by other object relation techniques. This work will be making an extension of the 2D object to object relations into 3D real metric scale

values for object location, distance, and volume. This extension allows for a more accurate representation of the scene's layout and can improve the already state-of-the-art accuracy of scene recognition from the original work by Song et al. [11]. The use of SOOR encoding in scene recognition offers several advantages over other object relation techniques, including its sequential encoding of spatial relationships, flexibility in representing the scene's structure, and its ability to capture a wide range of object relationships. These advantages make it a promising approach for improving the accuracy of scene recognition tasks.

In addition to directional relations, the Sequentially Encoding Object-to-Object Relations (SOOR) method focuses on spatial relations that can reflect the spatial layout of the scene. With co-occurring object based representations such as the co-occurrence matrix, some spatial relationships such as overlaps or closeness and relative positions are neglected. SOOR descriptions of scenes are generated by the objects, their attributes and relations in the following template:

"Attribute(i) Object(i) Relation(V) Relation(D) Attribute(j) Object(j) in Scene"

when i is not equal to j.

Attribute(*): a binary encoded size value of object *

00 being small, 01 medium, 10 large, 11 huge

Object(*): label of object *

Relation(V) = $[V_\alpha^{(i,j)}, V_\beta^{(i,j)}]$

$V_\alpha^{(i,j)} = [g(x_1^i - x_1^j), g(y_1^i - y_1^j), g(z_1^i - z_1^j), g(x_2^i - x_2^j), g(y_2^i - y_2^j), g(z_2^i - z_2^j)]$

cross relations of overlapping

$V_\beta^{(i,j)} = [g(x_1^i - x_2^j), g(y_1^i - y_2^j), g(z_1^i - z_2^j), g(x_2^i - x_1^j), g(y_2^i - y_1^j), g(z_2^i - z_1^j)]$

direct relations of overlapping

$$g(x) = \begin{cases} 0, & \text{if } x <= 0 \\ 1, & \text{if } x > 0 \end{cases} \tag{1}$$

Relation(D) = $[dist(center(b^i), center(b^j)), dist_{min}, dist_{max}]$

the distance between centers of objects, minimum box distance and maximum distance

$$\text{dist encoding} = \begin{cases} 0, & \text{if } dist <= threshold \\ 1, & \text{if } dist > threshold \end{cases} \tag{2}$$

An example of one SOOR representation would be "01 (small) chair 000000000110 (upper left corner overlapped in view) 001 (close distance) 11 (large) table in dining_room". This representation was extended from the original Song et al publication to include z depth components of the object-to-object relations and to use metric scale values instead of pixel locations. The size of the object is classified based on a predetermined range after sorting average volume of objects in the SUN RGB-D dataset, sorting by increasing volume, splitting the list of volumes into 4 quarters and averaging each of the sections. This resulted in a rough estimate of the average sizes of the smallest 25% of objects, the medium 25% etc.

The proposed SOOR binary encodings are then converted to decimals as each binary encoding will map to only one decimal number. Each object-to-object relation is captured into a 6 word sentence with grammatical order to be fed sequentially into a RNN sequential model. During training, the first 1200 words are passed in sequentially. The representation of the scene was limited to maximum 200 object-to-object relations with 6 words each. A GRU unit is used in the RNN model to obtain the hidden activation $\mathbf{h} = [h_1, ..., h_T]$. The RNN training architecture is formalized as follows:

$$\begin{aligned} r_t &= \sigma(W_t x_t + U_r h_{t-1}) \\ z_t &= \sigma(W_z x_t + U_z h_{t-1}) \\ \bar{h_t} &= tanh(W x_t + U(r_t \bigodot h_{t-1})) \\ h_t &= (1 - z_t) h_{t-1} + z_t \bar{h_t} \\ y &= argmax(\Phi(h_T)) \end{aligned}$$

Where σ is a logistic sigmoid function, $\bigodot$ is an element-wise multiplication, Φ is two fully connected layered neural network. h_T, the last element of the hidden activations, is passed through two fully connected layers to determine the scene category y of the input sequence.

3.2 Implementation of Frustum PointNets

A frustum-based method of Frustum PointNets was chosen for this thesis for training a 3D object detector. This method leverages established 2D object detectors during detection and the 3D object detectors can make use of the depth, shape and size of an object. Additionally, 3D object detection is valuable in semantic object-based scene classification for leveraging spatial relationships. While multi-viewed methods perform the best out of 3D object detectors, they perform at a slower and require more computational resources. And while single-shot methods are faster then region proposal methods, there are issues with generalization to sparse point clouds and indoor object performance. Frustum methods have proven to do well with sparse point clouds unlike most other methods with desirable performance on 3D data. This is of particular importance in the current study since the particular stereo camera of our setup produces sparse point clouds and can only produce dense point clouds at a slower rate which is undesirable for real-time robotics purposes.

Typical convolution networks often require regular input data such as image grids or 3D voxels for weight sharing and kernel optimizations. Voxel grids however renders the data voluminous while also introducing quantization artifacts that can obscure natural invariances of objects.

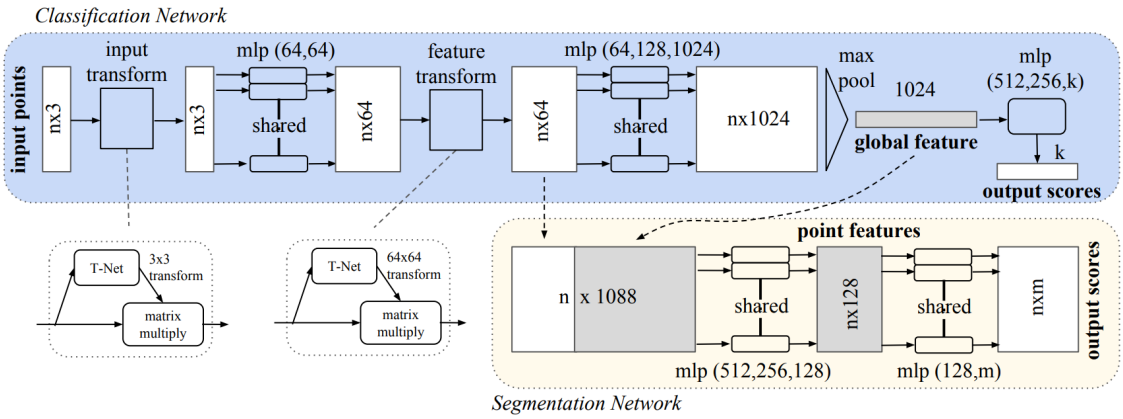


Figure 2: PointNet architecture[9].

Figure 2 shows the architecture of the PointNet network. PointNet is a deep neural network for processing point clouds. It has many applications such as classification, part segmentation, and semantic segmentation. The Frustum PointNet architecture for classifying 3D objects utilizes this method for point cloud processing. The classification network takes

n points as input, applies input transformations and feature transformations and finally aggregates point features by max pooling. Output is the classification scores for k classes. The segmentation network concatenates global and local features and outputs point scores using multi-layer perceptrons. A Batchnorm is used for all layers with ReLU and dropout layers are used for the last mlp in the classification network.

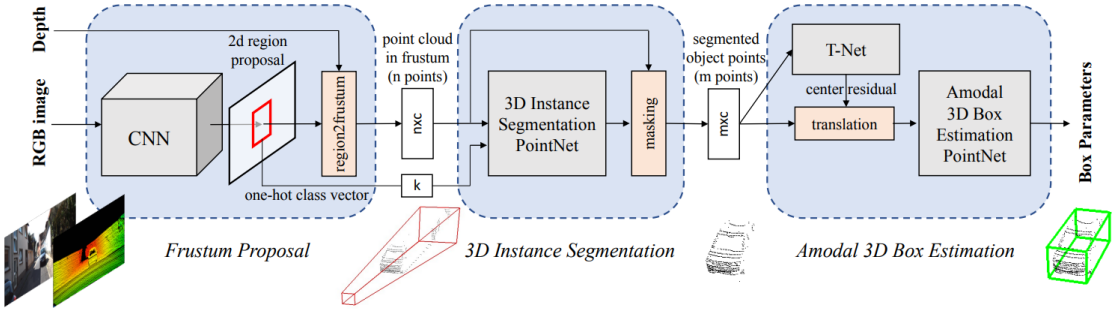


Figure 3: Frustum PointNet architecture[8].

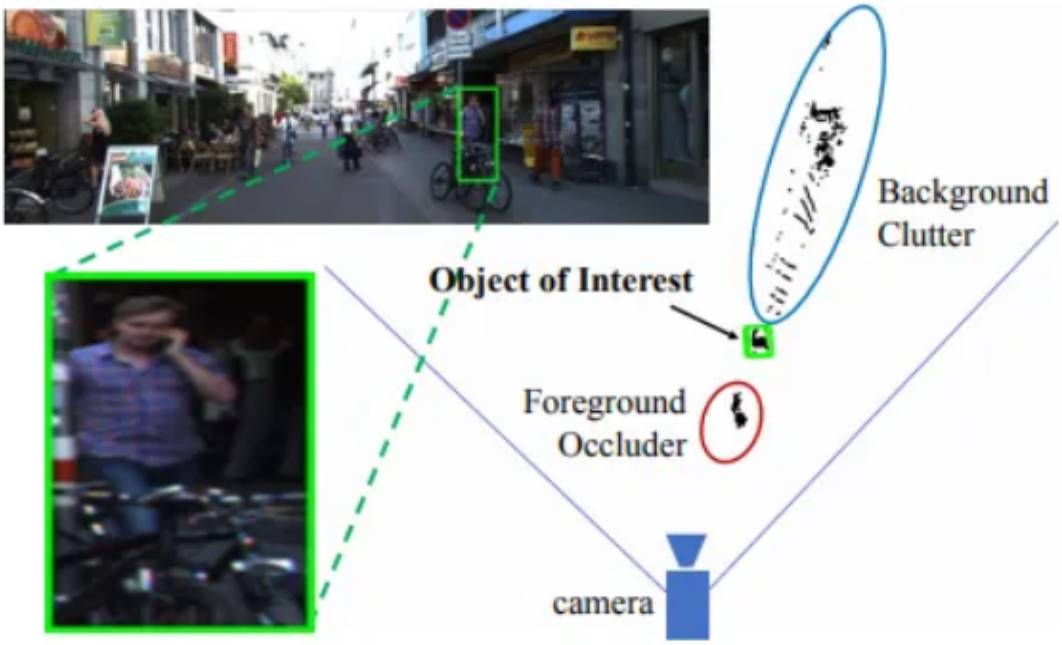


Figure 4: Frustum PointNet region proposal generation[8].

