2023

Enhancing Indoors Robotic Traversability Estimation with Sensor Fusion

Christos Sevastopoulos

ENHANCING INDOORS ROBOTIC TRAVERSABILITY ESTIMATION WITH SENSOR FUSION

by

CHRISTOS SEVASTOPOULOS

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
December 2023

To the ones who believe in me.

ACKNOWLEDGEMENTS

I would like to thank my supervising professors Dr. Fillia Makedon, Dr. Stasinos Konstantopoulos and Dr.Karkaletsis, for their guidance, patience and support. I would also like to thank my Ph.D. Committee Members, Dr.Ming Li and Dr.William Beksi. Finally, my labmates for their fruitful feedback and their valuable contribution to my work.

Monday, October 30, 2023

ABSTRACT

ENHANCING INDOORS ROBOTIC TRAVERSABILITY ESTIMATION WITH SENSOR FUSION

Christos Sevastopoulos , Ph.D.

The University of Texas at Arlington, 2023

Supervising Professor: Fillia Makedon

Generally speaking, traversability estimation illustrates the ability to navigate or move through a particular environment (indoors or outdoors). Indoor environments are governed by uncertainty and stochasticity arising from their complex structures encapsulating both static elements like furniture and walls, as well as entities such as moving humans. In our research, we underline the importance of blending semantic and spatial information for ensuring secure navigation for a mobile robot. We show that RGB sensors suffer from constrained situational awareness of the surroundings, thus highlighting the need to incorporate spatial and geometric data, which can collaborate synergistically to enhance overall perception and safety. Towards this direction, we examine indoors traversability estimation both on higher-level (GO/NO-GO decision) and also at a lower-level by identifying free-space zones that the robot can safely traverse.

We combine visual data (RGB) and Laser Range Finder (LRF) information both for annotating our dataset but also for enhancing the prediction compared to exclusive reliance on RGB information. In the core of our experiments, we use Transformer-

based architectures [1] due to 1) their efficacy in capturing spatial dependencies and sequences of varying lengths, which are common in indoor environments where objects are positioned in relation to each other 2) their notable transfer learning potential, since we are fine-tuning on our custom collected dataset and we need rich pre-trained features from a large scale dataset 3) their significant ability to handle multi-modal input sequences, since we are using different modalities.

We investigate the efficiency of employing a Multi-Head Self-Attention module as a fusion mechanism, leveraging its capability to assign varying weights across the input sequence. Ultimately, in order to estimate free-space, we employed a methodology predicated on the assumption that, larger depth values correspond to regions that the robot can safely traverse. Specifically, we implemented an efficient automated masking technique that leverages textural homogeneity, depth uniformity, and positive scenes to create meaningful segments before fine-tuning on our dataset. Applications of this work can be found in the following domains: 1) Navigation of autonomous agents or Mobility-impaired subjects 2) Safety in confined spaces such as warehouse/vineyard patrol robots 3) Search & Rescue applications.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In leading-edge mobile robotics research, a wide array of outdoor navigation applications such as planetary exploration, military operations, agricultural tasks etc. entails the necessity of adapting to the conditions encountered and, in particular, to address the challenges imposed by the terrain's contextual complexity. For every mobile robot, it is of indispensable importance to be able to identify its surroundings and translate the information perceived by its sensors to a meaningful volume of required knowledge. Subsequently, it will obtain the capacity to determine whether it can navigate in a safe while efficient manner. A vital part of autonomous navigation implies that the perceived structure of the environment has to be precisely illustrated in order to ensure whether a specific region can be traversed or not. Uneven terrains characterized by dense vegetation, foliage and potential presence of obstacles create a fundamental need to choose carefully between *proprioceptive* and *exteroceptive* sensors: Proprioceptive sensors measure values regarding the state of the robot itself (such as encoder clicks and joint angles) while exteroceptive sensors measure values from the environment (such as temperature and distance). Therefore, it is apparent that the selection of the sensors used has to be in accordance with each application's prerequisites and the robot's structural design.

1.1 Introduction

Traversability illustrates the difficulty of moving through a specific region and encompasses the suitability of the terrain for traverse based on its physical properties,

such as slope, roughness, surface condition [2], as well as the mechanical characteristics and capabilities of the robot. Furthermore, it might also establish a bedrock for path planning algorithms since it is inevitably incorporated in terrain indices (such as terrain roughness, terrain inclination) which are of central importance when considering the optimal path [3].

Although traversability estimation was initially framed as a binary classification problem [4], currently it can be viewed through the prism of multiple classes categorization with respect to the levels of traverse facilitation. Being able to evaluate terrains' traversability is a constitutional step towards designing a perception system [5] for such rough and rugged terrains while processing voluminous data acquired by different sensory techniques.

The emergence of *Deep Computer Vision* has expanded the field of traversability estimation, enabling the detection of features that traditional geometry-based approaches cannot access. It involves estimating compliance from visual information. While geometrically, the presence of obstacles might suggest non-traversable paths, a robot may still be able to navigate through compliant obstructions such as grass and foliage. Therefore, it becomes crucial to explore whether deep learning vision techniques can enhance environmental perception by adding semantic information to geometric data. Numerous studies highlight the significance of traversability analysis as a fundamental step in motion planning. Papadakis [4] provides a comprehensive review of how multi-sensor data acquisition, incorporating laser, stereo, color information, and accurate representation of vehicle-terrain interaction, can lead to meaningful traversability estimation in structured and unstructured environments. On the other hand, Kostavelis & Gasteratos [6] discuss various methods for extracting semantic mapping information and their potential applications in mobile robotics, including traversability assessment.

Figure 1.1 presents an overview of the contemporary advances in traversability estimation through the prism of deep learning techniques as well as conventional machine learning and non-trainable methods. Inferring terrain's traversability from geometric information can frequently bump into limitations as a consequence of the problems' high dimensionality while meaningful information is extracted from image data.

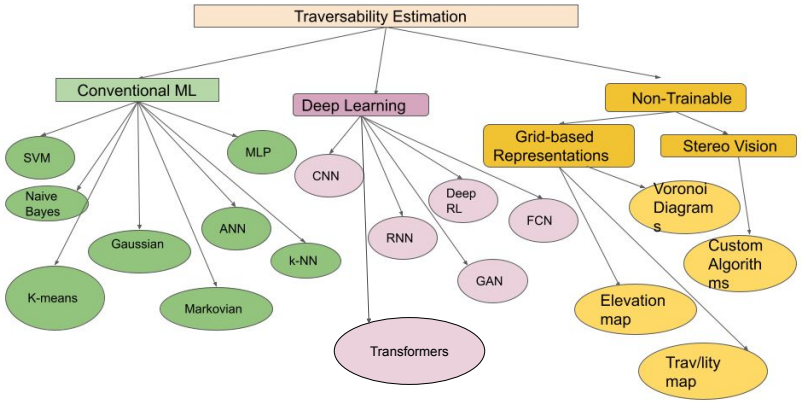


Figure 1.1: Overview of traversability estimation methods

On the grounds that data-driven methods can create structured representations derived solely by the available data and thus do not require the personal expertise of a human expert when it comes to accurate labelling and careful features' selection, it is arguable that they are favorable methods when handling large-scale data. However, both human-engineered and data-driven methods exhibit certain advantages and limitations 1.2 [7].

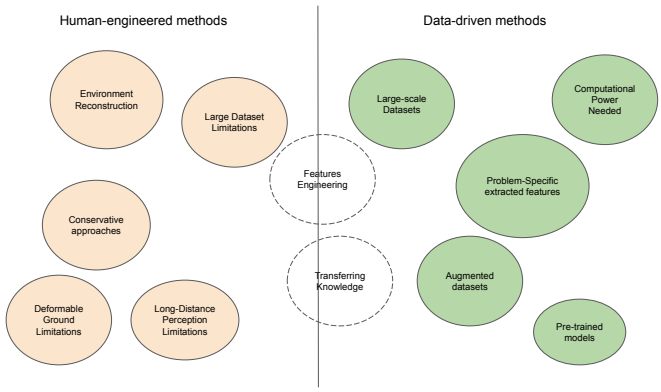


Figure 1.2: Facilitations and limitations between conventional vision and machine learning/deep learning techniques

1.1.1 Challenges encountered in traversability estimation scenarios

Traversability estimation techniques, which are indispensable in the fields of robotics and autonomous navigation, confront a plethora of formidable challenges. These challenges encompass the limitations inherent to sensors, including their range, field of view, and susceptibility to noise and calibration errors. The presence of dynamic environments, characterized by moving objects and people, introduces complexity and uncertainty into the estimation process. Ambiguity can further confound the task in intricate terrains, where distinguishing between traversable and non-traversable areas becomes arduous. Adapting to diverse terrains, different scales, and the computational intricacies involved adds additional layers of complexity.

The integration of data from a variety of sensors to achieve accurate estimations, effective training of machine learning models, and the need to account for environmental variability are substantial hurdles in this endeavor. Moreover, ensuring safety, assessing risks, and establishing rigorous validation standards remain pivotal concerns in this field. The omnipresent real-time constraints impose demands on the efficiency and reliability of traversability estimation techniques, making these challenges all the more crucial to address. In Figure 1.3 we notice that challenges related to the envi-

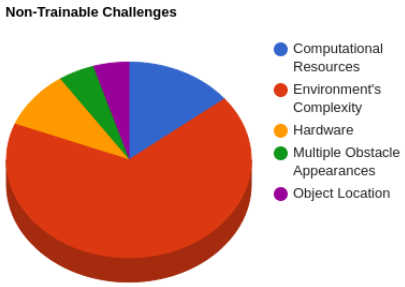


Figure 1.3: Challenges in non-trainable methods

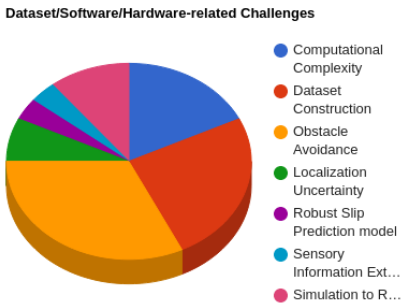


Figure 1.4: Dataset/Software/Hardware Challenges in ML,DL methods

ronment's complexity as well as the inadequacy of computational resources have been predominantly faced in non-trainable methods.

1.1.2 Challenges in Indoor Traversability Estimation

Indoor traversability estimation presents a multifaceted set of challenges, each of which plays a crucial role in the complexity of this field.

1. **Diverse Indoor Environments**: Indoor spaces exhibit remarkable diversity, ranging from cluttered and densely populated areas to meticulously structured office layouts with open spaces in between. Navigating these distinct environ-

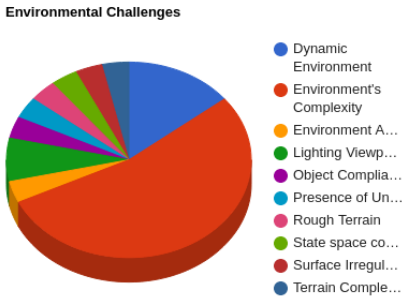


Figure 1.5: Environmental Challenges in ML,DL methods

ments necessitates adaptable algorithms capable of addressing varying degrees of complexity.

2. **Dynamic Environmental Factors**: The presence of dynamic elements, such as moving humans and objects, adds an inherent stochastic nature to indoor environments. Predicting and responding to these dynamic changes poses a significant challenge for traversability estimation systems.
3. **Fluctuating Lighting Conditions**: Lighting conditions indoors can fluctuate dramatically. Spaces may transition from well-lit areas to dimly lit or shadowy corners. These variations directly affect the performance of sensors and perception systems, demanding robust methods capable of handling diverse lighting scenarios.
4. **Reflective Surfaces and Obstacles**: Indoor spaces often feature reflective surfaces, including glass doors, mirrors, and polished floors. These surfaces introduce complexities in sensor data interpretation, requiring specialized algorithms to distinguish between obstacles and reflective elements.

Given these challenges, it becomes apparent that the integration of semantic and spatial information is of paramount importance. Combining these two dimensions

offers a promising avenue to enhance indoor traversability estimation, enabling more reliable and adaptable navigation in the diverse and dynamic indoor world.

1.1.3 Motivation & Significance of this Study

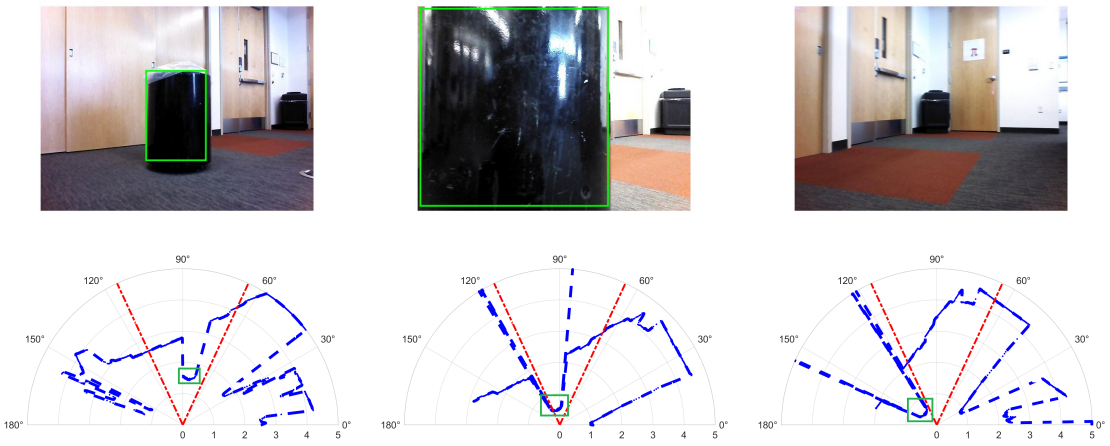


Figure 1.6: A sequence of frames depicting the importance of RGB and LRF fusion. Upper row, right-most image: Due to the camera's limited situation awareness, this frame is erroneously labeled as traversable, despite the presence of an obstacle on the left. Lower row: The LRF has wider FoV (blue graph) over the camera's FoV (red line), and thus it captures the obstacle's position (green box) consistently.

In this study, we aim to address several critical questions. First and foremost, we investigate whether relying solely on RGB information is sufficient for accurately estimating the traversability of a given scene. Additionally, we delve into the limitations associated with using RGB data exclusively. Specifically, we explore scenarios where obstacles are situated outside the field of view of the camera. In such case, relying solely on the RGB sensor, offers limited situation awareness in cases where the robot is placed very close to an obstacle and needs to perform a turn to avoid it (Figure 1.6). Driven by such limitation in surroundings' awareness, we highlight that perception is affected and certain methodological intervention may be necessary.

Therefore, the necessity of incorporating spatial and geometric information, along with the RGB, becomes apparent.

At a more granular level, it is imperative to identify the pre-trained features that yield valuable insights for the downstream task. Specifically, the tasks we consider in this study are 1) binary indoor traversability estimation and 2) free-space segmentation. We also investigate the methods for integrating semantic and spatial information, along with exploring the potential of depth information in representing open spaces.

Overall, the methodology presented expands along the following axes:

1) **RGB Binary Image Classification**: To explore the effectiveness of using only RGB images to distinguish between traversable (GO) and non-traversable (NO-GO) scenes.

2) **RGB-Laser Combination for Annotation**: To investigate the potential of combining RGB and Laser sensors for more accurate annotation of indoor environments. This dual-sensor approach enhances the quality of the labeled data used in the trainable method.

3) **RGB-Laser Fusion for Binary Prediction**: To improve the accuracy of traversability prediction. By integrating data from these two sources, this approach aims to provide a more comprehensive understanding of the environment by leveraging both semantic and spatial information.

4) **RGB-D for Free-Space Segmentation**: To perform free-space estimation by leveraging RGB and depth information.

In the core of our experiments, we employ transformer-based architectures for the following reasons:

- Their efficacy in capturing **spatial dependencies and sequences of varying lengths**, which are common in indoor environments where objects are positioned in relation to each other.
- Their potential for **transfer learning**, as we fine-tune them on our custom-collected dataset.
- Their significant ability to **handle multi-modal input sequences**.

Through the aforementioned viewpoints, this study aims to advance the state-of-the-art in indoor traversability estimation and free-space segmentation, ultimately contributing to safer and more efficient navigation for a mobile robot.

1.1.4 Thesis Outline

The structure of this thesis is as follows:

Chapter 2 presents the latest advancements in traversability estimation. We scrutinize the diverse array of sensors employed, ranging from traditional to cutting-edge technologies. Furthermore, we explore a spectrum of methods, encompassing both trainable and non-trainable approaches, and examine the range of robotic platforms utilized in this context. Chapter 3 introduces an indoor traversability estimation technique based on Generative Adversarial Networks (GANs), which laid the groundwork upon which our research is built. In this chapter, we underline the limitations of this architecture, emphasizing its shortcomings in training/inference time, and serves as a means of comparison with further experiments. In Chapter 4, we introduce the methodology that will be employed in subsequent sections, namely fine-tuning a transformer based architecture. We conduct fine-tuning experiments aimed at gaining insights into how the relative distance between our robotic platform and objects in the surroundings, renders the objects as obstacles and eventually impacts our vision-based perception. Chapter 5 describes, how the method introduced in Chapter 4 can be extended to multi-label classification. Chapter 6 portrays the integration of an additional modality, specifically the Laser Range Finder (LRF), as a method to enhance annotation and prediction. In Chapter 7, our attention shifts towards the identification of free space through a depth-guided segmentation approach. Finally, Chapter 8 presents conclusions based on our findings along with promising avenues for future research, highlighting areas where further investigation is needed.

CHAPTER 2

Comparing Non-trainable and Trainable Methods for Traversability Estimation in Indoor and Outdoor Environments

2.1 Non-trainable

2.1.1 Grid-based Representation

To construct a suitable representation of the environment that encapsulates essential knowledge about traversed areas, sensory information is processed using various techniques, including occupancy grids, digital elevation maps, and traversability maps. Specifically, the elevation map is structured as a two-dimensional regular grid, with each cell storing both a height value and variance information. As the robot navigates the environment and gathers new data, this map undergoes continuous updates. Traditionally, terrain traversability computation relies on assessing the traversability of individual grid cells within the elevation map. Elevation maps can be generated using on-board sensors such as Lidars and IMUs, leveraging the geometric properties of the adjacency grid, as detailed in [8]. This approach allows for the determination of traversability within the map, laying the groundwork for effective motion planning and navigation strategies.

Similar to the elevation map, the traversability map also adopts a regular grid representation. Employing a 2D grid-based approach that leverages fused stereo and visual data, this approach subdivides the environment into uniformly sized spatial cells [9]. However, the key distinction from the occupancy grid-map lies in the fact that each cell within the traversability map signifies the level of traversability rather than the occupancy status of the observed space.

A local traversability value is assigned to every cell within the elevation map, and through the computation of the traversability map, the traversability of a specific robot pose can be interpreted. The inherent directness and intuitiveness of traversability maps are rooted in their ability to be constructed based on sensor-acquired data. For instance, Wellhausen et al. [10] develop the traversability map by considering three fundamental terrain characteristics: slope, terrain roughness, and step height. In contrast, Fan et al. [11] represent the map as an aggregation of terrain properties (e.g., height, risk) over a uniform grid. The estimation of traversability must account for several potential challenges, including collision avoidance, step size, tip-over risk, contact loss, slippage, and sensor uncertainty, in addition to addressing localization errors. This holistic approach forms the basis for creating a planning framework, treating the problem as a Model Predictive Control (MPC) problem.

Properly identifying and calculating the values of these aforementioned topographical characteristics, together with the robot's mechanical capacity [12], an enriched traversability elevation map can be constructed. Consequently, it facilitates dynamic exploration tasks since it is going to dictate the exact location on which e.g. a walking robot can successfully land a valid step by adjusting its position and orientation accordingly [13, 14]. Alternatively, the process of obtaining a traversability map can serve as an intermediary step in calculating the robot's control commands, including steering and velocity adjustments.

For instance, Xie et al. [15] undertake the construction of a terrain map based on laser sensor data. This terrain map is subsequently transformed into a traversability map by assigning a Traversability Index (TI) value to each cell within the terrain map. Ultimately, employing a one-dimensional histogram known as the Traversability Field Histogram, the robot can navigate efficiently towards its intended destination.

In another approach, Martin et al. [16] emphasize, through their experimental work, how the utilization of onboard sensors such as GPS, accelerometers, and gyroscopes can facilitate the generation of traversability costmaps. These costmaps encompass four essential traversability metrics: power consumption, longitudinal slip, lateral slip, and vehicle orientation. These metrics provide valuable input for planning and executing efficient robotic navigation strategies.

2.1.2 Conventional Computer Vision

In the early days of Computer Vision (CV), traversability estimation heavily relied on making predictions based on the output generated by obstacle detection algorithms. A notable system, as presented by Thrun [17], leveraged monocular color vision data. This system focused on the analysis of individual pixels and their local visual attributes, such as intensity, color, edges, and texture. The crux of this approach centered around detecting pixels that exhibited dissimilarities in appearance compared to the ground and classifying them as obstacles. In essence, any discrepancy between a single pixel and the ground's appearance was identified and categorized as an obstacle.

Similarly, in the work of Huertas et al. [18], the emphasis was placed on discerning the boundaries of tree trunks against the background, a task determined by an edge detection algorithm. This early pioneering work in Computer Vision laid the foundation for more advanced traversability estimation techniques that have since evolved and integrated a broader spectrum of sensing and analysis capabilities.

Stereo imagery, along with color,hue,texture information can be practical in efforts of building a multi-algorithm approach [19] that is independently detecting numerous obstacles of governing characteristics such as tree trunks, water, excessive slope etc. by taking into account the terrain's attributes. Using stereo modelling and

outliers detection, Bajracharya et al. [20] build a uniform terrain mapping system that incorporates information on elevation,slope,roughness along with categorization of positive and negative obstacles.In specific, they focus on distinguishing thin structures with respect to the large amount of depth singularities and rich textural information involved, such as grass or sparse bushes, that cannot pose a genuine threat to the robot's safety.

Based on terrain features, such as slope and roughness, Castejon et al. [21] estimate the traversability characteristics, by exploiting the power of Voronoi Diagrams to model the XY dimensions of the outdoor environment as well as creating a qualitative representation that defines the traversability model. The latter provides useful geometrical information for constructing the Digital Elevation Map that eventually contributes in discretizing the workspace and isolate the cells that offer traversability information. As a means to execute a traversability analysis in complex catacomb-like environments, Bogoslavskyi et al. [22] perform experiments on a mobile robot solely collecting input from depth images drawn by a Kinect-style sensor. They use a sequential way of extracting the traversability interpretation starting from local traversability as a result of a single depth image and afterwards, proceed with integrating all the single-image traversability estimates, into a local traversability map. The way to ensure the efficacy of their method is to perform a pixel by pixel comparison between the traversability estimates and the custom-made structures with known 3D geometry. During certain evaluation trials, it is found that the traversable regions were dependent on a specific steering of the robot and thus their method is facing serious limitations.

2.2 Conventional Machine Learning

Early efforts at integrating sensory input with machine learning techniques primarily focused on obstacle detection and decision-making based on environmental cues.

For instance, Dima et al. [23] introduced a framework that harnessed the power of multiple sensors, including lasers, cameras, and infrared imagery. This sensor fusion approach aimed to detect humans, negative obstacles, and assess terrain traversability. It achieved this by combining the strengths of three different classifiers (AdaBoost, stacked generalization, experts) using manually collected data. This method exemplified how diverse sensory data could be leveraged for comprehensive environmental perception.

Pomerleau [24] explored the enhancement of autonomous driving through human demonstrations. By training an Artificial Neural Network (ANN) with camera input and incorporating domain-specific knowledge, the network demonstrated improved accuracy in steering an autonomous vehicle. This approach showcased the potential of machine learning techniques in autonomous navigation.

In environments characterized by significant uncertainty, regression methods played a pivotal role in traversability estimation. Ho et al. [25] utilized Gaussian Process (GP) Regression to predict a planetary rover's attitude and configuration angles by learning from experience on unstructured terrain. This method relied on formulating the GP regression problem effectively, using exteroceptive data as training input. It aimed to directly calculate traversability by estimating the kernel function's architecture to monitor vehicle states and uncertainty propagation. GP regression offered a continuous representation of the terrain, facilitating accurate traversability estimation even in areas with limited exteroceptive data. The authors suggested that

combining exteroceptive and proprioceptive learning could yield more comprehensive and precise traversability maps.

Extending the Gaussian Process regression model, Oliveira et al. [26] introduced an Uncertain-Inputs GP model. This model allowed for the consideration of localization and execution noise while modeling terrain roughness using vibration data as an input. By doing so, it provided a means to scrutinize the impact of noise on traversability estimation, showcasing the adaptability of GP regression in handling various sources of uncertainty in terrain analysis.

The Support Vector Machine (SVM), a kernel-based method known for its applications in classification, regression, and novelty detection, has gained widespread adoption due to its capability to make decisive classification decisions for new input vectors rather than providing probabilistic outputs [27]. One of its primary strengths lies in finding solutions that maximize the margin between distinct classes. In the context of road traversability detection, Bellone et al. [28] employ a carefully selected feature set generated through normal vector analysis. Their hypothesis suggests that using a normalized descriptor enriched with both geometric and color data enhances the generalization of the spatial descriptor. Their proposed descriptor, in conjunction with SVM employing four different kernels, demonstrates superior efficiency compared to certain standard descriptors, enabling the detection of road traversability from point clouds acquired in outdoor environments.

In a similar fashion, Zhou et al. [29] employ the AdaBoost algorithm in conjunction with Fuzzy SVM for feature selection on 3D point cloud data representing the ground surface. Their objective is to create a self-supervised visual learning system for terrain surface detection, particularly in forest environments. To train the classifier, they utilize a triangulated irregular network (TIN) to model the ground plane and extract training points from the 3D point cloud dataset. This approach

showcases how advanced machine learning techniques can be leveraged to address the specific challenges posed by complex natural environments.

An alternative approach involves harnessing combined color and depth descriptors to generate a textural descriptor, as described by [30]. These textural features, along with color information, serve as both training and testing inputs for an SVM classifier, facilitating terrain classification in areas covered with sand, grass, pavement, gravel, and litterfall. This method demonstrates the power of utilizing multi-modal data to improve terrain classification accuracy.

In another research endeavor, Kingry et al. [31] emphasize the significance of textural features in terrain classification. They extract key features from visual-spectrum images and employ an Artificial Neural Network (ANN) to identify terrain types such as grass, concrete, asphalt, mulch, gravel, and dirt. This approach showcases the effectiveness of machine learning techniques in discerning terrain characteristics from visual data.

Kim et al. [32] present a novel approach to unsupervised learning for terrain traversability prediction. They use stereo vision to autonomously collect labeled visual features corresponding to traversable or non-traversable examples. These labeled features are then fed into an online classifier learning algorithm. The operation of this algorithm consists of identifying and collecting appearance feature vectors from training examples and classifying newly collected image patches based on the learned models. The efficiency of this mechanism lies in utilizing labeled data collected within a specific time window for training, allowing the algorithm to effectively map input image patches to terrain map cells. Consequently, the output highlights areas classified as traversable, non-traversable, or unknown. Through experiments, it was demonstrated that this online learning approach enabled the robot to reach a goal location even when surrounded by densely packed tall grass, whereas a conventional planner

relying solely on estimating cost maps from elevation maps computed through stereo ranging proved unsuccessful due to limitations in feature space representation.

Happold et al. [33] undertake an ambitious effort that involves the fusion of geometry and traversability data to generate a comprehensive terrain assessment. Their approach utilizes a Multi-Layer Perceptron (MLP) with a single hidden layer, trained in a supervised manner using stereo images. The training data comprises eight-dimensional geometric vectors that capture various features, including slope, density, height, vertical distance, and more.

To collect the training data, the LAGR (Learning Applied to Ground Vehicles) platform acquires stereo image pairs, and a human expert labels each explored cell based on the perceived difficulty of traversing that specific cell, categorizing them as low, intermediate, high, or lethal. With a total of 4000 labeled cells, the analysis reveals that height and slope are the most influential features for determining terrain traversability.

To further enhance the terrain assessment, the classification derived from geometry features is integrated with color information, resulting in the creation of a cost map. This cost map serves as a valuable resource for path planning, enabling more informed decision-making during robot navigation in complex environments. This approach underscores the significance of incorporating both geometric and color-based data for robust terrain assessment and path planning.