Figure 3 illustrates the architecture for Frustum PointNets for 3D object detection. The first step is the **frustum proposal**. Using a 2D CNN object detector, a 2D object region is proposed on the RGB image. An example of this is shown in Figure 4. With known camera properties, a frustum can be extracted from the 3D point cloud using the 2D image bounding

box. The orientation of each frustum is normalized to a center view since there is a large variation of frustum directions otherwise.

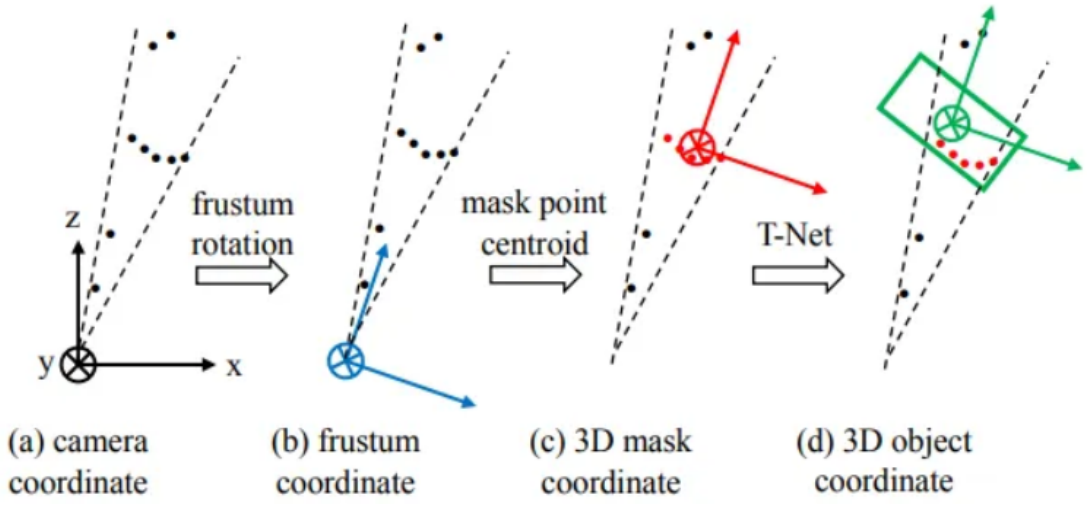
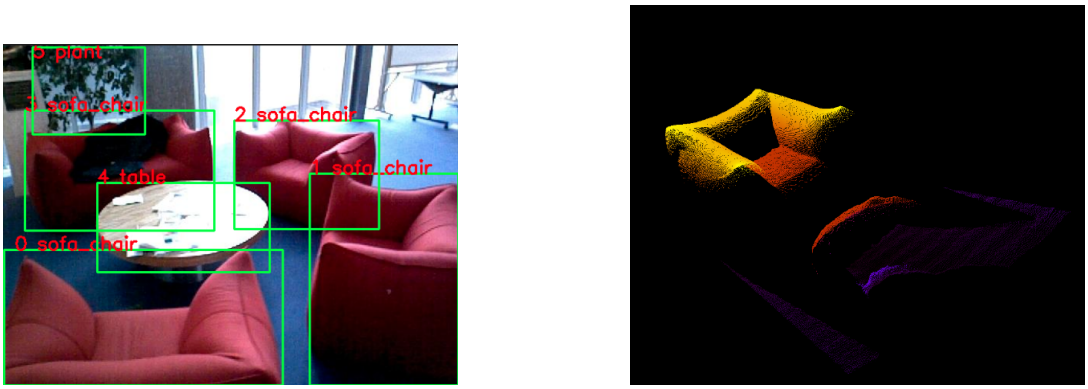


Figure 5: Frustum PointNet change of coordinate view and masking

Next, is the **3D instance segmentation** module. Similar to Mask-RCNN, which achieves instance segmentation by binary classification of pixels in the 2D image region, the instance segmentation module in this architecture does 3D segmentation by predicting the 3D bounding box center. Segmenting the object from background of a bounding box can be hard in a 2D image because of occluding objects and background clutter. This process is shown in Figure 5. This task is easier when using a 3D point cloud where pixels from the same object are close to each other vs occluding objects and background pixels. We use the PointNet point cloud processing network to segment the 3D instance. An example of this can be seen in Figure 6 below.



(a) 2D object detection image with bounding boxes

(b) Segmented Frustum PointNet of object 0 sofa_chair

Figure 6: Visualization of 2D object detection of a scene and its corresponding 3D instance segmentation for one object

The network takes a frustum point cloud as input and predicts a score for each point for how likely the point belongs to an object. Lastly, given the segmented object points, the **amodal 3D box estimation** module estimates the object's amodal oriented 3D bounding box through box regression PointNet (Box-Net) and a preprocessing transformer network (T-Net). The center residual predicted by the box estimation network is combined with previous center residuals from the T-Net and the masked centroid for the predicted center.

$$C_{pred} = C_{mask} + \Delta C_{t-net} + \Delta C_{box-net}$$

To obtain the 2D region proposals needed for frustum extraction, a pre-existing 2D object detector model was fine-tuned on the intended training dataset. Transfer learning has become an effective technique for improving the performance of deep learning models, especially when dealing with limited training data. In the context of detecting 2D objects using Faster R-CNN for the SUN RGB-D dataset, transfer learning can help to achieve good results by leveraging pre-trained models on larger datasets such as the COCO dataset. By fine-tuning the pre-trained model on the target dataset, the model can learn to extract useful features specific to the new task, while also retaining its ability to generalize to new examples. In addition, using transfer learning can reduce the amount of training time required and prevent overfitting. Overall, incorporating transfer learning into the pipeline for object detection using Faster R-CNN on the SUN RGB-D dataset can lead to better performance

and more efficient training.

The 2D object detector chosen was Faster R-CNN [70] which is an extension of Fast R-CNN. This architecture achieves best results on the COCO dataset (a large scale dataset for object detection and segmentation) currently which is why it was chosen for transfer learning. R-CNN is a Region-based Convolutional Neural Network[71] that can detect 80 different types of objects in images. The contribution of R-CNN was to extract features based on a CNN and proposes regions using a selective search algorithm. Fast R-CNN[72] is a single stage CNN and uses a new layer to pool regions of interest but neglects how the region proposals are generated in the original R-CNN network. The general architecture of Fast R-CNN is shown in Figure 7. The model accepts an image as input and the convolutional layer outputs a feature map that is fed into a ROI pooling layer. The pooling layer extracts a fixed-length feature vector from each region proposal. The feature vector is put into the softmax layer to predict class scores and also put into the fully connected layer to predict the bounding boxes of the detected objects.

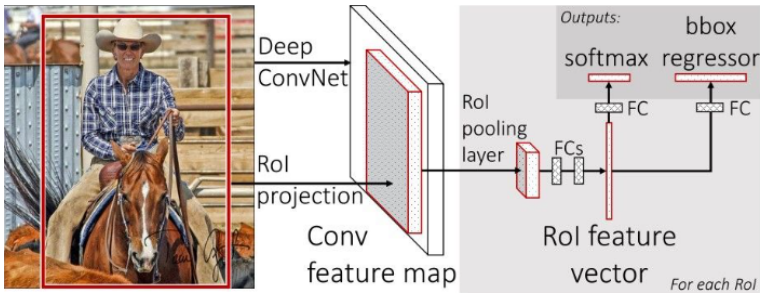


Figure 7: Pipeline for Fast R-CNN

Faster R-CNN uses a region proposal network to feed only proposed regions into Fast R-CNN as shown in Figure 8. Faster R-CNN first generates region proposals and for all region proposals in the image, a fixed-length feature feature vector is extracted from each region using the ROI pooling layer and the feature vectors are put into a softmax layer and a fully connected layer to predict class scores and bounding boxes.

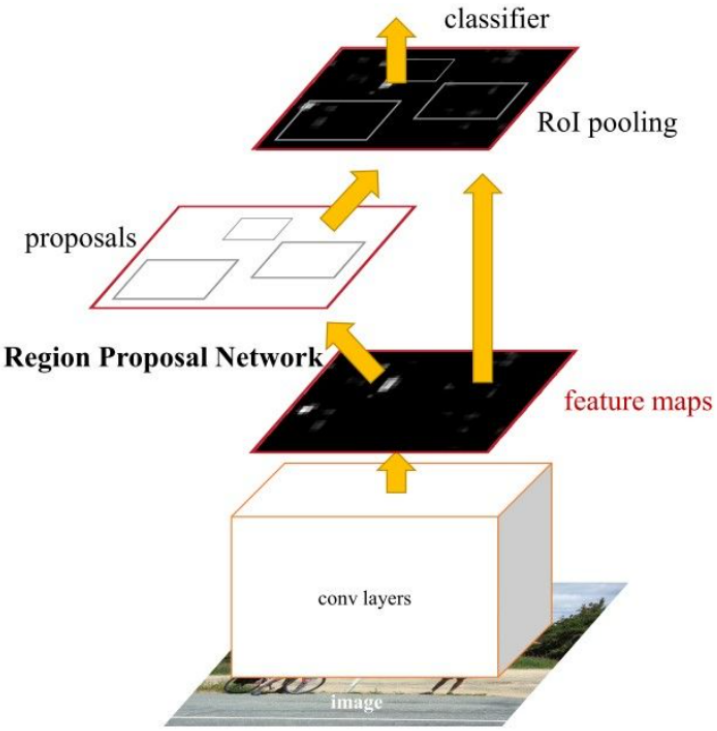


Figure 8: Pipeline for Faster R-CNN

Using a pre-trained model of Faster R-CNN from Meta's Detectron2 library [73] which provides state of the art detection and segmentation algorithms, the weights of the model were fine-tuned and trained on the SUN RGB-D dataset using only the 2D ground truth bounding boxes and provided object labels. The best performing Faster R-CNN model on the COCO dataset was chosen to be fine-tuned.

3.3 Data Collection

The present work utilized the SUN RGB-D dataset [10], which comprises 10355 RGB-D images of indoor scenes captured by four different sensors. This dataset is densely annotated with 146,617 2D polygons and 58,657 3D bounding boxes with object orientations, as well as 3D room layout and scene categories. The SUN RGB-D dataset includes data from NYU depth v2 [40], Berkeley B3DO [74], and SUN3D [69], and is organized by sensor. This dataset is currently the largest available dataset that provides information necessary for both 3D object detection and classification, as well as for training scene classification models. To increase the amount of data available for 3D object detection training, the training data

was augmented through bounding box perturbation and perspective shifts of 0.1m, resulting in five times the original amount of data. The bounding boxes were shifted randomly in a direction corresponding to 0.1 ratio of the original bounding box size.

Using the provided metadata, the SUN RGB-D data captured from Kinect 1, Kinect 2, RealSense and Xtion sensors were split into folders for calibration, depth, image, and label dimensions with text files that indicate which data IDs belonged to validation, test, or training data. Although the data was collected from different sensors, these sensors are all stereo cameras which is the type of sensor I will be implementing the scene classification pipeline with. Since the models are trained using stereo data, it proves viable with depth data from stereo vision.

The preset train-validation-test split provided by the SUN RGB-D dataset metadata was found to have an unbalanced amount of scenes in each set. For object detection training, the dataset's preset split was kept. However, for learning scenes, the dataset was re-split so that each set had a balanced amount of each scene relative to the percentage split allocated. Removing classes from a training dataset with imbalanced classes and re-sampling data can improve model performance and avoid model biases towards the majority classes. Thus, object classes with less than 20 instances were removed from training and testing. Scene classes with less than 50 instances were removed as well. This resulted in using 22 scene labels out of the original 45 labels. Each scene class was re-balanced for training by rearranging the order of the SOOR relations. Through training experimentation, it was found that at least 1000 instances of each scene was needed for training for best results. The 'furniture_store' class was also omitted due to the resemblance many layouts in this scene has to other labels in the set such 'kitchen', 'bedroom', 'dining_area', and 'home_office'. To avoid confusion in training, 'furniture_store' was omitted.

3.4 Mobile Base

To test the validity of the scene classification pipeline, a mobile robotic base is needed to replicate a scenario in which a service robot experiences different scenes in a building. It is important to evaluate the real-life applications of the algorithms we create to determine robustness, practicality, and its ability to generalize. Knowing fall-backs and strengths in

real life scenarios will be beneficial in continuing to improve service robots that require interactions with human environments.

Figure 9: CAD of wheelchair mobile base.

Figure 10: ZED2 stereo camera.

A mobile wheelchair was used shown in Figure 9 with a ZED2 stereo camera shown in Figure 10 with a pan-tilt unit for active vision. The PID controller code for the wheelchair has actively tuned through experiments. The pan tilt unit with the camera attachment has been implemented. A navigation stack was incorporated into the wheelchair controls to be combined in the future with scene recognition. The computer that will run the pipeline has a NVIDIA GeForce GTX 1070 graphics card with an Intel Core i7 CPU at 2.9GHz.

Since the stereo camera sensor used in this thesis is not one of the sensors that the SUN RGB-D training data was captured by, differences in quality and resolution of the depth and

image data must be considered. Stereo cameras can be different in terms of camera distance, focal length, range of depth, picture size, point cloud density etc. Thus, when testing using a different sensor, results may vary. Despite this, the SUN RGB-D dataset is a collection of data from 4 different stereo cameras and models generalize well to specification differences when tested on a hold out test set. The specifications are listed below in Table 3 where the ZED2 camera will be used for our implementation. The ZED2 has better or same specifications than the other sensors used for data collection and has options in parameter settings to reduce the size of the depth image.

Table 3: Specifications of stereo camera sensors[10][75]

Camera	Resolution of Color	Resolution of Depth	Max Depth	FoV
Kinect V2	1920x1080	512x424	4.5m	70/60
Kinect V1	640x480	320x240	4m	57/43
RealSense	1920x1080	604x480	10m	87/58
Xtion	1920x1080	604x480	3.5m	58/45
ZED2	**1920x1080**	**1280x720**	**20m**	**110/70**

Another key point to note is that the ZED2 camera has different modes of depth sensing in which the quality and density of the resulting point cloud can be improved upon. The maximum range can be extended to 40m and there is an option for NEURAL depth mode which results in a dense point cloud. Utilizing this feature and the Spatial Mapping feature, where a resulting fused point cloud can be made from the environment, the ZED2 camera can produce a dense point cloud to detect scenes and objects[75].

In the event that the differences in sensors prevent accurate detection of objects, a simple solution would be to utilize well established 2D object detectors and estimate their 3D location using the available point cloud information. The desired scene classification technique should not rely on precision but rather more abstract concepts such as object to object spatial relations which can be gathered easily with an estimate of object centers.

The importance of being able to classify scenes using spatial relationships comes with the mobility of service robotics that such classification algorithms can be implemented on. With the mobility of a robot, 2D object pixel relations will shift depending on the viewpoint of the camera. However, utilizing 3D spatial metric data, the relationship between objects do not change depending on the view. A chair's distance from the table does not change in the

3D world if the viewer moves from one side of the room to the other. However, the 2D pixel relation between the two objects will vary greatly. By incorporating 3D spatial information into scene classification algorithms, service robots can more accurately and reliably recognize and understand the scenes they encounter.

3.5 Pipeline

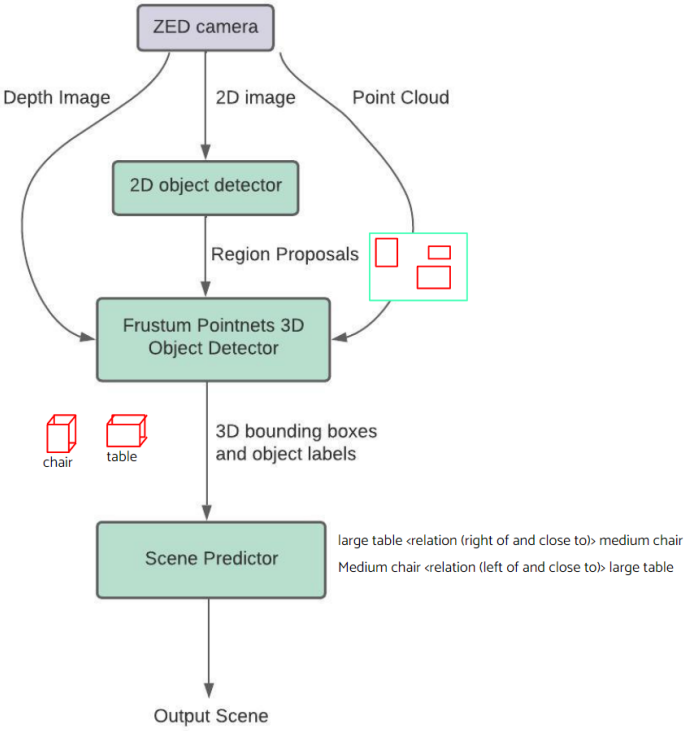


Figure 11: End-to-end pipeline for scene classification

The proposed end-to-end pipeline, as illustrated in Figure 11, comprises several components that work in tandem to predict the final scene. First, the 2D Faster R-CNN object detector generates region proposals in the image, which serve as input for the Frustum PointNets 3D object detector. The 3D object detector leverages the 2D region proposals and information from the ZED2 camera, including the depth image and point cloud, to predict the 3D bounding box and class labels of the object. The resulting predictions are then encoded into the Sequential Object-to-Object Relations (SOOR) format. Finally, the scene predictor module uses the SOOR relations to perform inference and produce the final predicted scene.

4 Results

4.1 3D Object Detection

A 2D object detector was first trained before training the Frustum Point-Nets model. This detector is needed to provide 2D box region proposals for the 3D frustum point cloud extraction.

Table 4: Training results and evaluation of Faster R-CNN on the SUN RGB-D dataset

Validation Accuracy	Test Accuracy	False Negative
87.3	83.4	23.2

We want to evaluate the performance of this model not only on its high class accuracy but also it's low rate of false negatives. Results are shown in Table 4. This is important in the desired use case of proposing regions for the 3D Frustum Point-Net object detector. A low rate of false negatives is desirable such that we do not miss possible regions to look at for 3D object detection.

In the context of detecting 2D objects using Faster R-CNN for the SUN RGB-D dataset, transfer learning was beneficial in achieving good results by leveraging the pre-trained model on the larger COCO dataset. By transferring the knowledge from pre-trained models to the target dataset, the model can learn to extract useful features that are relevant to the new task, while retaining its ability to generalize to new examples. Incorporating transfer learning into the pipeline for 3D object detection using Faster R-CNN on the SUN RGB-D dataset for the 2D proposal region obtains accurate results for valid proposals.

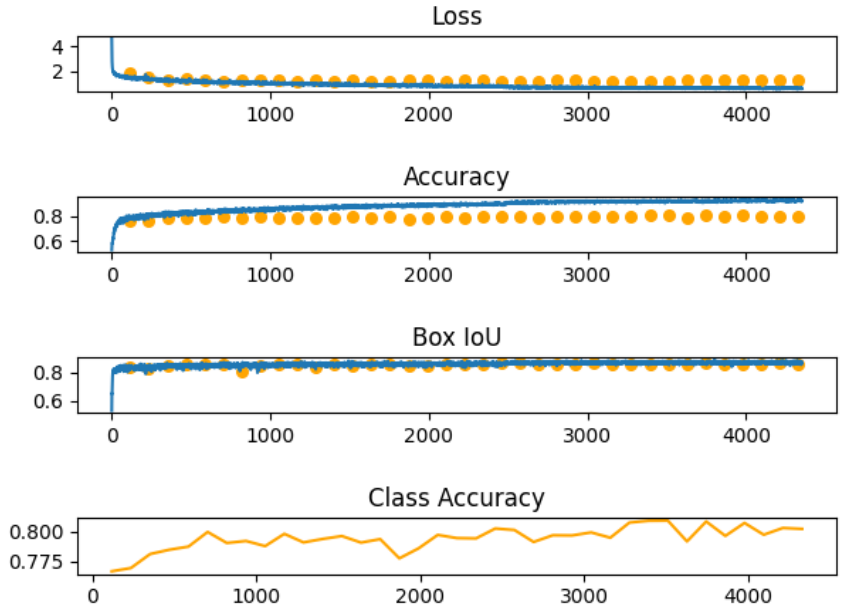


Figure 12: Frustum PointNet training curve of 30 object classes. X-axis lists the number of batch iterations, yellow dot represents the evaluation value after each epoch, blue line represents the continuous values during batch training

For training the Frustum Point-Net 3D object detector, models were trained using only 30 object classes to test the viability of the network. Figure 12 shows the training curves during this process. The model was saved every 10 epochs and was saved if the model out performs the previous best model after it's epoch evaluation on the validation set.