2.2.1 Probabilistic

Thrun [17] makes a statement of fundamental gist that '*As robots are moving away from factory floors into increasingly unstructured environments,the ability to cope with uncertainty is critical for building successful robots*'

Navigation in outdoor complex environments encapsulates the need to handle the uncertainty risen as an amalgam of different factors such as sensors' noise and error, robot's mechanical limitations and most importantly the environment's unpredictable nature which renders its modelling as a quite challenging task. A plethora of approaches that can represent uncertainty using probabilistic distributions and modelling has been introduced for deriving terrain traversability. In the work of Ollis et al. [34], the robot learns to calculate terrain costs through human demonstration, indicating the likelihood of obstacle presence. This learning process combines Bayesian estimates with geometric information gathered through stereo vision. The resulting terrain costs follow a specific distribution, allowing the determination of path traversability. It is inferred that cells with higher feature values are less traversable, providing valuable insights for navigation. Similarly, in another approach described in [35], training data is generated during a safe journey in which a human operator guides the robot. Human input is utilized without presuming correlations between features and traversability. The Positive Naive Bayes (NB) classifier is applied to estimate the frequencies of observed features, thereby determining the parameters of the probability distribution for traversability. This approach results in the formulation of a traversability map that serves as a powerful tool for representing safe paths for the robot to follow. To further enhance detection accuracy, Sock et al. [36] utilize the Bayes' formula, which combines input from independent sources to estimate a single entity. They fuse traversability maps obtained from a visual camera and a LIDAR using this formula. The resulting map is modeled as a Markov Random Field, and the cells are updated independently, allowing for a more robust and accurate assessment of traversability in various environments.

Another regularly implemented technique which aims to autonomously improve traversability estimation capabilities in unknown terrains is the use of a self-learning

framework where 3D information corresponding to a densely vegetated terrain is extracted from the point cloud and is afterwards fed, through the form of geometric features, to a geometry-based classifier [37]. The main rationale en route to estimate the ground's traversability implies that the geometry classifier supervises a second color-based classifier and hence an iterative process that the system is retrained while new labelled data enriches the representation of the ground's model which is constructed with the use of Gaussian Mixture Models (GMM). Leveraging the advantages that a self-learning framework offers reciprocally with the use of *superpixels* as visual primitives, Kim et al. [38] employ vision sensing to estimate the traversability of the terrain based on its appearance instead of using the geometric stereo vision information. Their superpixel-based approach which produces higher levels of accuracy on image region classification, involving features containing color and texture information in RGB, computes traversability using Bayes' rule along with a modified k-nearest neighborhood (k-NN) algorithm. As a way to distinguish between known and unknown regions, i.e. frontiers of the traversability map obtained by laser scans during autonomous exploration, Tang et al. [39] make use of the reachability map that reduces the traversability map's dimension. By enforcing the k-means clustering method on the frontier candidates, along with the use of the A* Algorithm for finding the optimal paths, the cells on the grip map are being labelled as reachable,dangerous or unknown. Furthermore, defining the boundaries, upper and lower, of the terrain map Fankhauser et al. [40] propose a mapping approach using proprioceptive sensing (kinematic and inertial measurements) relying on the current pose of the robot that is being constantly updated as well as the noise and uncertainty of the sensor and roll,pitch angles respectively. The gist of their method is built upon creating a robust robot-centric elevation map that generates its data through the uncertainty derived from the robot's incremental motion in the form of mathematical equations.

Although their experiments using legged robotic hardware managed to apprehend and make use of the environment's uncertainty, their implementation seems to be facing limitations that the authors manage to address for maps of larger size along with localization singularities due to their platform-specific method. Similarly, another platform-specific effort that the proposed robot-centric mapping system aiming to derive traversability using laser-based 3D SLAM is illustrated by Droeschel et al. [41]. Using proprioceptive sensing (IMU and local odometry) along with rotating laser scanner measurements that in the surface element representation are going to be interpreted as Gaussian Mixture Model observations, a pose graph is assembled by the maps of all the adjacent key poses in the direction of successfully computing the robot's localization. By performing graph optimization, local dense 3D maps are constructed and integrated to a global one that yields information for the robot's real time pose and can ultimately provide traversability costs for rough terrain navigation for each map cell.

The work of [42] adopts a holistic approach that takes into account both the topographical properties of the terrain and the kinematic and dynamic configuration of the robot, which directly affect its motion. Their terrain traversability assessment method is prediction-based and relies on the Rapidly-exploring Random Tree (RRT) algorithm. The algorithm begins by creating a reference map for prediction. This reference map is generated through prior experiments conducted on rough terrains. Once the reference map is established, the algorithm determines the path for the walking robot to follow based on this reference information. Traversability is assessed by evaluating various factors, including footholds, feet trajectories, and other constraints among the cells of the map. This comprehensive approach ensures that both terrain characteristics and the robot's capabilities are considered when determining

the traversability of a given path, ultimately leading to more informed navigation decisions.

2.3 Trainable Methods

2.4 Deep Learning

While conventional machine learning techniques can face serious restrictions in terms of being able to process the collected data in their initial state, Deep Learning methods offer the prospect of creating better representations and thus leading to better understanding without onerous engineering struggles. A Deep Learning asset that can go beyond conventional machine learning methods in traversability estimation scenarios is that it offers the prospect of creating implicit relationships among data. Traversability estimation has been examined from the various perspectives of unsupervised,semi-supervised,supervised and self-supervised learning. Contemporary methods, are often associated with models being trained in an end-to-end supervised fashion, as a means to simplifying the training process. Equivalently, unsupervised and semi-supervised methods that use pretrained models' features shortly before training on a supervised dataset for a specific downstream task have been a sharing a large amount of popularity too. A common approach in deep vision traversability estimation techniques pinpointed by the latest research efforts, is to process an input RGB image through a series of pre-processing techniques and convolutional layers of a self-supervised network, before the meaningful features are fed to a traversability prediction network, i.e., a classifier (Figure 5.1).

2.4.1 Supervised

Supervised learning has emerged as a powerful tool for traversability computation in rough terrains, primarily due to its effectiveness in predicting and classi-

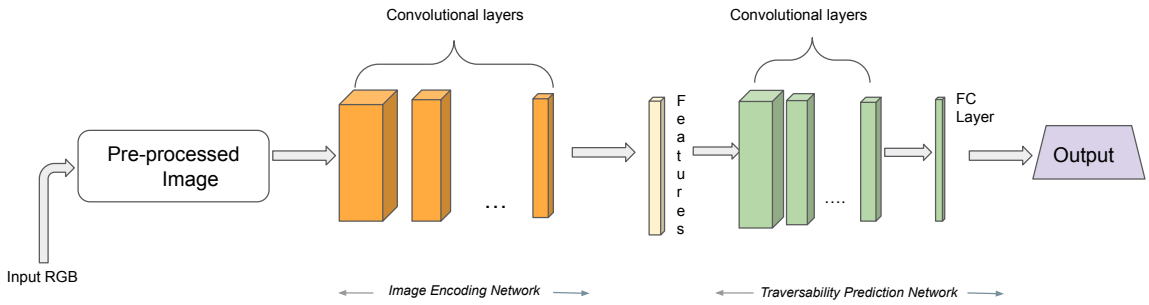


Figure 2.1: A typical pipeline describing the process of identifying the meaningful features through a series of convolutions before determining the traversability output through a subsequent convolutional network and a Fully Connected (or more) layer

fying terrains and regions based on specific class labels. The core concept in supervised learning involves mapping input sensory data to a target value, often indicating whether the terrain is traversable or not, after a training set has been acquired. In the study presented in [43], the focus is on investigating whether a traversability classifier trained on synthetic heightmaps can perform well on real heightmaps. In essence, the traversability estimation problem is framed as a heightmap classification problem, where the determination of whether a specific patch is traversable relies on comparing the outcomes of a standard computer-vision feature extraction technique with those of a Convolutional Neural Network (CNN). The deep learning architecture developed for this task consists of adjacent convolutional and max-pooling layers, followed by a Fully Convolutional (FC) layer with two output neurons. The results indicate that the CNN estimator outperforms feature-based approaches on both synthetic and real-world heightmaps. Furthermore, it is noteworthy that the training conducted in a simulation environment successfully transfers to traversability maps that reflect previously unseen real-world terrains. This finding highlights the potential for leveraging synthetic data for training models that can perform well in real-world scenarios, a promising approach in traversability estimation.

In the interest of performing long-range terrain segmentation using RGB stereo images in outdoor environments, Zhang et al. [44] design an end-to-end training deep CNN architecture aiming to augment the network's generalization efficiency. An encoder-decoder scheme using input feature and reference maps (calculated by the disparity images) is used. In particular, the encoder includes five layer ensembles consisting of convolutional layers followed by batch normalization ones right before the ReLU activation function and the pooling layers. The decoder uses upsampling layers that perform a deconvolution operation on the input features maps before the latter being processed by the aforementioned ensemble. It is reported that the introduction of the reference maps at the 1st, 3rd, and 5th ensemble layers provided a good balance between the segmentation guidance and the noise suppression. Subsequently, the output of the decoder is fed into a multi-class softmax classifier to generate predicted labels while performing pixel-wise classification. For their experiments, the used six hand-labeled image datasets containing RGB images and the disparity maps drawn from areas of dirt, foliage, natural obstacles (trees and dense shrubs), mulch etc. Subsequently, each pixel is classified into one of three terrain classes of traversable, non-traversable, and unknown regions respectively.

An assessment of terrain traversability for performing autonomous classification of Martian terrains is explored in [45] where a framework named Soil Property and Object Classifier (SPOC) that provides pixel-by-pixel image classification into one of 17 terrain classes is constructed upon a CNN. In a supervised fashion, human experts append the imagery dataset with new classes while the Mars Rover explores different sorts of terrains. A fully differentiable and trained for 6 hours end-to-end Fully Convolutional network , including multiple stages of filtering, various CNN layer dimensions (64,128,256,512) and downsampling, is used as the understructure before an upsampled penultimate layer classifies the input raw image (orbital or ground).

The output of the classification acts as the input to the cost function of the optimal route planner for landing site traversability analysis and also for building a robust slip prediction model.

The work by Tai et al. [46] explores the development of a human-inspired system trained in a supervised manner, which involves fusing Convolutional Neural Networks (CNNs) with a decision-making process. This system is designed to control a robot's exploration of an indoor environment, where the network's output generates control commands for the robot. In terms of the deep learning model employed, the input consists of a depth map. This input is processed through a series of three repeated layers that include convolutional operations, activation functions, and pooling layers. Following this, there is a fully connected layer with five nodes, each corresponding to one of the robot's possible states, including moving forward and various turning actions. The training dataset is comprised of indoor depth information collected using a Turtlebot equipped with a Kinect sensor. Human operators define the ground-truth data, and the robot's decision-making actions are designed to mimic human-inspired intelligence. This approach showcases the potential of combining deep learning with decision-making processes to enable a robot to explore and navigate indoor environments in a manner reminiscent of human decision-making.

Pfeiffer et al. [47] use a global motion planner, through which the robot learns a navigation policy in a supervised manner. They feed a CNN with fused pre-processed collected laser data with the relative goal position. Their CNN architecture encompasses two residual building blocks including shortcut connections that can address training complexity. Moreover, they train and test two CNN models in simulation (with a small differentiation in their dimensions), before expanding to a real platform that traverses an area with obstacles such as tables, chairs etc. Their results indicate that their model was not only able to learn the desired navigation strategies but also

to transfer the knowledge among different unseen environments. However, some impediments occur with the rise of the environment's complexity since it is stated that the CNN is not able to act as global path planner.

In contrast with methods that exhibit pure adherence to frames' binary classification as traversable or not, Palazzo et al. [48] design a supervised model that can analyze multiple traversability routes through the medium of the encoder-decoder architecture. Notably, while the problem is examined as a regression one, their aim is to estimate and predict the traversability costs of various routes even on scenarios that no labels are provided. Using collected RGB images as inputs, the utilized architecture consists of a fully convolutional network module for feature extraction, followed by two layers; a convolutional and a fully-connected respectively. The bottom line of their method lies on training a model to predict correct traversability scores on the source dataset, while carrying out unsupervised domain adaptation on the target data.

2.4.2 Self-Supervised

Self-supervised learning (SSL) is a form of supervised learning that human intervention, in terms of labeling, is not necessitated. In specific, the agent investigates a partition of unlabeled data, interprets it, and then, by developing a reliable representation, it is able to produce the labels missing and thus develop a sturdy perspective about the remaining part while automatically creating a labeled dataset. A key aspect of SSL that renders it as the contemporary most promising direction towards traversability estimation in unknown environments, is the ability to establish larger proportions of data efficiency in deep learning models that aim, as a consequence of reduced demand, for hand-labeled training data. Subsequently, it battles against the pure reliance on extensive amounts of data, and it is proven to be highly beneficial

especially for scenarios that involve updated data collections for different tasks as described by [49].

One of the first endeavors in determining long-range traversability using short-range data and self-supervision is described in [42]. In pursuance of training a vision classifier for a four-wheeled rover in a Martian-like rough terrain, the authors present two self-supervised approaches for local and distant terrain classification respectively. Short range data input, acquired both from vision and vibration sensors, creates a "local training" framework fusing texture,color and geometry information for all the encountered classes i.e rock,sand, grass. Using the short range training, the second approach for "remote training" employs stereo processing to identify the distance to patches in the image and, by position estimation, to identify when the rover has driven over a particular patch of the terrain. Consequently, long-range data was classified with respect to the classes having previously been identified for the "local" scenario. Collecting data while the rover traverses the terrain and setting a threshold for the data points collected for the visual classifier, training was realized with an SVM classifier, and by fusing the class likelihoods of the color, visual texture, and geometric sensing modes using Naive Bayes, terrain classification is performed.

Another work, interpolating a long-range vision classifier trained in self-supervision is portrayed by Hadsell et al. [50]. The classifier's output allow successful detection of trees, obstacles etc., by having the horizon as its perspective, and thus determining the traversability of the input large image patch patches. With regards to the features extractors involved, a total of four approaches working in an interleaved fashion is presented, each one being trained with either labeled or unlabeled offline data. The efficiency reported in their results is produced by the use of a multi-layer CNN that was initialized with deep belief net training, consisting of two convolutional and max-

pooling between them layers, which was responsible for independently pretraining each layer in both unsupervised and supervised manners.