Training happened for 30+ epochs but the best model was saved at epoch 17. As can be seen in Figure 12, over-fitting of the data started occurring after epoch 17 where batch accuracy was increasing while evaluation accuracy remained similar and batch loss decreased while evaluation loss remained similar. The model evaluated at 80% accuracy for validation and at 79% for a hold out test set that included 5050 new and unseen RGB-D images. Mean accuracy precision (mAP) was 0.52. With this success, a new model was trained on the full set of 1000+ viable object classes.

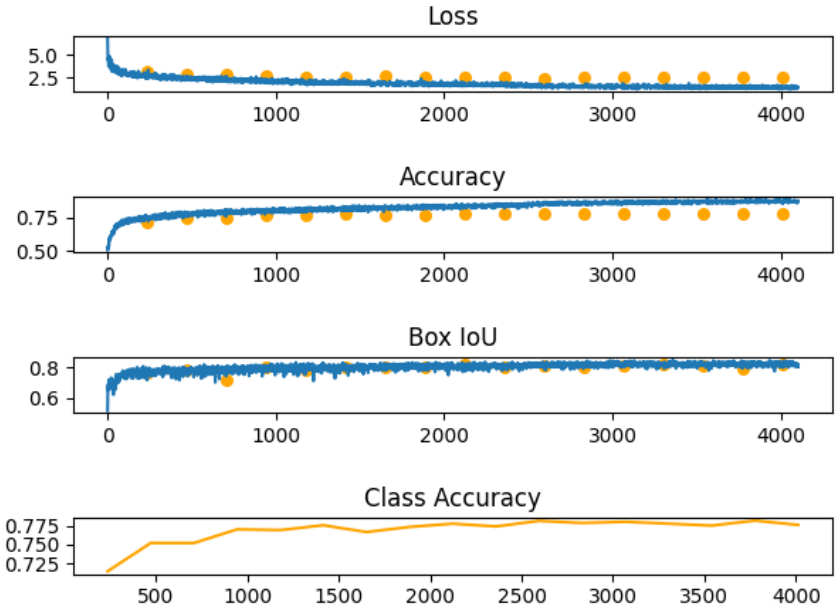


Figure 13: Frustum PointNet training curve of all object classes. X-axis lists the number of batch iterations, yellow dot represents the evaluation value after each epoch, blue line represents the continuous values during batch training

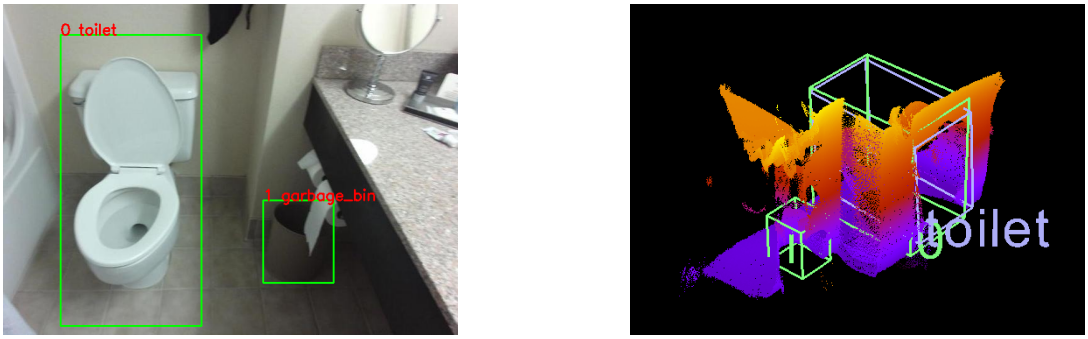
The evaluation metrics used for object detection are as follows. We report the validation and test accuracy as average class accuracy for objects that were detected. mIoU is the mean Intersection over Union which is often used to evaluate performance of object detection by comparing the ground truth bounding box to the predicted bounding box. $IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$ this is a ratio between the area of overlap between the predicted bounding box and the ground truth bounding box and the area of union which is the total area comprised of both the predicted bounding box and the ground truth bounding box. Box IoU reports both the IoU of the predicted 3D bounding box against the ground truth bounding box. This metric reports the first number as the IoU for the bounding box if the view was front and center for the object and we evaluated on only the first face of the box in 2D. The second number reports the 3D IoU of the predicted box and the 3D ground truth box. mAP reports the mean Accuracy Precision. Detected results are considered to be true or false positives according to the box IoU. To be considered a correct detection, the overlap area must exceed 0.7 by the previously described metric.

The results of the object detection model training indicate that the best model was

achieved at epoch 10, after which overfitting began to occur. The achieved validation accuracy of 78% for all object classes during training is a good indication of the model's ability to accurately detect objects in the images used for training. The test accuracy of 77% is also encouraging, although slightly lower than the validation accuracy, suggesting that the model is capable of generalizing well to new data. The achieved mean average precision (mAP) of 0.41 is a measure of the model's precision in object detection, and while it may seem relatively low, it is important to note that mAP can vary greatly depending on the specific dataset and object classes being detected and can especially vary when evaluating in 3 dimensions. Overall, the results suggest that the object detection model shows promise in accurately detecting objects, with some room for improvement in precision. The detailed results can be found in Table 5.

Table 5: Training results and evaluation of Frustum Pointnets

Classes	Validation Accuracy	Test Accuracy	mIoU	box IoU	mAP
30	80	79	0.64	0.51/0.44	0.52
Full 1000+	78	77	0.61	0.47/0.38	0.41



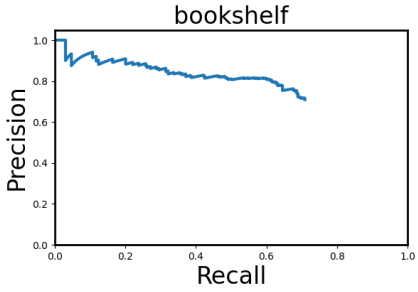
(a) visualization of toilet 2D ground truth bounding box

(b) Visualization of 3D detected toilet bounding box

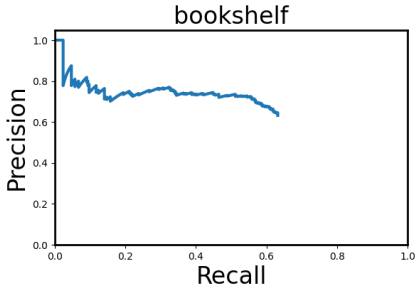
Figure 14: Visualization of detected and ground truth boxes. Only "toilet" visualized for easier view.

Figure 14 shows a visualized example of the object detector. Only "toilet" was shown for clarity. The 3D green box is the labeled ground truth from the SUN RGB-D dataset and the blue box is the predicted box. As can be seen, the predicted bounding box has slight shifts from the ground truth bounding box. In practice however, the exact location of a detected object is of less importance than the ability to detect objects and the accuracy of object class

prediction since this work focuses on the general spatial relations between objects in a scene.

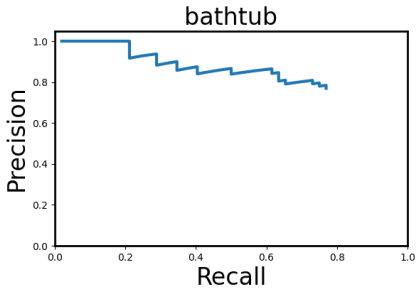


(a) "bookshelf" class precision-recall curve reduced set model

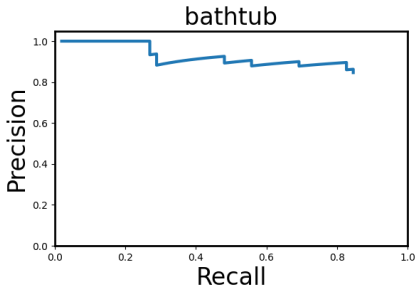


(b) "bookshelf" class precision-recall curve full set model

Figure 15: "bookshelf" class precision-recall curves comparison between the model trained on a reduced class set and the model trained on the full class set



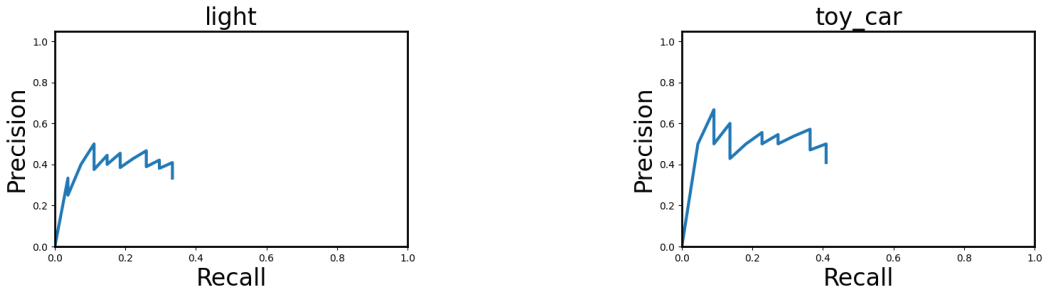
(a) "bathtub" class precision-recall curve reduced set model



(b) "bathtub" class precision-recall curve full set model

Figure 16: "bathtub" class precision-recall curves comparison between the model trained on a reduced class set and the model trained on the full class set

The precision-recall curves for the "bookshelf" and "bathtub" object classes, as shown in Figure 15 and Figure 16, provide insights into the performance of the object detection models. The curves demonstrate that the performance of the models varies for different classes of objects, indicating that the models are more successful at detecting certain types of objects than others. Interestingly, the model trained on the reduced class outperforms the model trained on all classes for the "bookshelf" class, while the model trained on all classes performs better for the "bathtub" class. However, the differences in performance between the models are relatively small and not significant enough to declare one model vastly superior to the other.



(a) "light" class precision-recall curve full set model

(b) "toy_car" class precision-recall curve full set model

Figure 17: Precision-recall curves for objects that have high variability in appearance for the model trained on the full class set

The results of the object detection model show that certain object classes, such as "lights" and "toy cars" in Figure 17 exhibit lower recall and precision than others. This is not surprising, as these objects have high variations in appearance, making them more difficult to accurately detect. It is also possible that some objects, such as toy cars, appear less frequently in the dataset, leading to a lower quality of feature learning for these classes. Overall, these results suggest that the performance of the model can be impacted by the complexity and frequency of appearance of certain object classes.

4.2 Scene Classification

We train the SOOR RNN scene classifier with ground truth object classes and bounding boxes from the SUN RGB-D dataset. After parameter tuning, the model with the best results was saved and the training curve is shown below in Figure 18. The epoch with the best evaluation was saved and training was stopped once validation accuracy did not improve within the most recent 10 epochs. We train and test when at least 2 objects in the room are present to capture the object-to-object pair relations explicitly and the co-occurrence relationships implicitly.

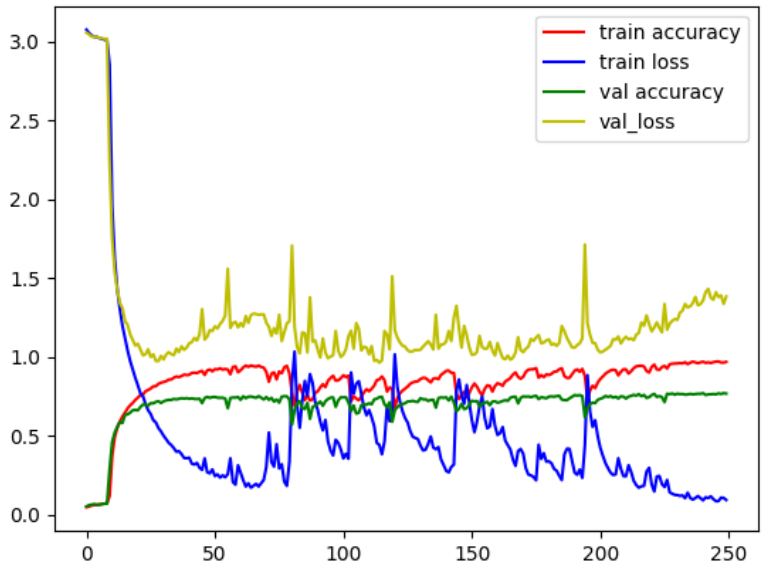


Figure 18: SOOR RNN training curve of reduced 22 scene classes. X-axis lists the number of epochs

The results of training and testing is shown in Table 6. We evaluate up to the top 3 predictions for the re-balanced test set using the ground truth bounding boxes and labels of objects from the SUN RGB-D dataset initially. These results are shown in Table 7. The results show that the model is capable of accurately predicting the top three classes that best describe the scene depicted in the input image on the re-balanced test set, with an accuracy of 76.9% for the top prediction, and 90.9% for the top three predictions. Achieving such high accuracies on the test set suggests that the SOOR 3D depth encoding approach provides sufficient data to classify scenes accurately, and that the model is capable of generalizing well to unseen data.

Table 6: Training results and evaluation of SOOR RNN

Training Accuracy	Validation Accuracy	Test Accuracy
78.7	70.9	65.7

Table 7: Top Predictions of SOOR RNN on SUN RGB-D

Top 1 Prediction	Top 2 Prediction	Top 3 Prediction
76.9	87.1	90.9

We compare the performance of scene classification on the SUN RGB-D on different individual scene classes. Classes and their top 3 prediction accuracies from test results are shown in Figure 19. The classes total were: 'bedroom', 'classroom', 'living_room' 'office', 'study_space', 'recreation_room', 'printer_room', 'bathroom', 'conference_room', 'corridor', 'dining_room', 'home_office', 'kitchen', 'discussion_area', 'office_kitchen', 'idk', 'dining_area', 'rest_space', 'library', 'computer_room', 'lab', 'lecture_theatre'. The 'idk' label indicates 'I don't know' for scenes that human dataset labelers were unable to classify.

Prediction Accuracy	bedroom	classroom	living_room	office	study_space	recreation_room	printer_room	bathroom	conference_room	corridor	dining_room
Top 1	62	59	69	56	79	99	98	97	67	82	76
Top 2	79	77	82	76	91	100	100	98	83	88	84
Top 3	86	87	87	81	93	100	100	99	87	92	89
Prediction Accuracy	home_office	kitchen	discussion_area	office_kitchen	idk	dining_area	rest_space	library	computer_room	lab	lecture_theatre
Top 1	87	78	67	95	78	75	54	78	86	91	85
Top 2	92	87	81	99	87	83	76	87	90	97	91
Top 3	94	91	84	100	89	86	82	88	95	97	93

Figure 19: Scene classification test results per class using ground truth class labels and centers

The results of the individual class prediction accuracy provide valuable insights into the performance of the proposed scene classification approach. It is not surprising that scenes such as printer rooms have a high prediction accuracy rate, as printers are often the defining object in such rooms. However, it is interesting to note that scenes such as dining area, conference room, and discussion area share common objects such as tables and chairs, which are prominent in the scene have slightly lower classification accuracies than those that do not share common objects. These results suggest that the proposed approach is capable of accurately identifying scenes based on the presence of identifier objects and is capable of identifying scenes with shared objects well but to a lesser degree of accuracy. While humans can often distinguish these scenes based on the placement of such objects, there may still be disagreements even between human classifiers as the functions of such rooms (dining area, conference room, and discussion area) can be performed in all of the listed scenes. This highlights the potential for ambiguity in scene classification, which should be considered in further development of framing the scene classification problem.

After validating and verifying the SOOR encoding approach in 3D, the entire pipeline was constructed beginning with generating region proposals from the 2D image with Faster R-CNN, then using the Frustum PointNets 3D object detector to classify and predict the object class and 3D bounding box, encoding this data into pairwise object-to-object relations, and finally passing the encoding into the scene classifier for final scene predictions. The data

that was used were the 2D images and 3D depth data in this pipeline. The provided ground truth bounding boxes for 2D and 3D were unseen. Table 8 shows top 3 prediction accuracies utilizing the entire pipeline.

Table 8: Top Predictions of full scene classification pipeline on SUN RGB-D

Top 1 Prediction	Top 2 Prediction	Top 3 Prediction
65.7	79.8	85.1

As anticipated, the accuracy of predictions on the full pipeline is lower. This is because the object detection results, as shown in the previous section, are not always 100% accurate for the SUN RGB-D dataset. Since the objects and their relationships are the basis of scene representation, imperfect object detection would inevitably lead to lower scene classification accuracy. The individual class results given by the full pipeline are shown in Figure 20.

Prediction Accuracy	bedroom	classroom	living_room	office	study_space	recreation_room	printer_room	bathroom	conference_room	corridor	dining_room
Top 1	62	59	69	56	79	99	98	97	67	82	76
Top 2	79	77	82	76	91	100	100	98	83	88	84
Top 3	86	87	87	81	93	100	100	99	87	92	89
Prediction Accuracy	home_office	kitchen	discussion_area	office_kitchen	idk	dining_area	rest_space	library	computer_room	lab	lecture_theatre
Top 1	87	78	67	95	78	75	54	78	86	91	85
Top 2	92	87	81	99	87	83	76	87	90	97	91
Top 3	94	91	84	100	89	86	82	88	95	97	93

Figure 20: Scene classification test results per class using Frustum Pointnet detection

Notably, certain classes decreased in their classification accuracy. Such classes include: bedroom, conference_room, dining_room, home_office, kitchen, discussion_area, office_kitchen, idk, dining_area, rest_space, library, computer_room, lab, and lecture_theatre. Although most classes decreased in accuracy by similar amounts, lecture_theatre and discussion_area's prediction accuracy decreased significantly. This may be caused by the common objects found in these scenes being both primarily tables and chairs. The amount of tables and chairs and their placements are what distinguishes a lecture theatre, discussion area, conference area from each other. In cases like these, the amount of chairs surrounding the table matters for prediction results. If detection was less accurate and there were more false negatives, the classifier model will tend to predict those scenes that generally has a lower amount of chairs. For example, if there were 10 chairs surrounding a table in a discussion area and the object detector detected 5 out of 10 chairs, the classifier would more likely predict dining area where 5 chairs surrounding a table is more likely. The difference in prediction accuracy from using the ground truth object instances in comparison to using a trained object detector proves

the significance that the quality of the object detector provides.

(a) "conference_room" class

(b) "classroom" class

Figure 21: Visualising the importance of spacial relations between objects in scene classes

Figure 21 shows two different scenes in the test data that contain the same objects in similar frequencies with different spatial layouts. Figure 21a show a conference room with tables organized in a U shape with chairs surrounding it while Figure 21b show 2 chairs behind each table. During object detection, the model detected 8 chairs and 4 tables for the conference room, and 7 chairs and 5 tables for the classroom. Both scenes have identical objects and almost identical frequencies of objects yet the scene classification module trained on 3D SOOR encodings predicted scene classes correctly as a top 1 prediction and distinguished each scene from the other. Clearly, the spatial relationships between objects play a critical role in determining the class of a scene especially when common objects such as tables and chairs appear in many different scenes.

Table 9: Comparison of state-of-the-art scene classification results

SOOR (2020)[11]	Cbcl (2020)[55]	ResNet101-RNN (2022)[56]	Us	OMNIVORE (2022)[57]
55.5	57.8	60.1	65.1	67.2

The results of the proposed method for scene classification are promising and comparable to state-of-the-art performance as shown in Table 9. This suggests that the 3D depth information and metric scale values provided by the SOOR 3D encoding can be effective in improving scene classification accuracy. Notably, the proposed method outperforms the original SOOR object-to-object scene encoding in 2D by around 10%. These results highlight the importance of incorporating 3D depth data in scene classification tasks and suggest that the proposed method can be used as an effective tool for scene analysis and understanding.

A model was trained using the same architecture as before but training inputs consisted only of object-object pairs to learn the co-occurrence of objects without spatial relationships. Table 10 show the results that followed. The model trained only on object co-occurrences performed slightly worse than the model trained on object-to-object pair relations. This implies the spatial relations between objects hold significance and provides useful information in determining the class of a scene. Additionally, purely object co-occurrence scene representations rely more on the object detector to detect objects present and classify them accurately. The amount of information that one object pair provides is significantly less than the information provided by the object-to-object relation encoding which includes not only the object classes present but their size and location relative to each other. This spatial relation implicitly describes the layout of the scene which has been proven by topological approaches to be enough information alone to classify scenes.

Table 10: Top Predictions of scene classification using co-occurrence on SUN RGB-D

Top 1 Prediction	Top 2 Prediction	Top 3 Prediction
62.3	67.7	79.5

Across the board however, scene classification accuracies are low for the SUN RGB-D dataset. This may be due to loose and ambiguous labeling and data obtained from many different camera sensors.