Recent advances in Deep Learning have spurred significant research attention towards addressing the novelty detection problem in indoor and outdoor robot navigation. Notably, Richter and Roy [51] have presented a distinctive approach that tackles the issue of novelty detection using autoencoders, regardless of the extent of training the robot has received. In their methodology, the robot iteratively collects training data, self-labels it, and feeds it into a conventional feedforward neural network. This network comprises three hidden layers with sigmoid activation functions, followed by a softmax output layer that predicts collisions or lack thereof. Simultaneously, an autoencoder is employed, featuring three hidden sigmoid layers and a sigmoid output layer. The autoencoder's role is to reconstruct similar inputs and assess whether the new images closely resemble those in the training data. In the event of the autoencoder detecting something novel in the environment, the robot makes a decision towards a safe behavior. Otherwise, it continues to augment its familiarity with known environment types. However, it's important to note that there may be instances of misclassification of novelty in images, which could arise from inadequate training of the collision predictor. To mitigate this issue, the authors address it by matching the size and architecture of the hidden layers in both networks. This thoughtful approach aims to enhance the accuracy and reliability of the novelty detection system.

Wellhausen et al. [52] follow a similar approach in their study, where the robot collects RGB-D images in a self-supervised manner and is trained on a dataset consisting of 10,000 training image patches corresponding to traversable areas. The key component of their approach is the utilization of autoencoders for novelty detection. In terms of the architecture of the autoencoders used in their study, the encoder

consists of three consecutive blocks. Each block includes a convolutional layer with a kernel size of 5, followed by a Rectified Linear Unit (ReLU) activation function. The first two blocks are followed by a MaxPooling layer, while the final block is followed by an additional convolutional layer with a kernel size of 1. On the other hand, the decoder in their architecture mirrors the encoder's structure but employs nearest-neighbor upscaling layers instead of max-pooling layers. This design allows the autoencoder to effectively reconstruct input data and assess whether the new images align with those in the training dataset, enabling the robot to detect novelty in its environment and make appropriate decisions accordingly.

A deep neural network architecture consisting of multiple stream as a means to address the learning of features of different modalities, is presented by Valada et al. in [53] where using the convoluted mixture of deep experts (CMoDE) fusion technique, they provide a semantic segmentation technique that relies heavily on the use of ResNet [54] and dilated convolutions. They perform platform-specific experiments on a forested environment incorporating scenarios of detrimental nature for the robot such as low-lighting, motion blur,occlusions etc. Regardless of the absence of prior map, their tests showed that the robot could successfully traverse the trail by using on the fly semantic segmentation.

An alternative approach adopting self-supervised learning is presented by Shah et al. [55] where traversability is determined upon learning a traversability function 'T' which describes whether any controller can successfully navigate among collected observations (RGB images). This aforementioned navigation policy shall also take into account the environment's physics, rather than pure geometry, in order to decide which objects, tall grass for instance, are traversable or not. Subsequently, the ultimate goal of the policy is to successfully predict the estimated number of time steps required by a controller P to navigate from one observation to another. Both T

and P architectures encompass the use of a MobileNet encoder [56] followed by three dense layers that, using supervised learning, they project 1024-dimensional latents from the collected images to 50 class labels and 3 waypoint outputs corresponding to the relative pose between the collected observations for the cases of T and P respectively. These two learnt functions along with past gathered experience gained by arbitrarily chosen trajectories is unified to a system 'VinG' that governs a goal-oriented behaviour that solely relies on offline experience. En route to introducing a path planning framework that leverages learning of terrain's visual representations used in combination with unlabelled human navigation demonstrations, Sikand et al. [57] present a work that aims to create a mapping from the representation space to the terrain costs that the robot encounters while traversing a specific terrain. Visual data is collected during a self-supervised procedure in which the robot traverses a setting including sidewalks,trees,and terrains of grass,dirt etc. With a small human demonstration intervention, the datasets created, comprise of triplets of image patches corresponding to an anchor patch, a *similar* patch that portrays the same image as the anchor but from a different viewpoint and *dissimilar* one that includes information of patches that the human avoids choose and are far from the embedding space. As a means to form the visual representation, a CNN with two convolutional layers followed by three fully connected layers is used.

A conjointment of supervision between an unsupervised acoustic proprioceptive classifier that self-supervises an exteroceptive visual one is explored by Zurn et al. [58]. The robot equipped with both a stereo camera and a microphone, traverses various complex terrains and collects visual (terrain patches) and audio (vehicle-terrain interaction) data respectively. This is achieved by associating the visual features of a ground's patch right in front of the robot with the auditory features that correspond to the area that the robot is traversing. Projecting the camera images into a

birds-eye-view perspective, they act as weakly labeled training data for the semantic segmentation network trained in an unsupervised manner. By teleoperating a rubber-wheeled robot, they collect visual (24 thousand images) and auditory data (4 hours of video) of five different terrains: asphalt, grass, cobblestone, parking lot, and gravel. In regard to the architecture used, the authors implement an encoder/decoder architecture for the audio data and the MobileNet V2 model (pretrained on the ImageNet dataset) for the visual feature extraction network. Moreover, for the semantic segmentation of the terrains, they adopt the AdapNet++ network with an EfficientNet backbone. Their self-supervised exteroceptive semantic segmentation model achieved a comparable performance to supervised learning with manually labeled data.

An automated self-supervised learning method and the corresponding prediction of navigation-relevant terrain properties is presented in [59]. Specifically, they conduct their experiments using the ANYmal robot [60] by measuring the interaction, during locomotion, between the robot's sensorized feet and the terrain and then projecting the robot's footholds into camera images. Using this foothold projection system, they annotate semantic classes in the images by assigning a semantic label to each time step in the sequence while human annotation is only implicated in observing and marking the possible transitions between terrain types with a concrete time stamp and the terrain's type. The possible terrain types involved are asphalt, gravel path, grass, dirt, and sand. A learnt, while walking, terrain property, called *ground reaction score*, provides an alternative way to generate the footholds' labels in a self-supervised way. For the semantic segmentation purposes of this study, the authors use a CNN which architecture is based on ERFNet [61]. By tele-operating the robot through different environments, they collect a dataset of 70000 training and 15000 validation images respectively. Ultimately, in addition to the application

of their approach to a legged robot like ANYmal, it is mentioned that it could be deployed to other types of ground robots as well.

2.4.3 Unsupervised and Semi-Supervised

Semi-supervised and unsupervised learning provide auspicious ground for focusing on the essential segments rather than precise pixel-wise classification that require labelling of the training images.

One preceding method is described by Shneier et al. [62] using range and color information, build unsupervised models solely derived from the geometry and appearance of the scene. These models are learnt using clusters of neighboring data, extracted from the same physical region and can enclose an estimate of the traversability cost. However, due to the fact that the features learnt within the models are self-sufficient, not requiring any range data, causes the association between the region models and the traversability estimation to be uncertain for distant regions.

In recent research, a fundamental aim of the use of unsupervised learning in traversability estimation problems is to to learn a particular set of features that can be transferred to the network that will subsequently be trained on a specific downstream task (such as image classification) which determines traversable areas. Transfer learning approaches normally require a large-volume training dataset (e.g., Pascal VOC, KiTTi, ImageNet) to train, and by using the pre-trained weights of a model, a classification task can yield higher levels of accuracy and abstraction.

An instance of a deep-learning-based architecture, considering the unsupervised problem as supervised, that aims to train a generative model is served by using *Generative Adversarial Networks (GANs)* [17]. GANs have enjoyed wide use in computer vision research over the span of the previous decade and can automatically train a generative model while using both a generative and a discriminative model.

Hirose et al. [63] employ transfer learning to train a model for indoor traversability estimation using positive examples generated by an on-board fisheye camera attached to a Turtlebot, along with an attached laptop, over a time window of 7.2 hours. Their model architecture includes two standard CNNs, which form the adversarial modules—an essential part of the Generative Adversarial Network (GAN) framework. These modules consist of a generator and a discriminator, both trained in a supervised fashion. Notably, the authors find that even a small amount of annotation can lead to a significant improvement in performance. The generator takes a latent vector z as input, generated by an additional network called the inverse generator, which is trained concurrently with the generator and discriminator. To enhance the accuracy of their unsupervised method, they introduce an additional Fully Connected (FC) layer, which is trained in a supervised manner to classify scenes as "GO" or "NO-GO." This linear classifier uses GAN knowledge extracted from three specific GAN features. To assess the network's effectiveness, a saliency map is employed to highlight meaningful segments of the images. These segments typically correspond to the right and left sides of the input images and play a crucial role in determining indoor traversability, as they indicate the presence of walls or corridors. Ultimately, the authors suggest that their approach could be a valuable tool in the development of cost maps, which are crucial for robot navigation and motion planning in indoor environments.

Extending their work in performing GAN predictions for indoor traversability scenarios, Hirose et al. [64] introduce the GONet framework that uses the same aforementioned idea of semi-supervised learning incorporating a small amount of negative training data that can be proven to be more advantageous than solely including positive data, in improving traversability estimation. Indicatively, the GONet architecture consists of two models, one responsible for extracting features from positive

automatically labeled examples of traversable areas extracted by a fisheye camera mounted on a robot and the second which performs the final classification after been trained on both positive and negative examples. Additionally, in order to exploit the temporal nature of the collected data, an LSTM unit that captures the temporal dependencies in the data is added, creating a new separate model named GONet+T, and the output it yields is further fed to aa fully connected layer responsible for subsequently predicting the traversability probability. Strengthening the performance of GONet, a second extension named is introduced. GONet+TS is trained identically as the GONet+T and addresses the limitations in prediction owing to environment's structural complexity captured in stereo images. Performing indoor experiments with the TurtleBot2 platform and using the saliency map, despite the effectiveness presented in all methods performed, GONet+T and GONet+TS highlighted smoother predictions in indoors traversability estimation than GONet due to the inclusion of the LSTM layer.

Utilizing GONet's application with VUNet, a dynamic-scene view synthesis method [65], the authors present a unified system that can single out the traversable areas in the robot's vicinity. VUNet is the result of combining two supplementary networks SNET and DNet that have the ability to model static and dynamic transformations based on robot's actions. SNET is responsible for predicting static (S) and DNET for predicting dynamic (D) changes in the parts of the environment due to robot motion respectively. SNEt's architecture is based on the encoder-decoder scheme, and ought to the fact that the sampling procedure reuses original pixels of the input image, sharper images are generated. On the other hand, DNet is built upon a conditional adversarial network architecture. As a consequence, due to the bifold character of that synthesized approach, both static (e.g., walls, windows, stairs) and dynamic (e.g., humans) components of the environment can be predicted from differ-

ent camera poses in future time steps. In order to estimate future traversability, two applications based on assisted teleoperation are introduced i.e. early obstacle detection (moving pedestrians) and multi-path future traversability estimation. As inputs to VUNEt, the last two acquired images and a virtual navigation command, i.e., a linear and angular velocity are used. During the first experiment, VUNet predicts the motion of the human in the image and informs the teleoperator using warning signals and emergency stop commands. With regards to the second application, the system is able to generate virtual velocities for five different paths around the robot, and hence by predicting the images using scene view method, it can compute the traversability for each of the paths.

Using GONet and VUNet as a solid baseline, an 8-convolutional layer architecture named PoliNet [24] is trained to learn the Model Predictive Control-policy (MPC) for performing safe visual navigation of a mobile robot with mere human supervision. Concretely, by combining VUNet-360 (a variant of VUNet that uses input from a 360 camera) with the aforementioned traversability estimation network GONet, PoliNet can produce the velocity commands necessary for the robot to successfully follow a visual path in a safe manner. The control policy tries minimize the difference between an image taken from the 360 camera at time t and the next sub-goal image in the trajectory. Hence, the control policy is responsible for finding the appropriate location in a way that the current image looks similar to the one of the sub-goal's. PoliNet is trained offline before getting transferred to the online setup. Data was collected both in simulation and in the real world. With regards to the real data, the robot was teleoperated and gathered a total of 10 a half hours of 360-camera RGB images. Although their experiments show their method to be generally robust, there were instances in which the robot was not able to circumvent

large obstacles, mainly due to the fact that traversability was only considered as a soft constraint in the optimization problem.

Training GANs occasionally suffers from an array of reasons such as catastrophic forgetting [66] as well as difficulties in convergence, mode collapse and instability, due to design-related issues such as network architecture, appropriate selection of objective function, etc. [67]. For the cases of gathering data in an unsupervised manner or with scarce labels, such as an autonomous visual data collection by a mobile robot, recent advances in self-supervised contrastive learning offer the advantage of optimizing the learning capabilities of the designed model or operating in conjunction with the semi-supervised learning approach that is tailored to the downstream task that is examined. For instance, Goh et al. [68] use the popular approach of SimCLR [69] to perform a martian terrain segmentation analysis with limited data corresponding to classes such as oil, bedrock, sand, big rock, rover and background. Using supervised contrastive learning, Gao et al. [70] manually label a set of anchor patches in their effort to efficiently create a feature representation that is able to distinguish different traversability regions. Shah & Levine [71] combine (a) the output of a heuristic model trained on teleoperated prior data using the contrastive InfoNCE loss function [72]; with (b) the output of a local traversability model towards successful path planning.

2.4.4 Deep Reinforcement Learning

As described by Sutton [73], *in uncharted territory-where one would expect learning to be most beneficial-an agent must be able to learn from his own experience.* Reinforcement learning (RL) enables a robot to autonomously discover an optimal behavior through trial-and error interactions [74]. What differentiates the field of robotics in terms of applying reinforcement learning to, is the amount of challenges encompassed. One major arduousness is the high dimensionality and complexity of

the states involved as well as the adversity in performing a complete and noise-free observation of the true state. What is more, another considerable difficulty that RL is facing in Robotics, derives from the fact that interactions between a mechanical system and its environment can harm the platform or any humans involved. However, RL can be proven to be an effective arrow in the quiver when the robot is navigating through complex and dynamic environments [75]. On top of that, conventional RL algorithms fused with Deep Learning, can handle many practical problems, where the incorporated states of the Markov Decision Process exhibit high levels of dimensionality and thus optimal policies are easier to be learnt.