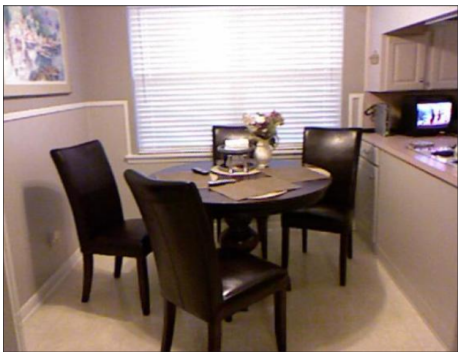
(a) 'rest_space' example 1

(b) 'rest_space' example 2

Figure 22: Examples of rest space where scene label is not obvious

Some examples of ambiguous labeling are shown in Figure 22 below with two examples of 'rest_space' yet both would be difficult to classify into a scene class individually. Both examples show examples of the same scene class but with drastically different objects and

use cases for such objects. In this case, the question would be to define what 'rest' meant in general. For a less ambiguous case such as 'bedroom', the function is to sleep in this scene and often this is performed with a bed. The existence of a bed in the room is to sleep on. However, for 'rest_space', a leisure piano playing session can be considered rest for some such as in Figure 22b and sitting at a table talking to a coworker such as in Figure 22a is rest for others.



(a) 'kitchen' example

(b) 'dining_area' example

Figure 23: Examples of loose labeling

There are points in the SUN RGB-D data where scene labels were disagreeable. Figure 23 show examples of labels that loosely describe the scene. Figure 23a is labeled 'kitchen' yet a more fitting scene label for this scene given the existing labels of the set would be 'dining_area' since most other 'dining_area' labels in the scene resemble this layout as shown in 23b.

(a) 'recreation_area' example

(b) 'living_room' example

Figure 24: Examples of ambiguous labeling

There are many examples of blurry definitions between scene labels as well. In Figure 24, both scenes appear as living rooms yet Figure 26a shows a 'recreation_room' and Figure

26b shows a 'living_room'. One could also argue for the label of 'rest_area' for both scenes as well.

The scene classification accuracies on the SUN RGB-D dataset are lower than desired across all studies. This could be attributed to several factors such as loose and ambiguous labeling, the diversity of data obtained from different camera sensors, and variations in lighting conditions and image quality. These factors can make it difficult for the model to generalize and accurately classify scenes. However, it should be noted that these challenges are also present in real-world scenarios and the dataset provides a valuable resource for testing and improving scene classification models under realistic conditions. Additionally, while the accuracies may be lower than desired, the proposed method's ability to accurately distinguish between scenes with similar objects and spatial layouts demonstrates its potential usefulness in real-world applications.

These examples of loose and ambiguous scene labeling highlight the overarching difficulty in scene classification. Even as humans, often we can disagree on scene labels and as such, many scenes perform similar functions with different labels. Despite this, the performance of the pipeline does well and when the top 1 prediction is incorrect, the predicted label would often be in an adjacent scene such as 'conference_room' and 'discussion_area'. The ambiguities in scene classes and diversity of scene appearances shed light on a potential overarching flaw in the way we define the scene classification problem and how we understand the way humans distinguish between scenes today.

The results of the SUN RGB-D dataset scene classification task highlight the challenges in defining and understanding the scene classification problem. As humans, we rely on contextual information, past experiences, and other sensory inputs beyond the visual appearance to distinguish between different scenes. However, defining and quantifying these contextual factors in a way that can be used for machine learning algorithms remains a significant challenge. Moreover, the diversity of appearances within a given scene class further compounds this challenge, as there may be significant variation in the appearance of objects and the spatial arrangement of objects within a given scene class. Addressing these challenges will require a multi-disciplinary approach that involves expertise in computer vision, cognitive science, and psychology to better understand the underlying mechanisms of human perception and the factors that influence scene understanding.

4.3 Home Scenes and Mobile Base Implementation

The mobile base described previously was used to test the proposed methods of scene classification in a mobile robotic setting. Testing robotic algorithms on a real robot is crucial in ensuring their effectiveness in real-world applications. While datasets can provide insights into the expected performance of an algorithm, it is often not enough to account for the complexities and unpredictability of real-world environments. By testing on a real robot, we can observe how the algorithm performs in real-time, identify potential errors and limitations, and make necessary adjustments to improve its functionality. This is especially important for further intended applications such as autonomous service robots where the consequences of a malfunctioning algorithm can be severe. Ultimately, testing on a real robot can help to ensure that robotic algorithms are reliable, accurate, and safe for use in various applications, thus accelerating the adoption of robotics technology in diverse fields.

The scene classification pipeline was used in conjunction with the ZED2 camera image and point cloud feed. Data was collected at York University which included two scenes of 'lab' and 'discussion_area'. Data was also collected from home environments including 'bedroom, 'dining_area', 'living_room', 'kitchen' and 'bathroom'. For the 'kitchen' class uniquely, classification was attempted when one object was detected as opposed to the 2 needed for object-to-object pair relation encoding. In this case, the object-to-object pair would be against itself. This was necessary in the kitchen case since the appliances in the real world kitchen set are more modern than the appliances in the dataset that the object detectors are trained on. An example is shown in Figure 25. Thus, often only one object would be detected.

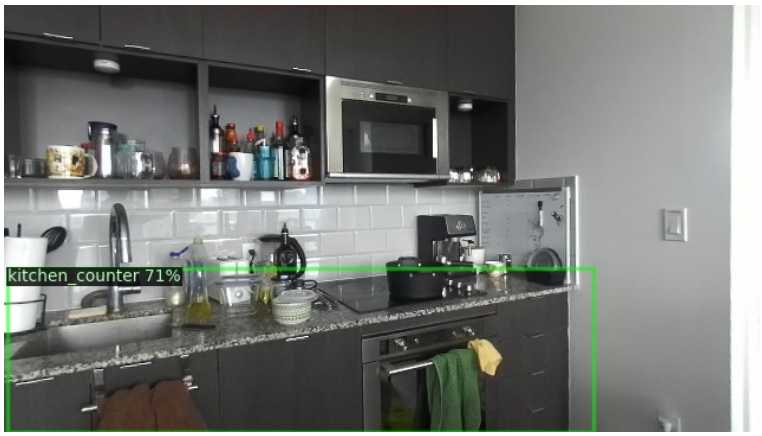


Figure 25: Kitchen with new appliances and only one detected object

In the case of good classification while only detecting one object, clearly the occurrence of certain objects can be enough to influence a valid decision on the class of a scene. This was also observed in scenes such as 'bathroom' where the existence of 'toilet' or 'bathtub' influenced the decision heavily regardless of other objects it co-occurred with whether it be 'sink', 'garbage_can', or 'basket'. However, when the camera moves to a view where the toilet or bathtub is not visible, the sink's co-occurrence and spatial relationship to the garbage can inspired less confidence in the scene and would have similar confidence rates between 'kitchen' and 'bathroom'.

The good classification results when only detecting one object highlights that the presence of certain objects can significantly influence the classification of a scene. Even when only one object is detected, a valid decision on the class of the scene can still be made. This was also observed in scenes such as 'bathroom' where the existence of 'toilet' or 'bathtub' influenced the decision heavily regardless of other objects it co-occurred with. When the co-occurrence of other objects like 'sink', 'garbage can', or 'basket' was observed, the decision was still heavily influenced by the presence of 'toilet' or 'bathtub'. This suggests that certain objects may have a stronger association with particular scene classes than others.

Moreover, the mobile classification results reveals that the camera's viewpoint and object visibility can affect the classification results. When the camera moves to a view where the 'toilet' or 'bathtub' is not visible, the decision on the scene class becomes less confident, and the classification rate may be similar between 'kitchen' and 'bathroom'. In such cases, the co-occurrence and spatial relationship between other objects such as 'sink' and 'garbage

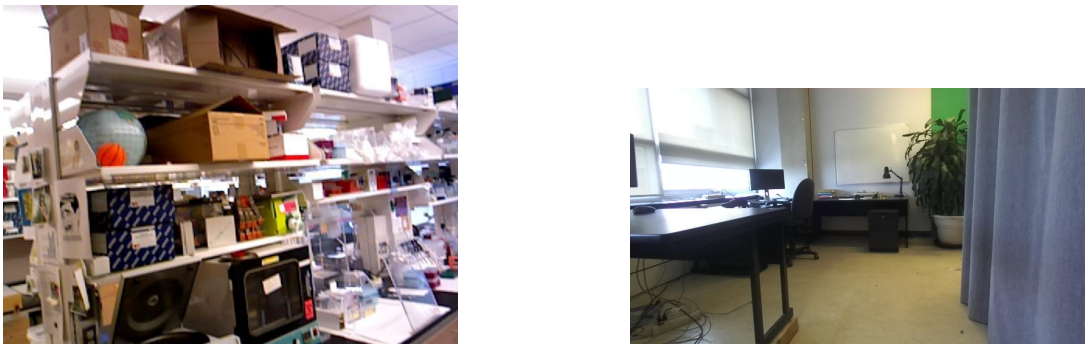
can' may not provide enough evidence to distinguish between the two different scene classes. Overall, the analysis demonstrates that the presence and visibility of specific objects play a crucial role in scene classification accuracy, and camera viewpoint and object visibility can significantly impact the classification results.

Table 11: Top Predictions of full scene classification pipeline on live data

Top 1 Prediction	Top 2 Prediction	Top 3 Prediction
72	76	87

The results of the study showed that the performance of the scene classification pipeline on real-time data was more accurate than its performance on the test set for the SUN RGB-D dataset, as presented in Table 11. The higher accuracy observed in the real-time data can be attributed to the better quality of images obtained from the ZED2 camera, which leads to improved object detection. Object detection is an essential component of the scene classification method used in this study, and better object detection contributes to more accurate scene classification.

The results also indicate that the approach used in this study is applicable to different stereo-camera sensors and is likely to perform better with newer and more advanced cameras and object detection techniques. This is because the scene classifiers primarily use object-to-object pair spatial relations, which are not dependent on the precise point-clouds and high-quality cameras required by other state-of-the-art methods for feature extraction. The generalizability of the approach makes it more practical for real-world applications, where different types of cameras and sensors may be used.



(a) 'lab' example from SUN RGB-D (b) 'lab' example from YorkU

Figure 26: Difference between the 'lab' label in SUN RGB-D and the lab data points captured

During the testing phase, experiments were conducted on a mobile robot at York University to evaluate the performance of the scene classification pipeline. The data gathered for the 'lab' scene did not resemble a typical 'lab' scene in the training dataset. As shown in Figure 26, there were notable differences between the labs in the dataset and the live lab data collected. The scene classifier identified these scenes as 'office' and 'computer_room' more than 'lab', which is not surprising given the dissimilarities between the scenes. This underscores the importance of ensuring that the training dataset is representative of the target environment and that the model is tested on a diverse set of scenes to ensure its robustness and generalizability.

Figure 27: Discussion area at York University

The scene classification pipeline utilized in this study encountered difficulties in accurately classifying scenes that have multiple uses. This was observed in the case of the discussion area in Figure 27, where the scene classifier often identified the space as a 'dining_area', 'rest_area', or 'conference_room', all of which were considered correct since the area serves multiple purposes. The same issue was encountered in the case of the new test data for 'bedroom' in Figure 28, where the space was used both as a bedroom and an office, and the scene classifier would often label it as both in its top 2 predictions. This mixture of use cases often occurred in the training data as well, which explains the lower accuracies in their respective scene classes during testing.

Figure 28: Bedroom in collected data

These two instances highlight the difficulties of accurately classifying scenes that have multiple uses, as the context and intent of the person identifying the space changes the likely label they use for the scene. In such cases, it is important to consider the different potential labels for the scene based on its various uses and consider each of them as a potential classification outcome. This issue should be addressed in future studies by considering more complex scene class labels that are better suited to handle multiple-use scenes.

5 Conclusion and Future Work

Scene classification is a task in computer vision that involves categorizing an image into one of several predefined scene categories. The goal is to teach machines to recognize and understand the content of an image, which can have numerous practical applications and in particular understanding the contexts in which objects occur in for search and retrieval tasks for mobile service robots. By utilizing 3D point cloud data and 3D object detection, we propose an end-to-end method to classify scenes using novel scene representation with the intent of encapsulating the spatial relationship information between objects and their relation to scene classes. This method utilize objects and their likelihood to not only co-occur but to appear with an expected spatial relationship. Firstly, we detect 2D object regions of interest using a fine-tuned Faster R-CNN model which is passed into a Frustum PointNets model for 3D object classification and 3D box predictions. These object classes and boxes are used to encode the object-to-object spatial relations that represent a scene in the form of sequential representations. Without the limitations of fixed data structure, richer types of relations such as extended directional relations, distance and area are encoded from 3D metric values. Finally, these scene encodings are used to predict the scene class.

The proposed method achieves comparable results of 65.5% accuracy to state-of-the-art methods evaluated on the SUN RGB-D dataset using only object class and spatial information. It was found that utilizing 3D object-to-object relations greatly improves scene classification compared to previous studies where only 2D object relations were considered and performs better than using only object co-occurrence data which highlights the importance of 3D object spatial relations in distinguishing scene classes. Many ambiguities with scene classes and labels result in lower accuracies in performance but demonstrates the challenges that the scene classification task comes with.

The results from this thesis work highlight the importance of incorporating 3D object and spatial relational data in scene classification tasks and suggest that the proposed method can be used as an effective tool for scene analysis and understanding. However, it should be noted that there is still room for improvement in terms of accuracy and scalability, and further research is needed to investigate the potential of incorporating other types of data and feature extraction techniques such as utilizing feature interest points from a CNN used

in current state-of-the-art leading methods in scene classification.

The results of the SUN RGB-D dataset scene classification task demonstrate the difficulty of defining and understanding the scene classification problem. The limitations in current machine learning algorithms to replicate human perception and contextual factors in distinguishing between different scenes highlight the need for a multi-disciplinary approach to solve the issue. The variation in appearance and spatial arrangement of objects within a given scene class add further complexity to the problem. Addressing these challenges will require knowledge in computer vision, cognitive science, and psychology to develop a more comprehensive understanding of human perception and the factors that contribute to scene understanding. Overall, this study provides valuable insights into the difficulties of scene classification and highlights the need for further research in this area.

A future extension of this work can be to explore higher levels of object-to-object relations from object pairs to object constellations where objects and the other objects they occur with are represented in a 3D spatial graph. In this way, objects that occur in scenes may be represented in a more complex manner and their relations to other objects in the room and their relations may influence and aid final scene predictions. When humans see objects in a room, they are not just considering relations between object pairs to classify the function of a scene. They consider groups of objects and their relationship and function together. Scenes may include many of these object groups and scenes may include just one object group that greatly influences the class and function of the scene. Representing objects in this way can provide more context for the scene and its classification. Graph convolutional networks (GCNs) can be utilized to represent objects in a scene and their relationships with each other. By representing objects as nodes in a graph and using GCNs to model the relationships between them, the resulting feature representation can capture the dependencies between objects and their spatial arrangement in the scene. This approach can lead to improved accuracy in scene classification, as demonstrated by recent studies on outdoor satellite images using scenes such as school, baseball field, etc as nodes to classify a larger 2D satellite image[76]. This work provides grounds to reason that an extension to indoor scenes using 3D objects as nodes as scene representations can be a viable approach to scene classification.

In future research, utilizing the camera mount's ability to pan and tilt can be a

valuable feature to enhance scene classification performance. This feature provides multiple viewpoints of the same scene, which can aid in detecting objects that may have been obscured in other views, as well as increasing the density of the point cloud spatial map. Mimicking the active vision that humans use while observing a scene, a pan-tilt unit can provide more useful information and the ability to focus on certain interest points in a room, which can improve scene classification accuracy. By incorporating these features, future research can leverage the power of active vision to better model human perception and improve the performance of scene classification algorithms.

Scene classification is a challenging task in computer vision that requires incorporating various features such as object co-occurrences and spatial relations to teach machines to recognize and understand the content of an image. This study proposes an end-to-end method using 3D object detection and spatial relations that performs better than previous studies with similar approaches, highlighting the importance of 3D object spatial relations in distinguishing scene classes. However, further research is needed to improve accuracy by exploring higher levels of object-to-object relations and utilizing camera mounts' ability to pan and tilt to mimic active vision. Addressing the challenges of fully imitating human scene classification will require a multi-disciplinary approach to solve the issue, including knowledge in computer vision, cognitive science, and psychology, to develop a more comprehensive understanding of human perception and the factors that contribute to scene understanding. Ultimately, this study provides valuable insights into key aspects and challenges of scene classification and paves the way for further research in this area.

References

[1] T. A. Patel, V. K. Dabhi, and H. B. Prajapati, "Survey on scene classification techniques," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2020, pp. 452–458. DOI: 10.1109/ICACCS48705.2020.9074460.

[2] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, 2018. DOI: 10.1109/TPAMI.2017.2723009.

[3] T. D. Garvey, "Perceptual strategies for purposeful vision," 2011.

[4] D. Bhardwaj and V. Todwal, *A survey on indoor-outdoor scene classification with deep learning techniques*, Dec. 2020. [Online]. Available: https://www.ijntr.org/download_data/IJNTR06120034.pdf.

[5] L. Xie, F. Lee, L. Liu, K. Kotani, and Q. Chen, "Scene recognition: A comprehensive survey," *Pattern Recognition*, vol. 102, p. 107 205, 2020, ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2020.107205. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S003132032030011X.

[6] A. Pronobis, "Semantic mapping with mobile robots," 2011.

[7] D. Zeng, M. Liao, M. Tavakolian, *et al.*, "Deep learning for scene classification: A survey," *CoRR*, vol. abs/2101.10531, 2021. arXiv: 2101.10531. [Online]. Available: https://arxiv.org/abs/2101.10531.

[8] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, *Frustum pointnets for 3d object detection from rgb-d data*, 2017. DOI: 10.48550/ARXIV.1711.08488. [Online]. Available: https://arxiv.org/abs/1711.08488.

[9] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *CoRR*, vol. abs/1612.00593, 2016. arXiv: 1612.00593. [Online]. Available: http://arxiv.org/abs/1612.00593.

[10] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 567–576. DOI: 10.1109/CVPR.2015.7298655.

[11] X. Song, S. Jiang, B. Wang, C. Chen, and G. Chen, "Image representations with spatial object-to-object relations for rgb-d scene recognition," *IEEE Transactions on Image Processing*, vol. 29, pp. 525–537, 2020. DOI: 10.1109/TIP.2019.2933728.

[12] Y. Ke and R. Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," vol. 2, May 2004, pp. II–506, ISBN: 0-7695-2158-4. DOI: 10.1109/CVPR.2004.1315206.

[13] X. Wang, T. X. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 32–39. DOI: 10.1109/ICCV.2009.5459207.

[14] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008, Similarity Matching in Computer Vision and Multimedia, ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2007.09.014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314207001555.

[15] M. J. Shafiee, B. Chywl, F. Li, and A. Wong, "Fast YOLO: A fast you only look once system for real-time embedded object detection in video," *CoRR*, vol. abs/1709.05943, 2017. arXiv: 1709.05943. [Online]. Available: http://arxiv.org/abs/1709.05943.

[16] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *CoRR*, vol. abs/1612.08242, 2016. arXiv: 1612.08242. [Online]. Available: http://arxiv.org/abs/1612.08242.

[17] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018. arXiv: 1804.02767. [Online]. Available: http://arxiv.org/abs/1804.02767.

[18] W. Liu, D. Anguelov, D. Erhan, *et al.*, "SSD: single shot multibox detector," *CoRR*, vol. abs/1512.02325, 2015. arXiv: 1512.02325. [Online]. Available: http://arxiv.org/abs/1512.02325.

[19] C. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD : Deconvolutional single shot detector," *CoRR*, vol. abs/1701.06659, 2017. arXiv: 1701.06659. [Online]. Available: http://arxiv.org/abs/1701.06659.

[20] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, vol. abs/1708.02002, 2017. arXiv: 1708.02002. [Online]. Available: http://arxiv.org/abs/1708.02002.

[21] Y. Cabon, N. Murray, and M. Humenberger, "Virtual KITTI 2," *CoRR*, vol. abs/2001.10773, 2020. arXiv: 2001.10773. [Online]. Available: https://arxiv.org/abs/2001.10773.

[22] T. Lin, M. Maire, S. J. Belongie, *et al.*, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. arXiv: 1405.0312. [Online]. Available: http://arxiv.org/abs/1405.0312.

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *CoRR*, vol. abs/2010.11929, 2020. arXiv: 2010.11929. [Online]. Available: https://arxiv.org/abs/2010.11929.