An end-to-end deep reinforcement learning approach for a mobile robot navigating an unknown environment is portrayed by Tai & Liu [76] in which the inputs are raw depth images and the control commands serve as the outputs. As a means to create a feature representation, they use a CNN architecture with three convolutional layers which weights are initialized by a pre-trained model and three fully-connected layers for exploration policy learning. Since the robot is navigating in an indoor environment filled with obstacles, the feedback in terms of negative reward is obtained by the potential collision between the robot and obstacles. After training the model for many thousands of iterations, simulative and real-world indoor experiments demonstrated efficacy in obstacle avoidance along with improvement in the traversable areas detection. A dueling architecture (Figure **??**) named D3QN was initiated in the work of Xie et al. [77] in the pursuance of obstacle avoidance using the concepts of convolution and *deep Q Learning* as its main foundations. It consists of a fully convolutional neural network, that outputs depth information from an RGB image which has previously been blended with additional noise and blur to adapt to real-world scenarios, followed by a deep Q network [78] that encloses a convolutional and a dueling network [59] while the main hypothesis implies that the training, taking

place in simulation (Gazebo), will provide adequate knowledge to be transferred to real-world implementations. In terms of training speed, it was proven that the D3QN architecture is almost twice faster than DQN, highlighting its efficiency on obstacle avoidance scenarios and, consequently in efforts of determining traversable obstacle-free regions. Zhang et al. [79] present an architecture that accepts depth images along with the environment's obtained elevation map and the robot's 3D orientation as the inputs, which are fused and fed into an Advantage Actor-Critic (A3C) model [80]. Before their merging, depth and elevation map information is each passed through a four layered convolutional structure followed by a pooling layer whereas the 3D orientation is passed directly to a fully connected layer and then merged with the elevation information. All input sources are then concatenated and fed to an LSTM that can improve capturing the underlying states of a partially observed environment. Eventually, the actor and critic components consist of a fully connected layer each with the difference that in the actor part the output vector's values are normalized by the Softmax function. Although both training and testing phases took place in a simulative 3D environment with varying levels of terrain's traversability, their results showed that the agent sufficiently learnt to traverse different terrains towards a predefined goal location, and occasionally around non-traversable objects, with an average Deep RL decision-making time of 0.074 seconds.

An off-policy algorithm [81] acting as a powerful tool in improving the learning capabilities of an end-to-end RL approach, named BADGR (Berkeley Autonomous Driving Ground Robot), uses a self-supervised data labelling mechanism that is not built upon any human supervision or SLAM techniques in simulation. By collecting data using a random control policy, collisions are detected either by LIDAR or IMU as the robot stores the sensory observations along with the corresponding actions taken. Events are labeled in a self-supervised manner by the collected dataset and

then appended to it. Input RGB images act as the current observation that along with a future sequence of actions such as control commands, future events can be predicted. As far as the model's architecture is concerned, the input images are fed into three convolutional and 4 fully connected layers each one followed (except for the last one) by a ReLU activation function, in order to form the initial hidden state of a recurrent LSTM unit that will handle each future action and yield the corresponding predicted future event After deploying the BADGR system in real-world environments, it was shown that, by using only 42 hours of autonomously collected data, it could successfully traverse areas of tall vegetation and bumpy terrains.

By taking the history of proprioceptive states into account, Lee et al. [82] undertake the rough terrain traversability estimation as a temporal problem that requires a robust controller to produce the appropriate actuation. In this context, a sequential *Temporal Convolutional Network (TCN)*, comprised of convolutional layers each followed by a ReLU activation function, uses input from joint encoders and IMU and, accordingly, implicitly learns to analyze contact and slippage events while a four-legged robot is navigating in complex terrains including those of mud,sand,snow etc. Towards this goal, they claim that direct RL techniques might not be fruitful due to large time processing and thus a teacher-student policy is selected instead. First, the teacher policy based on ground-truth knowledge concerning the interaction between the robot and the terrain, is trained on simulation. Afterwards, it supervises a student learning and the eventual student policy acts on the real robot. An additional concept introduced during the training stage, that enhances the robustness of the method is encapsulated on the adaptive nature of synthesized terrains in order for the controller to traverse them. Finally, as a means to integrate the neural network to regulate the controller, the Policies Modulating Trajectory Generators (PMTG) is employed.

Other methods include the implementation of deep inverse reinforcement learning [83] for determining off-road traversability [84]. Towards this direction, the authors propose a two-CNN structure that encompasses the vehicle kinematics in the states (2D poses) which unavoidably leads to an increase of the state-space complexity. In their experiments, data is collected from a laser scanner which, is transformed to input features for a five-layered fully convolutional network structure that by recurrently applying a convolution layer for 120 and 150 times per network, the value iteration is completed with noticeable reduction of the computational burden.

2.4.4.1 Transformer-based Architectures

Pre-training transformers [85] [86], paves the pathway to create rich representations that can be thereupon used for fine-tuning on the desired downstream tasks.

ViT (Vision Transformer) [1] has been the cornerstone in modern Computer Vision applications. Relying on self-attention-based architectures, ViT-based methods have shown outstanding results in image classification tasks [1, 87–89] surpassing conventional methods such as Convolutional Neural Networks [90]. Furthermore, a plethora of contemporary research endeavors in the field of robotics and autonomous driving, reveals the indispensable role that ViT exhibits, in the extraction of semantic information

Although transformer-based models' efficiency is dependent on very large datasets, pre-trained ViT networks encode representations that are rich in semantic features and can be fine-tuned to the desired downstream image classification tasks [91, 92]. ViT's strength derives from multiple reasons as highlighted by Raghu et al. [90], but the following two are vital the methodology that this dissertation implements: 1) the large amount of global information in their lower layers; and 2) the preservation of input spatial information even at their final layers. Furthermore, transformer-based

frameworks have also played a prominent role in semantic segmentation-related research endeavors, due to the efficiency and robustness they exhibit in creating rich representations. Specifically, SegFormer [93] has gained widespread popularity due to its ability to leverage multi-scale features as outputs and to combine local and global attention.

Strudel et al. [94] extend the use of ViT, and introduce *Segmenter*, a transformer-based model for semantic segmentation. Generally speaking, the hierarchical structure is highly suitable for tasks involving pixel-level predictions, such as semantic segmentation and object detection. SegFormer [95] combines ViT-architecture with lightweight *Multilayer Perceptron (MLP)* decoders, and has shown significant efficiency for semantic segmentation tasks. Other transformer-based models highlighting the diverse applicability of such in computer vision tasks can be found in [96] with particular focus on image restoration, in [97] addressing image classification, object detection, and semantic segmentation, and in [98] specifically targeting the segmentation of medical imaging.

Owing to their proficiency in handling inputs as sequences, transformers can capture long-range correlations, enabling robust multi-modal frameworks. Multi-Head Self-Attention (MHSA) [99] constitutes the main component of a ViT, contributing to much of its success over ResNet [100], the uncontested state of the art in Computer Vision. MHSA can be a powerful tool in controlling the mixture of information among different parts of an input sequence and thus leading to richer representations. As described by Tsai et al. [101], the multi-head cross-modal attention module is responsible for updating each modality's sequence via low-level external information. Eventually, the cross-modal transformer learns to correlate meaningful elements across different modalities. Endeavors in robotics fusing modalities through the use of the MHSA module include navigation using natural instruc-

tions and graph [102], multi-robot collaboration for unknown environment exploration [103], and UAV-driven segmentation [104].

CHAPTER 3

GAN-based indoors traversability estimation

3.1 Overview

Training on simulation data has proven invaluable in applying machine learning in robotics. However, when looking at robot vision in particular, simulated images cannot be directly used no matter how realistic the image rendering is, as many physical parameters (temperature, humidity, wear-and-tear in time) vary and affect texture and lighting in ways that cannot be encoded in the simulation. In this chapter we examine a different approach for leveraging simulated environments: We build two environments on Unity, acquire simulated visual datasets, and used them to reproduce experiments originally carried out in a physical environment. The purpose of this work is to explore whether these simulated environments can act as test-beds for further deep learning experiments as well as experiment with a GAN-based architecture as a feature extractor for binary traversability estimation.

3.2 Methodology

Our experiment replicates the network architecture proposed by Hirose et al. [63, 64], namely a *Deep Convolutional Generative Adversarial Network (DCGAN)* ensemble of two adversarial modules, i.e., a *generator (Gen)* and a *discriminator (Dis)*, along with a third network, the *inverse generator* (Gen^{-1}). The inverse generator is trained on the real-image dataset and outputs the value of the latent vector z that is fed as an input to the Generator (Figure 5.1). All three networks (Gen, Dis, Gen^{-1}), are built as standard CNN which are trained simultaneously in an unsupervised man-

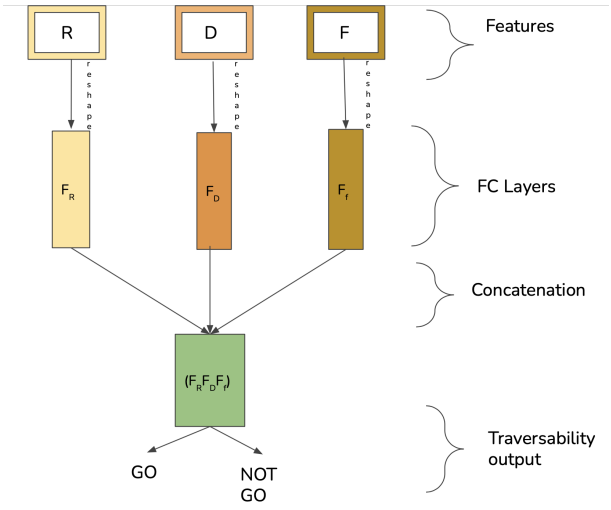


Figure 3.1: The *Residual Loss (R)*, *Discriminator Loss (D)*, and *Discriminator Feature (F)* tensors are reshaped and concatenated before being used as inputs for the fully convolutional classifier.

ner. Three tensors R, D, F are extracted from the discriminator and used as features for an additional classifier layer that decides if the scene encountered by the robot is GO or NOT GO. Specifically:

- R corresponds to the *Residual Loss* and is expressed as the pixel-wise absolute difference between real and generated images.
- D corresponds to the *Discriminator Loss* and is expressed as the absolute difference between the last convolutional layer features of the discriminator applied to the real data, and the same layer features applied to the generated images.
- F corresponds to the *Discriminator feature* expressed as the last convolutional layer features of the discriminator applied to the real data.

These tensors are reshaped into vectors and concatenated into the classifier input. The classifier is a Fully Convolutional (FC) layer trained in a supervised way on a small amount of both positive and negative annotated images. The FC layer is trained to output the final 'GO' or 'NO-GO' decision (Figure 3.1).

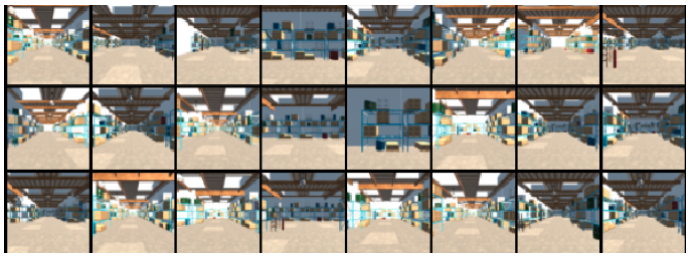


Figure 3.2: Training images (Warehouse)

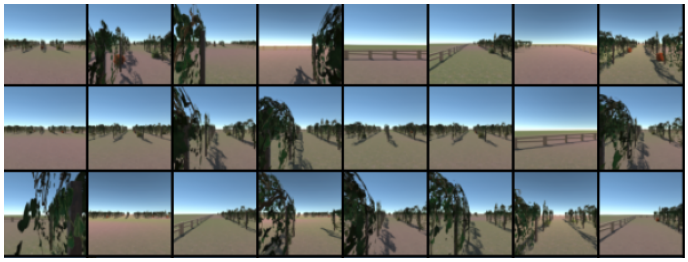


Figure 3.3: Training images (Vineyard)

3.2.1 Training

We have collected images through the robot autonomous navigation in Unity and randomly split the runs between training and testing. For both the indoor (Figure 3.2) and outdoor (Figure 3.3) scenario the resulting dataset, collected in a self-supervised manner, consists of approximately 17000 positive training images. On top of that, an additional 1% of manually labelled negative images, along with an equal number of positive ones arbitrarily selected from the large dataset, is used in order to train and test the FC layer. We conduct our experiments on a Intel(R) Core(TM) i7-5820K CPU @ 3.30GHz, using Nvidia Cuda 11.4 GPU. Using the Pytorch Framework and Python 3.8.10, we train for a total of 60 epochs using a learning rate of 0.00001, Adam optimizer with beta1 = 0.85.

Features	Accuracy [%]	Recall [%]	Precision [%]	F1 score
Vineyard environment				
R+D+F	90	97	85	91
R+D	75	76	74	75
D+F	84	86	85	84
R+F	96	95	95	95
R	51	53	76	36
D	85	85	84	86
F	95	92	91	93
Warehouse environment				
R+D+F	94	94	90	92
R+D	79	80	81	79
D+F	88	91	87	90
R+F	90	90	90	92
R	53	51	41	34
D	80	80	78	79
F	93	93	91	92
Physical experiment, as reported by hirose2017go				
R+D+F	94.25	95.75	92.96	94.33
R+D	91.63	94.00	89.74	91.81
D+F	93.00	96.50	90.19	93.24
R + F	93.13	95.00	91.56	93.25
R	85.38	83.50	86.75	85.10
D	91.63	94.50	89.36	91.86
F	92.25	95.50	89.67	92.49

Table 3.1: Results on the test dataset using networks of different variations for the simulated environments

3.3 Results

We evaluate the environment's suitability by assessing each network on the test datasets. Specifically, in order to measure the performance of each network, we use accuracy, recall, precision, and F1 score as metrics (Table 3.1).

The following points support our claim that the same conclusions are reached about the method from the simulated and the physical experiments:

- The F feature is the one that performs best by itself, and R is the feature that performs worst, although the differences are more pronounced in the simulated

experiments than in the physical experiments. So the simulated experiments rank the features correctly, but give an exaggerated impression of their performance relative to each other.

- Among the two-features models, those that include F perform better than the one that does not. This observation is conforming to the physical experiment, although, similarly to above, the differences are more distinct in the simulated experiments.
- In the RDF model, recall is higher than precision, pointing to the direction of making the model more specific in order to improve accuracy. This observation is accordant with the physical experiment.
- Regarding the individual features, both the simulated and the physical experiments show that D and F need to be more specific.