[24] A. Ghasemieh and R. Kashef, "3d object detection for autonomous driving: Methods, models, sensors, data, and challenges," *Transportation Engineering*, vol. 8, p. 100 115, 2022, ISSN: 2666-691X. DOI: https://doi.org/10.1016/j.treng.2022.100115. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666691X22000136.

[25] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, *Deep learning for 3d point clouds: A survey*, 2019. DOI: 10.48550/ARXIV.1912.12033. [Online]. Available: https://arxiv.org/abs/1912.12033.

[26] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, *Multi-view 3d object detection network for autonomous driving*, 2016. DOI: 10.48550/ARXIV.1611.07759. [Online]. Available: https://arxiv.org/abs/1611.07759.

[27] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, *Multi-task multi-sensor fusion for 3d object detection*, 2020. DOI: 10.48550/ARXIV.2012.12397. [Online]. Available: https://arxiv.org/abs/2012.12397.

[28] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, *Ipod: Intensive point-based object detector for point cloud*, 2018. DOI: 10.48550/ARXIV.1812.05276. [Online]. Available: https://arxiv.org/abs/1812.05276.

[29] S. Shi, X. Wang, and H. Li, *Pointrcnn: 3d object proposal generation and detection from point cloud*, 2018. DOI: 10.48550/ARXIV.1812.04244. [Online]. Available: https://arxiv.org/abs/1812.04244.

[30] X. Zhao, Z. Liu, R. Hu, and K. Huang, "3d object detection using scale invariant and feature reweighting networks," ser. AAAI'19/IAAI'19/EAAI'19, Honolulu, Hawaii, USA: AAAI Press, 2019, ISBN: 978-1-57735-809-1. DOI: 10.1609/aaai.v33i01.33019267. [Online]. Available: https://doi.org/10.1609/aaai.v33i01.33019267.

[31] B. Yang, W. Luo, and R. Urtasun, "PIXOR: real-time 3d object detection from point clouds," *CoRR*, vol. abs/1902.06326, 2019. arXiv: 1902.06326. [Online]. Available: http://arxiv.org/abs/1902.06326.

[32] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network," *CoRR*, vol. abs/1608.07916, 2016. arXiv: 1608.07916. [Online]. Available: http://arxiv.org/abs/1608.07916.

[33] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "STD: sparse-to-dense 3d object detector for point cloud," *CoRR*, vol. abs/1907.10471, 2019. arXiv: 1907.10471. [Online]. Available: http://arxiv.org/abs/1907.10471.

[34] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang, "Structure aware single-stage 3d object detection from point cloud," Jun. 2020, pp. 11 870–11 879. DOI: 10.1109/CVPR42600.2020.01189.

[35] N. Ali and B. Zafar, "15-Scene Image Dataset," Aug. 2018. DOI: 10.6084/m9.figshare.7007177.v1. [Online]. Available: https://figshare.com/articles/dataset/15-Scene_Image_Dataset/7007177.

[36] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 413–420. DOI: 10.1109/CVPR.2009.5206537.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[38] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27, Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/3fe94a002317b5f9259f82690aeea4cd-Paper.pdf.

[39] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, 2018. DOI: 10.1109/TPAMI.2017.2723009.

[40] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Computer Vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 746–760, ISBN: 978-3-642-33715-4.

[41] A. Bosch, X. Muñoz, and R. Martí, "Which is the best way to organize/classify images by content?" *Image and Vision Computing*, vol. 25, no. 6, pp. 778–791, 2007, ISSN: 0262-8856. DOI: https://doi.org/10.1016/j.imavis.2006.07.015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885606002253.

[42] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 2006, pp. 2169–2178. DOI: 10.1109/CVPR.2006.68.

[43] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 413–420. DOI: 10.1109/CVPR.2009.5206537.

[44] A. Bassiouny and M. El-Saban, "Semantic segmentation as image representation for scene recognition," in *2014 IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 981–985. DOI: 10.1109/ICIP.2014.7025197.

[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, ISSN: 0001-0782. DOI: 10.1145/3065386. [Online]. Available: https://doi.org/10.1145/3065386.

[46] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27, Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/3fe94a002317b5f9259f82690aeea4cd-Paper.pdf.

[47] L. Herranz, S. Jiang, and X. Li, "Scene recognition with cnns: Objects, scales and dataset bias," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 571–579, 2016.

[48] S. H. Khan, M. Hayat, M. Bennamoun, R. Togneri, and F. A. Sohel, "A discriminative representation of convolutional features for indoor scene recognition," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3372–3383, 2016. DOI: 10.1109/TIP.2016.2567076.

[49] S. Liu, G. Tian, and Y. Xu, "A novel scene classification model combining resnet based transfer learning and data augmentation with a filter," *Neurocomputing*, vol. 338, pp. 191–206, 2019, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.01.090. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231219301833.

[50] Y. Liu, Q. Chen, W. Chen, and I. Wassell, "Dictionary learning inspired deep network for scene recognition," ser. AAAI'18/IAAI'18/EAAI'18, AAAI Press, 2018, ISBN: 978-1-57735-800-8.

[51] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," *CoRR*, vol. abs/1512.04150, 2015. arXiv: 1512.04150. [Online]. Available: http://arxiv.org/abs/1512.04150.

[52] N. Sun, W. Li, J. Liu, G. Han, and C. Wu, "Fusing object semantics and deep appearance features for scene recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 6, pp. 1715–1728, 2019. DOI: 10.1109/TCSVT.2018.2848543.

[53] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene cnns," *CoRR*, vol. abs/1412.6856, 2014.

[54] L. Liu, J. Chen, P. W. Fieguth, G. Zhao, R. Chellappa, and M. Pietikäinen, "A survey of recent advances in texture representation," *CoRR*, vol. abs/1801.10324, 2018. arXiv: 1801.10324. [Online]. Available: http://arxiv.org/abs/1801.10324.

[55] A. Ayub and A. R. Wagner, "Centroid-based scene classification (CBSC): using deep features and clustering for RGB-D indoor scene classification," *CoRR*, vol. abs/1911.00155, 2019. arXiv: 1911.00155. [Online]. Available: http://arxiv.org/abs/1911.00155.

[56] A. Caglayan, N. Imamoglu, A. B. Can, and R. Nakamura, "When cnns meet random rnns: Towards multi-level analysis for rgb-d object and scene recognition," *Computer Vision and Image Understanding*, vol. 217, p. 103 373, 2022, ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2022.103373.

[57] R. Girdhar, M. Singh, N. Ravi, L. van der Maaten, A. Joulin, and I. Misra, "Omnivore: A single model for many visual modalities," *CoRR*, vol. abs/2201.08377, 2022.

[58] A. Mosella-Montoro and J. R. Hidalgo, "2d-3d geometric fusion network using multi-neighbourhood graph convolution for RGB-D indoor scene classification," *CoRR*, vol. abs/2009.11154, 2020.

[59] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 328–335. DOI: 10.1109/CVPR.2014.49.

[60] B. X. Chen, R. Sahdev, D. Wu, X. Zhao, M. Papagelis, and J. K. Tsotsos, "Scene classification in indoor environments for robots using context based word embeddings," 2019. DOI: 10.48550/ARXIV.1908.06422. [Online]. Available: https://arxiv.org/abs/1908.06422.

[61] X. Song, S. Jiang, L. Herranz, and C. Chen, "Learning effective rgb-d representations for scene recognition," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 980–993, 2019. DOI: 10.1109/TIP.2018.2872629.

[62] Z. Xiong, Y. Yuan, and Q. Wang, "Msn: Modality separation networks for rgb-d scene recognition," *Neurocomputing*, vol. 373, pp. 81–89, 2020, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.09.066. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231219313347.

[63] D. Du, L. Wang, H. Wang, K. Zhao, and G. Wu, "Translate-to-recognize networks for rgb-d scene recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2019, pp. 11 828–11 837. DOI: 10.1109/CVPR.2019.01211. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.01211.

[64] S. Huang, M. Usvyatsov, and K. Schindler, "Indoor scene recognition in 3d," *CoRR*, vol. abs/2002.12819, 2020. arXiv: 2002.12819. [Online]. Available: https://arxiv.org/abs/2002.12819.

[65] M. Awan, M. Rahim, N. Salim, M. Mohammed, B. Zapirain, and K. Abdulkareem, "Efficient detection of knee anterior cruciate ligament from magnetic resonance imaging using deep learning approach," *Diagnostics*, vol. 11, p. 105, Jan. 2021. DOI: 10.3390/diagnostics11010105.

[66] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[67] E. Fazl-Ersi and J. K. Tsotsos, "Histogram of oriented uniform patterns for robust place recognition and categorization," *The International Journal of Robotics Research*, vol. 31, no. 4, pp. 468–483, 2012. DOI: 10.1177/0278364911434936. eprint: https://doi.org/10.1177/0278364911434936. [Online]. Available: https://doi.org/10.1177/0278364911434936.

[68] P. Espinace, T. Kollar, N. Roy, and A. Soto, "Indoor scene recognition by a mobile robot through adaptive object detection," *Robotics and Autonomous Systems*, vol. 61,

no. 9, pp. 932–947, 2013, ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2013.05.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889013000821.

[69] J. Xiao, A. Owens, and A. Torralba, "Sun3d: A database of big spaces reconstructed using sfm and object labels," in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 1625–1632. DOI: 10.1109/ICCV.2013.458.

[70] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015.

[71] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013.

[72] R. B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015.

[73] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, *Detectron2*, https://github.com/facebookresearch/detectron2, 2019.

[74] A. Janoch, S. Karayev, Y. Jia, *et al.*, "A category-level 3d object dataset: Putting the kinect to work," in *Consumer Depth Cameras for Computer Vision: Research Topics and Applications*, A. Fossati, J. Gall, H. Grabner, X. Ren, and K. Konolige, Eds. London: Springer London, 2013, pp. 141–165, ISBN: 978-1-4471-4640-7. DOI: 10.1007/978-1-4471-4640-7_8. [Online]. Available: https://doi.org/10.1007/978-1-4471-4640-7_8.

[75] *Capture the world in 3d*. [Online]. Available: https://www.stereolabs.com/.

[76] J. Liang, Y. Deng, and D. Zeng, "A deep neural network combined cnn and gcn for remote sensing scene classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4325–4338, 2020. DOI: 10.1109/JSTARS.2020.3011333.