On the other hand, the following points run counter to our above-mentioned claim:

- In the vineyard scenario, dropping the D feature increases the levels of performance, although in the physical data all features provide some information and the RDF model is the most performant one.
- Among the two-features models, those that exclude R perform better than the one that does not. This observation is not consistent with the physical experiment, where the experiment that excludes D performs equally well as the one that excludes R.
- Regarding the precision and recall of the R feature, the physical experiments show that R needs to be generalized to increase recall, but the simulated experiments give mixed results.

Overall, we notice that our results are in relative accordance with the results obtained using the physical platform and the highest accuracy was achieved using the ensemble of RDF features for the warehouse environments which is more abundant in features

than the vineyard. Inevitably, some limitations are faced due to the conventionalized nature of the simulated environment. Enriching the environment with more elements of physics and stochasticity as well as integration with a physical platform constitute the main avenues of our future research.

3.4 Discussion & Challenges Encountered

We created two simulated indoor and outdoor Unity environments. and through GAN-based deep learning experimentation we aim to test our main hypothesis that the simulated environments provide a solid groundwork for robot vision problems. We have found that our approach is valid for the purpose of feature selection, as the relative ranking of the models using different mixtures of features is consistent between the simulated and the physical experiments. Inevitably, some limitations are faced due to the conventionalized nature of the simulated environment. What is more, we noticed that the architecture used (three networks trained concurrently) 1) requires a lot of computational power 2) is quite difficult to converge. Generally speaking, GANs suffer from a number of reasons such as catastrophic forgetting [66], as well as difficulties in convergence, mode collapse and instability due to design [67].

CHAPTER 4

RGB-based indoor traversability estimation

4.1 Overview

In this chapter, we delve into ViT, a transformer-based architecture that will constitute the cornerstone of our research endeavors. We want to leverage the notion of transfer learning from a large pretrained-backbone and highlight its dominance against CNN-based pretrained backbones (GAN-based, ResNet). In particular we are fine-tuning a ViT on our custom collected dataset while exploring the notion that the relative distance between the robotic platform and an object, is what determines whether this object is actually an obstacle or not.

4.2 Methodology

In the core of our fine-tuning approach relies on a Vision Transformer (ViT) encoder, which consists of a series of self-attention and feed-forward layers. ViT encoders have proven to be highly effective in achieving both strong generalization and handling large-scale datasets. In our research, we employ a pre-trained ViT model that has been trained on ImageNet-21k using the self-supervised Masked Autoencoders (MAE) technique [105].

The MAE technique involves the following steps:

1) Masking Random Patches: Random patches of an input image are masked, effectively hiding certain portions of the image.

2) Encoder (ViT) Processing: The encoder, based on the ViT architecture, is applied only to the visible parts of the image, which are the unmasked regions.

3) Decoder Operation: The decoder operates on both the masked tokens and the encoded patches. This process aims to reconstruct the missing pixels in the masked regions.

After the pre-training phase using MAE, the decoder component is discarded, and the pre-trained encoder is retained. This encoder is then fine-tuned on specific image classification tasks. This approach aligns with our fine-tuning methodology, where we aim to fine-tune the pre-trained ViT encoder on our dataset, focusing on the task of indoor traversability estimation.

In our case, the fine-tuned encoder is used to classify RGB scenes encountered by the robot in its surroundings, ultimately determining whether these scenes correspond to traversable areas. By leveraging the pre-trained ViT encoder's ability to capture rich features and representations, we aim to improve the accuracy of our indoor traversability estimation models.

It has been shown that MAE have the ability to learn visual scene semantics in a holistic manner and that accounts for choosing these pre-trained weights for our task. Besides, MAE have also shown substantial efficiency in transfer learning tasks such as object detection, instance segmentation, and semantic segmentation which makes it as a reliable pre-training approach for our traversability estimation task. We also experiment with the Vit-B/16 base model, pre-trained on ImageNet-21k [1], [85]. This represents a standard ViT approach that can be supported by the available computational resources and also serves as a comparison against ViT_{MAE}.

The pre-training phase plays a crucial role in enabling our model to learn a rich inner representation of images. This representation captures meaningful features that are valuable for our traversability estimation task. To leverage this representation effectively, we employ supervised fine-tuning, which involves attaching a projection head to the pre-trained model. The projection head consists of two fully-connected

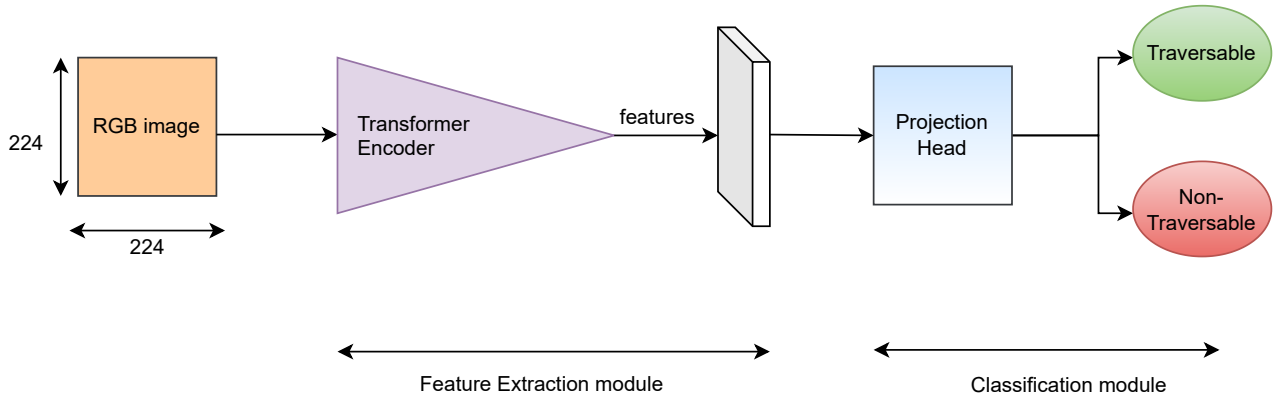


Figure 4.1: Overview of the method

layers. The pre-trained ViT model generates a feature vector with a size of 768x1 for each input image. This feature vector is then passed through the projection head, which performs classification to determine whether the encountered scene is traversable (GO) or non-traversable (NO-GO), as illustrated in Figure 4.1. We opt for this relatively simple network structure to mitigate the risk of overfitting, given the limited amount of annotated data available for our task. The inclusion of a fully-connected layer before the final classifier has been shown to enhance the model's accuracy compared to using a single linear layer alone. This design choice helps improve the model's ability to capture relevant features and make accurate predictions.

4.3 Experimental Setup

4.3.1 Dataset Collection and Annotation

Figure 4.2: Dataset illustration, Sets 1 to 5 (from left to right)

4.3.2 Dataset collection

In our experiments we used the Summit-XL Steel robotic platform4.3 to gather data. A human operator directly teleoperated the robot with a wireless PS4 controller. We used ROS Melodic[1] and the *message_filters* data synchronization package[2]. to operate the robot and to record the RGB, laser range finder, IMU, and encoder channels.

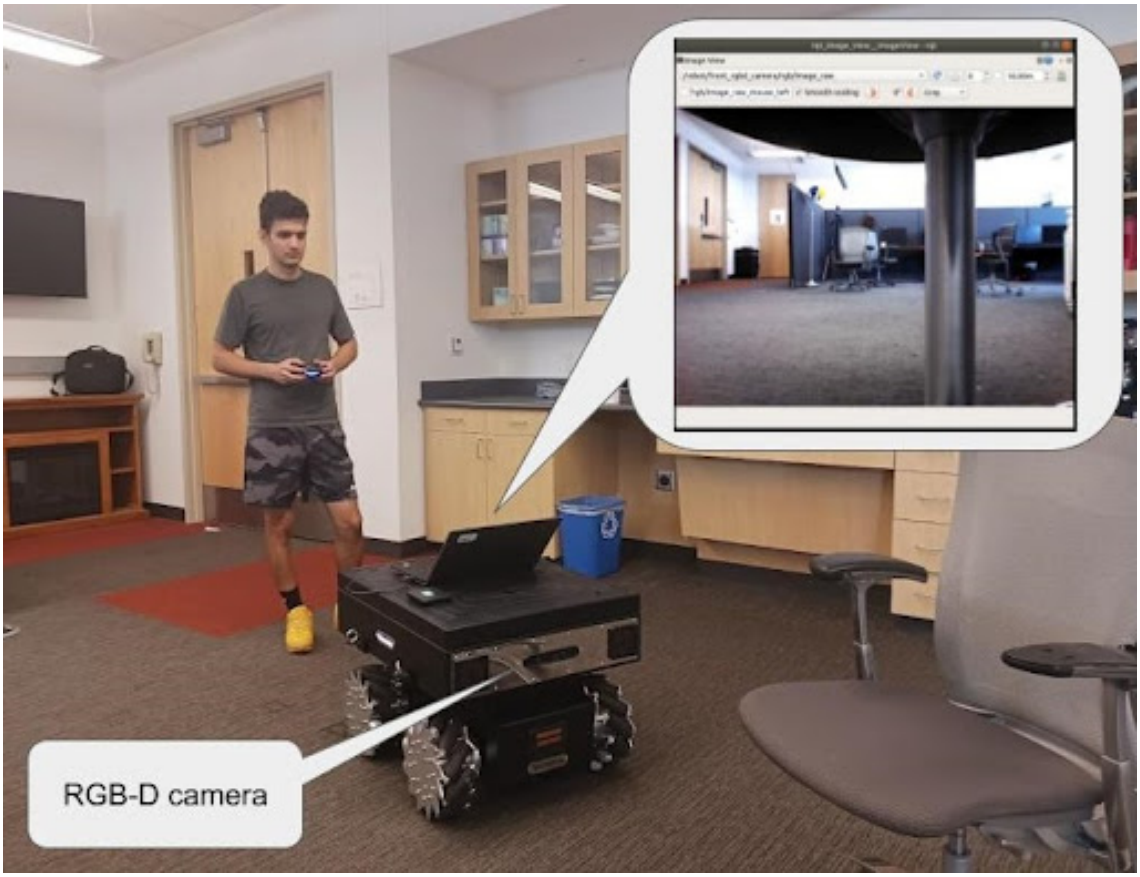


Figure 4.3: The Summit-XL steel platform used in our experiments

[1] `http://wiki.ros.org/melodic`

[2] `http://wiki.ros.org/messagefilters`

The robot was deployed in five different buildings around the University of Texas, Arlington (UTA) campus. We navigated the robot in hallways and areas of shifting illumination and various objects. In the occurrence of a possible interference with a human or an object, the operator stopped the robot before getting too close. The dataset collection process led to a total number of 24432 images. Each set presents the following attributes:

- Set 1: Features dark illumination, dense and voluminous objects, narrow paths for robot traversal, and unique object structures.
- Set 2: Exhibits bright illumination, includes scenes with walls, fewer static objects, and static/moving humans.
- Set 3: Shows moderate illumination, contains chairs and table booths, and includes static/moving humans.
- Set 4: Presents bright illumination, features long hallways, and encounters a smaller number of objects.
- Set 5: Displays bright illumination, represents a laboratory setting with workstations, includes static obstacles, and static humans.

Given the uniform academic nature of the data collection environment, it is expected that certain object classes such as chairs, desks, hallways, and doors would be present in all buildings. However, these objects exhibit variations in geometry and appearance due to differences in lighting conditions. Among the five sets, Sets 1 and 5 demonstrate the most prominent disparities compared to the other three sets, as depicted in Figure 4.2.

4.3.3 Dataset Annotation

Experiments were carried out by teleoperating the robotic platform around university buildings and hallways. We applied a similar labelling process as the one

described by Hirose et al. [64]. Specifically, we used extracted velocity from the odometry provided by the robot platform's encoders. We set a threshold value of 1 m/s (heuristic value) and we consider a time window of 2.5 seconds. If within this window, all robot's wheels' velocity v_t is constantly above the threshold value of 1 m/s and the robot is moving forwards, then we determine that the central frame during this window portrays a traversable scene. We also ignore images captured while the robot is turning. We determine that the frame encountered in time t is positive (traversable), if $v_t \geq 1$ or unlabeled, otherwise.

The rationale behind the selection of images for supervised fine-tuning was to create a dataset in an empirical fashion that includes a balanced number of positive and negative images. Additionally, it includes scenes where certain objects (chairs, trash bins, desks, corridors) are drawn from different environments, exhibit different geometric and color properties, and can appear as obstacles or not due to their location witnessed by the robot's front camera.

As for the positive images used in the experiments, we initially randomly chose a certain proportion from the positive labelled dataset, and we discarded hallway images that, although they present vast amounts of homogeneity, were collected during different runs of the same environment and thus they did not substantially contribute towards features' variety. Since we did not use any other modality (i.e. laser) for the automated labelling procedure, there were instances of noise in the positive images, such as humans showing up unexpectedly. We manually removed this instances from the positive set and complemented the dataset with other positive examples.

With regards to the negative images, we used the unlabeled dataset to manually select frames that include a plethora of static obstacles, dangerous areas, or humans that appear in the proximity of the robot. As a means to act towards an enhanced direction of safety, we also included ambivalent scenes that for instance, a human

subject is moving away from the robot, but due to the uncertainty that might govern humans' moves, we decided to label these frames as negative too.

4.3.4 Fine-tuning

We used the PyTorch[3] framework as the basis of our experiments. Training was done on a machine with 2 Titan RTX GPUs (24GB GDDR6 RAM, 4608 Cuda Cores). We used horizontal flip as a dataset augmentation technique. Using the cross-entropy loss function, we trained for 50 epochs unless an early stopping callback terminated the trial upon observed convergence. As training parameters we used: batch size = 16, learning rate = 0.01 and weight decay = 5e-4. For the fine-tuned transformer we freeze all deeper layers and replace the classifier head with the projection head that consists of two fully-connected layers; the last one performs the classification. We fine-tuned the layers using stochastic gradient descent (SGD).

4.4 Results

We perform an ablation study to evaluate the performance of our fine-tuning method on our custom dataset. We perform 5-fold cross validation on four buildings selected for training and the remaining one for testing. By folding on the buildings, we exploit the visual dissimilarity between semantically equivalent classes (chairs and desks) between buildings. This comparison enables us to evaluate the ability of the proposed method to generalize beyond learning visual representations of specific objects.

After experimenting with 600, 800, 1000, 1200 data instances we noticed that the best performance of the supervised classifier is achieved when using 1200 samples for each set i.e 600 for positive and 600 for negative. Using the same projection

[3] `https://pytorch.org`

head architecture, we also fine-tune a deep residual network (ResNet) [106], in particular the ResNet50 variant, that has been pre-trained on ImageNet-21k. We chose ResNet50 ought to the fact that it produced the best results compared to its other ResNet counterparts (ResNet10, 18, 34) for our small dataset. Similarly to above, we replaced the classifier with the projection head for the classification, and the best results were achieved using a learning rate of 0.001.

Furthermore, we train a GAN ensemble network following the methodology described in [64]. Due to our limited size custom dataset, we used the LORIS indoors dataset [107] that bears some resemblance to ours, as the one to train the feature extraction module on. Specifically, we used approximately 65k images collected from indoor scenes (home, office, market, cafe, corridor). Ultimately, we train from scratch a small convolutional network of four convolutional and two fully connected layers that each, except for the final, is followed by a ReLu activation function. Accuracy is chosen as the performance metrics of the experiments conducted and Table 7.1 summarizes the best results achieved by each network configuration.

To further evaluate our fine-tuning method we perform a direct one vs one test set comparison. This will enable us to monitor the strengths of generalization of the fine-tuning method. For both fine-tuned ViT and ResNet, we choose the datasets that, after performing 5-fold cross validation, presented the maximum (Set 5) and minimum (Set 1) accuracy. The results are shown in Tables 4.2 and 4.3, respectively.

4.5 Results and Discussion

Table 7.1 presents the results of the ablation study. Overall, we notice that the fine-tuned ViT$_{MAE}$ outruns all other networks while exhibiting important levels of consistency. This is in consensus with the results in literature [1], [90] that ViT can show remarkable performance while outperforming CNNs in image classification

tasks, and in this case reinforced by the MAE training that includes the notion of learning visual semantics holistically. As far as ViT-B/16 is concerned, there were scenarios in which ResNet performed slightly better or in a quite comparable fashion. Hence, we can infer that for our small dataset that contains open paths as well, ViT-B/16 does not demonstrate adequate enhancement in performance compared to ResNet50. Ensemble GAN performance is poor, due to the difficulty in training all the networks of the ensemble with an adequate number of data. Finally, our custom supervised CNN achieved relatively low accuracy for practical purposes.

Set 1 presents the smallest amounts of accuracy due to being the one with the most uniquely distinct features in terms of visual information. Namely, compared to the others sets, Set 1 is the one with the least bright illumination as well as areas with dense and more voluminous objects and hence this set is largely differentiated. On the other hand, the high levels of performance on Set 5 can be justified by the fact that this set is the most balanced in terms of varying features. It includes more balanced levels of information regarding ambience illumination and less surprising object location patterns.

Table 4.1: 5-fold cross-validation results

Accuracy[%]	Testing Set 1	Testing Set 2	Testing Set 3	Testing Set 4	Testing Set 5
ViT$_{MAE}$	79.05	90.16	83.41	89.75	92.00
ViT-B/16	77.68	82.74	79.33	85.35	85.78
ResNet50	78.10	83.66	76.91	87.83	87.41
Ensemble GAN [64]	69.41	65.08	63.26	69.58	66.16
Custom CNN	75.25	71.03	68.83	71.75	84.16

Tables 4.2, 4.3 report the individual performances on Sets 5 and 1 respectively. Aside from the fact that less data is used for training, we observe that ViT$_{MAE}$ is the most performant one. ViT$_{MAE}$ performance convincingly surpasses the performance

Table 4.2: Individual Results by Testing on Set 5

Accuracy[%]	Train on Set 1	Train on Set 2	Train on Set 3	Train on Set 4
ViT_{MAE}	85.50	87.25	87.58	84.66
ResNet50	65.90	75.33	85.25	80.00

Table 4.3: Individual Results by Testing on Set 1

Accuracy[%]	Train on Set 2	Train on Set 3	Train on Set 4	Train on Set 5
ViT_{MAE}	77.08	81.58	75.90	82.00
ResNet50	67.33	75.33	63.50	75.60

of ResNet50 and proves that adaptation to new unseen environments is achieved with satisfactory accuracy. The lowest accuracy achieved when testing on Set 1 (Table 4.3) is noted when training on either Set 2 or Set 4. This is noticed due to the fact that both Sets 2 and 4 include brighter and open space areas. Contrarily, on Table 4.2 we observe that when training on Set 1 and testing on Set 5, the performance remains at high standards, because of the more distinct negative areas that are richer in feature representation. Hence, the attribute of our dataset that construes an object as an obstacle given its relative position, seems to be exploited at full extent when using a Vision Transformer pre-trained with MAE.

As a means to examine the semantic relevance of the extracted features, we performed a two dimensional t-SNE [108] embedding and visualized our dataset in Figure 4.4.

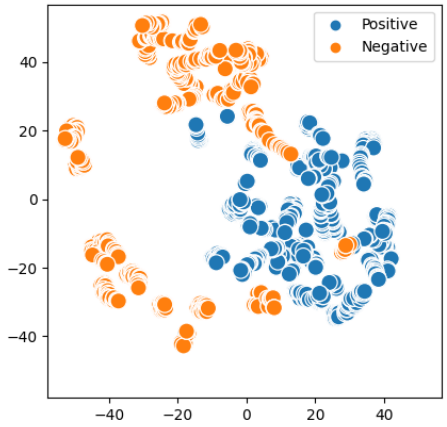


Figure 4.4: t-SNE embedding of features

We present the results on Set 5, when training on Sets 1, 2, 3, 4 combined, that attained the highest levels of accuracy. By observing this visualization, we can witness considerable amounts of spatial correlation. In specific, the x,y axes denote the outputs of the feature extractor as mentioned in Figure 4.1. Hence, we can infer that the features learnt, contain significant amount of semantic information needed for this binary image classification task.

4.6 Discussion & Challenges Encountered

We address a traversability estimation problem as a binary classification task among scenes that encompass objects during a mobile robot navigation. We show that transferring features using a Vision Transformer can act as a powerful paradigm for traversability estimation tasks in small datasets such our collected custom one. Fine-tuning a Vision Transformer pre-trained with MAE, led to a stronger performance than the one of a fine-tuned well-established state-of-the-art deep architecture for image classification such as ResNet. This work produced a high-level understanding

of the encountered scene in the vicinity of the robot. It was noted that the perception algorithm managed to have satisfactory performance on distinguishing between objects that can possibly act as obstacles within the robot's trajectory. However there are two main limitations. First, the labelling method, can be quite cumbersome when it comes to annotating large amounts of negative data and second, the lack of spatial information that can be the input to a motion planner. Along with the previous two aforementioned challenges, enriching the algorithm's perception with specific labels could provide some additional insight on the scene understanding.

In the following chapters we show how to address these limitations.

CHAPTER 5

RGB-based indoor multi-label classification using RGB instances

5.1 Overview

This work presents a method for extracting high-level semantic information through successful landmark detection using RGB images. In particular, the focus is placed on the presence of particular labels (open path, humans, staircase, doorways, obstacles) in the encountered scene, which can be a fundamental source of information enhancing scene understanding, and paving the path towards the safe navigation of the mobile unit. Experiments are conducted using a manual wheelchair to gather image instances from four indoor academic environments consisting of multiple labels.

5.2 Methodology

We expanded the use of the fine-tuned ViT model to perform multi-label image classification on wide-lens image data collected using a manual wheelchair. The overall approach follows a similar pipeline to the one described in Section 3.1.1. However, there is a distinction in the final output, where the model classifies images into one or more of the following classes: open path, doorways, staircase, humans, and obstacles (Figure 5.1). To handle multi-label classification, we employ the *BCEWithLogitsLoss* loss function. This loss function combines a Sigmoid layer and the Binary Cross-Entropy (BCE) loss in a single class, making it suitable for multi-label classification tasks.

$$l_c(x, y) = L = \{l_1, ...l_N\}^T, l_n = -w_n[y_n log\sigma(x_n) + (1 - y_n)(1 - log\sigma(x_n))]$$

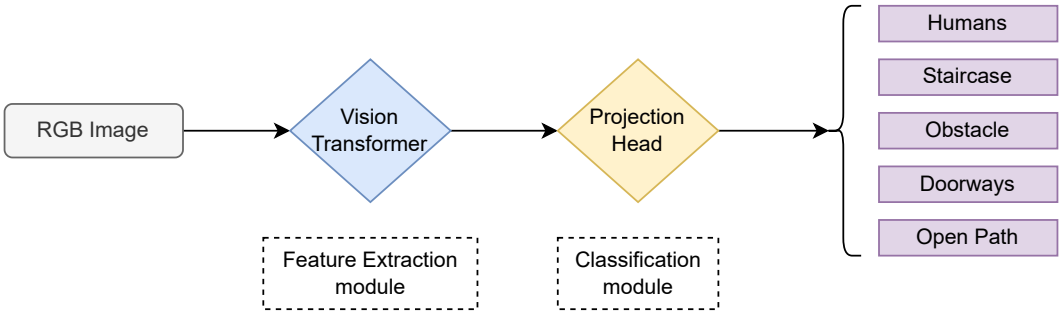


Figure 5.1: Pipeline of the proposed method

(5.1)

The choice of using the *BCEWithLogitsLoss* version of the Binary Cross-Entropy (BCE) loss is motivated by its superior numerical stability. This variant of the loss function employs the log-sum-exp trick, which helps mitigate numerical issues and instability that can arise during training, especially in situations involving large datasets and complex models.

5.3 Experimental setup

5.3.1 Hardware

Throughout the experimental process, a human operator navigated a standard wheelchair in four different buildings around the University of Texas, Arlington (UTA) campus. Data were recorded using a GoPro HERO10 camera, that records in 60 frames per second and it is mounted on the wheelchair seat (Figure 5.2). For each building, the wheelchair was navigated in safe areas such as hallways, doorways while encountering static (chairs, bins, tables, lockers) or dynamic (humans) obstacles. Also, ascending and descending staircases are targeted, as additional areas of interest.

Figure 5.2: The configuration used for the experiments consists of a GoPro HERO10 camera mounted on the seat of the manual wheelchair

5.3.2 Data Collection and Processing

Data were recorded for approximately 150 minutes and created a dataset of 2704 images. The initial image size was 1920x1080 pixels before resized to 224x224 pixels, to match the resolution of the pretrained dataset. All images were manually labeled. The dataset includes 2119 single-labeled images and 585 instances that comprise of various combinations of the labels (open-path, humans, staircase, doorway, obstacles). Among the multi-labeled images, 367 instances are described by two-labels and 218 instances by three-labels. Sets 1, 2, 3, 4 include 678, 697, 659, 670 image instances respectively. The labelling process is identical to the one presented in Chapter 4

5.3.3 Fine-tuning

For the conducted experiments, Pytorch[1] is used as the backbone framework. Training was done on a machine with 2 Titan RTX (24GB GDDR6 RAM, 4608 CUDA Cores) GPUs. Horizontal flip is performed as a means to augment the dataset. Training is taking place for 50 epochs, using the BCE loss function unless an early stopping callback terminated the trial upon observed convergence. Furthermore, the training parameters used were: batch size = 16, learning rate = 0.01 and weight decay = 5e-4. For the fine-tuning part, all transformer's deeper layers are frozen and the classifier is replaced with two fully-connected layers; the last one performs the classification. Layers were fine-tuned using stochastic gradient descent (SGD).

5.4 Results

To evaluate the performance of the proposed fine-tuned method on the custom dataset, an ablation study is conducted. Four-fold cross validation is performed on three buildings selected for training and the remaining one for testing. The rationale behind folding on the buildings is to exploit the visual dissimilarity between semantically equivalent classes between buildings. This comparison is going to help us evaluate the ability of the proposed method to generalize beyond learning visual representations of specific landmarks. Utilizing the same architecture for the projection head, a deep residual network(ResNet50) that has been pre-trained on ImageNet-21k, is fine-tuned. The classifier is replaced with the projection head for the classification.

Additionally, a GAN ensemble network is trained following the methodology described by Hirose et al. in [64]. We use the GO Stanford[2] dataset and pre-train on approximately 75k unlabeled fisheye images. Finally, a small convolutional network

[1] `https://pytorch.org/`

[2] `https://cvgl.stanford.edu/gonet/dataset`

is trained, comprising of four convolutional and two fully connected layers that each, except for the final, is followed by a ReLU activation function. Hamming loss is chosen as the performance metrics (as suggested in [109]) since it only penalizes the individual labels and we experiment with different values for τ. For both fine-tuned ViT and ResNet, the datasets that, after performing 4-fold cross validation, presented the highest (Set 4) and minimum (Set 3) hamming loss, are chosen. The results are shown in Figure 5.3.

The focus of this work's approach is heavily dependent on landmarks' detection and this is crucial to ensure safe wheelchair navigation. The detection of staircases, humans and miscellaneous static obstacles is prioritized by assigning a lower value for τ. Since humans' motion is governed by uncertainty and it is crucial to act in a conservative manner, given that predictions must align with the axis of safety, the best results in terms of humans detection, are achieved when $\tau_{humans} = 0.15$. Similarly, the best detection results for staircases, static obstacles, doorways and open paths were achieved when $\tau_{stairs} = 0.17$, $\tau_{obstacles} = 0.18$, $\tau_{doorways} = 0.15$, $\tau_{open} = 0.80$ respectively.

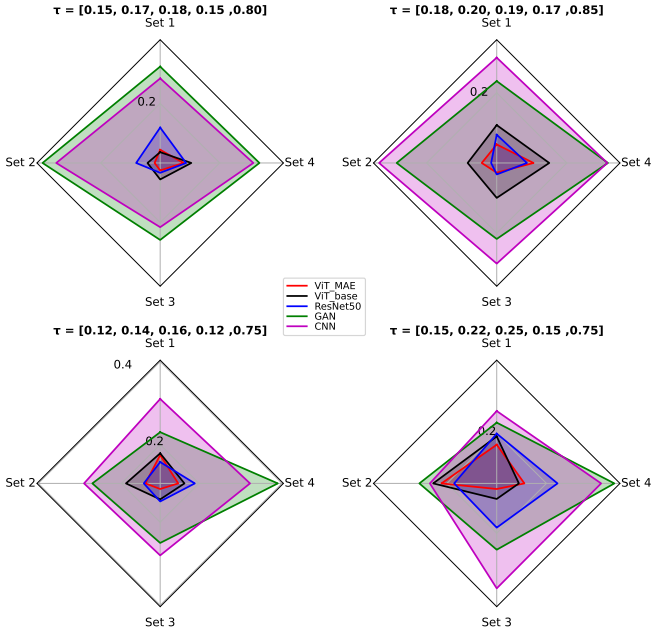


Figure 5.3: Methods' performance for various values of the threshold τ using the Hamming loss metric. Larger Hamming loss implies lower network performance

Figure 5.3 presents the results of the ablation study. Fine-tuned ViT$_{MAE}$ outperforms all other networks while displaying critical levels of consistency. This is in consensus with the results in literature [1], [90] in which ViT's performance can significantly outrun CNNs' in image classification tasks. This argument is also supported by the fact that MAE training includes the notion of learning visual semantics holistically. With regards to ViT-base-patch16-224, it does not demonstrate significant improvement compared to ResNet50. GAN's performance is lower, due to the difficulty in training the ensemble's networks with an adequate number of data whereas the custom fully supervised CNN did not exhibit major amounts of efficiency for practical tasks.

The lowest values of hamming loss, implying high levels of performance, are observed for Set 3. This is due to the fact that Set 3 displays considerable amounts of balance with respect to varying illumination and object features. Contrariwise, Set 4 presents the largest amounts of hamming loss because it is the one with the most uniquely distinct features in terms of visual information. Compared to the others sets, Set 4 is significantly more differentiated including the darkest of illumination as well as areas with dense concentration of bulky objects. The best performance of ViT_{MAE} is achieved when using τ values = [0.15, 0.17, 0.18, 0.15, 0.80] for humans, staircases, static obstacles, doorways and open paths respectively.

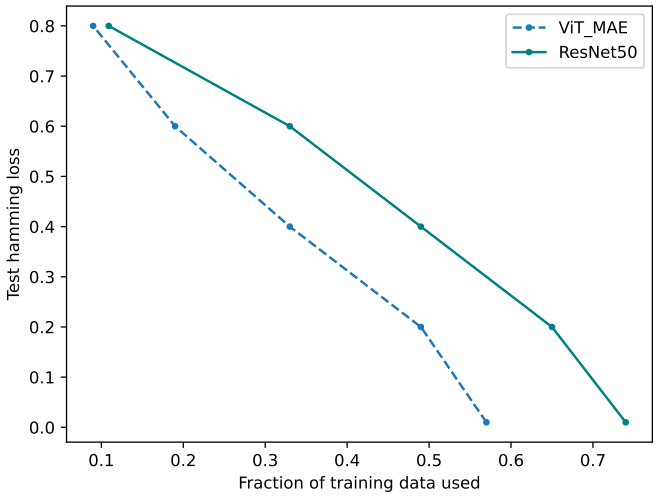


Figure 5.4: Graph of test hamming loss against fraction of training data used for Set 3

Figure 5.4 displays a comparison between the hamming loss as computed by fine-tuning the MAE and ResNet50 on Set 3 that exhibits the best performance. In specific, the fine-tuned ViT_{MAE} convincingly outperforms fine-tuned ResNet50, with the performance margin, described by the hamming loss, widening as the fraction of

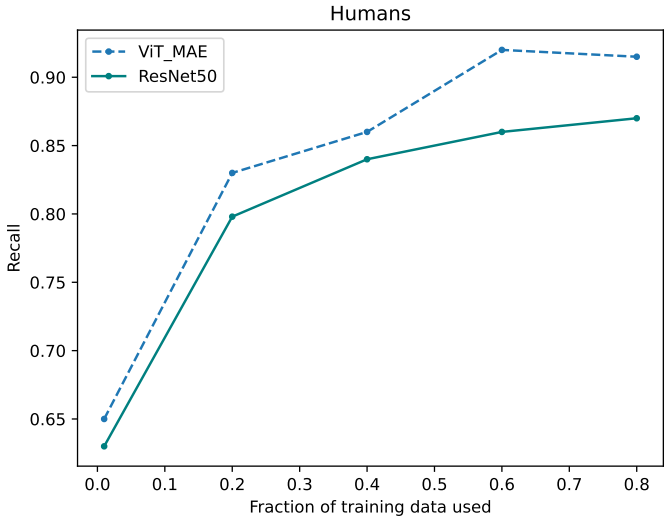


Figure 5.5: Comparison between the two prevalent fine-tuning methods for the "humans" label when testing on Set 3 for different amounts of training data

training data increases. Additionally, it is noticed that even for a small amount of training data available, ViT_{MAE}'s hamming loss is smaller than the ResNet50 one. This shows that ViT_{MAE} can be largely beneficial in scenarios where only a small amount of training instances is available. In Figure 5.5, the recall is examined as observed in Set 3 for the images that include the "humans" label. ViT_{MAE} consistently achieves around 86% recall for training sets larger than 40%, while ResNet50 achieves lower performance. Hence, it can be inferred that ViT_{MAE} can sufficiently address the presence of humans in the scene. Overall, the attribute of our dataset that construes an object as an obstacle given its relative position, seems to be exploited at full extent with the use of a Vision Transformer pre-trained with MAE.

The confusion matrices depicted in Figure 5.6 provide an illustrative representation of the ViT_{MAE}'s best performance as noted on Set 3. Overall, the detection performance achieves high levels of efficiency. In addition, the results are consistent along the various labels irrespective of the notable differences among the sets,

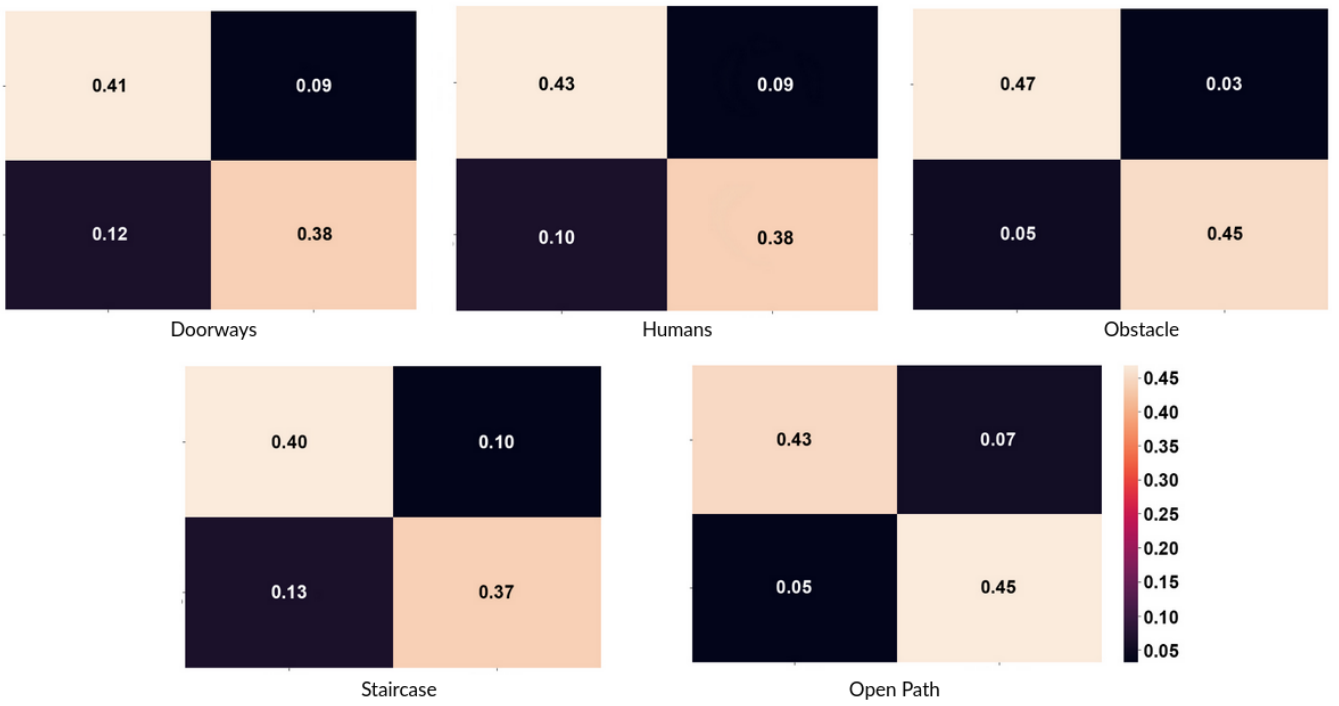


Figure 5.6: Confusion matrices for each label as observed in ViT_{MAE}'s best performance on Set 3

which were collected in different buildings. This can be attributed to the presence of pre-trained self-attention layers along with the property that Masked Autoencoders portray, which is to learn visual scene semantics in a comprehensive manner. The aforementioned arguments reinforce the claim that ViTs provide generalizable solutions to the multi-label classification problem for small datasets.

5.5 Discussion

A method that extracts high-level semantic information regarding the scene's navigability through landmark detection, is proposed. Experiments were conducted in different indoors environments using a manually driven wheelchair and a wide-lens camera. The results indicated that our multi-label classification method achieves high performance without the loss of generalization and enriches scene understanding.

Therefore, the proposed approach can act as a preceding step before designing the motion planning (autonomous or not) of a manual wheelchair.

Furthermore, the results showed that fine-tuning a Vision Transformer can act as a powerful tool for multi-label classification tasks in small datasets. We show that fine-tuning a Vision Transformer pre-trained with MAE, led to a stronger performance compared to state-of-the-art deep architecture for image classification such as ResNet. However, the lack of spatial information is again a challenge for labelling and also for prediction. The utilization and fusion of additional modalities (depth, laser), that, along with RGB images, could lead to deeper evaluation of the scene as well as incorporating spatial information.

CHAPTER 6

RGB-Laser Range Finder Image Classification

6.1 Overview

In this approach, we propose a dual-stream, semi-supervised, attention-based approach that employs feature fusion of RGB and Laser Range Finder (LRF) modalities. Our method leverages the strength of two powerful transformer-based networks, i.e. Vision Transformer (ViT) and SegFormer, along with LRF information, to adequately predict whether the scene encountered in the image is safe for a robot to traverse. Towards this effort, we introduce an automated labelling system profiting from the combination of raw velocity readings and laser scanning information. Moreover, we aim to show that overall GO/NO-GO detection is enhanced by fusing RGB and laser modalities through the employment of a Multi-Head Self-Attention (MHSA) module.

6.2 Methodology

6.2.1 Multi-Head Self-Attention

Multi-Head Self-Attention (MHSA) is a key mechanism in deep learning, particularly in the context of neural networks and natural language processing tasks. It's an extension of the standard self-attention mechanism, allowing a neural network to weigh the importance of different parts of an input sequence when making predictions or generating representations. In MHSA, the input sequence is linearly transformed into three sets of vectors: Query (Q), Key (K), and Value (V). These transforma-

tions allow the network to learn different aspects of the input sequence. This can be mathematically represented as:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V$$

Where: X is the input sequence, W^Q, W^K, and W^V are learnable weight matrices.

6.2.1.1 Scaled Dot-Product Attention

For each position in the input sequence, MHSA calculates the attention score by taking the dot product of the Query vector for that position with the Key vectors for all positions in the sequence. The scores are then scaled to prevent gradients from becoming too small or too large. This operation reflects how much each position should attend to other positions in the sequence. The attention scores can be calculated as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Where: softmax is the softmax function, d_k is the dimension of the Key vectors.

MHSA employs multiple attention heads in parallel. Each head projects the input into different subspaces and calculates attention scores independently. These multiple attention heads capture different patterns and relationships within the input sequence, providing the model with a richer set of information. The output of each head is concatenated and linearly transformed to obtain the final output:

$$\text{MHSA}(X) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W^O$$

Where: h is the number of attention heads, Concat denotes concatenation, W^O is a learnable weight matrix.

6.2.2 Advantages and Applications

MHSA has become a fundamental building block in many state-of-the-art deep learning models due to its ability to capture long-range dependencies in sequences, parallelization, interpretable representations, and adaptability to various types of data. What is more, it has found applications in various natural language processing tasks, including machine translation, text summarization, named entity recognition, and language modeling.

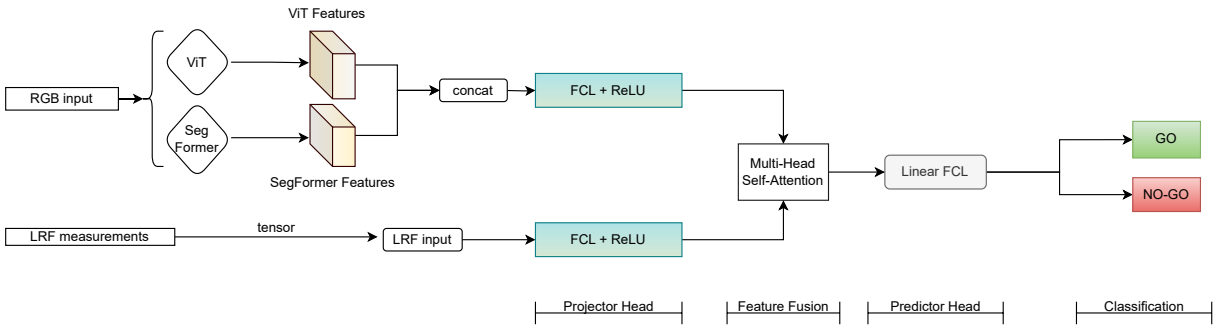


Figure 6.1: Proposed Methodology

6.2.3 Our Approach

In this effort, we proposed a dual-stream architecture that exploits feature fusion from two pre-trained backbone architectures, that have been trained on two closely related tasks, i.e. image classification and semantic segmentation. Particularly, we are using two networks pretrained on ImageNet-1K: 1) a ViT pre-trained in a self-supervised manner using the *Masked Siamese Networks* technique [110] and

2) a SegFormer. The core idea is to combine contextual information from the two aforementioned backbone architectures, that have been trained to perform semantic interpretation in a general (image-level) and more fine-grained (pixel-level) degree.

Pre-trained ViT is reinforced by the Masked Siamese Networks (MSN) technique [110], a crucial ability to learn representations with high semantic information, is introduced through matching masked prototypes with unmatched patches, given two views of the same image. MSN exhibits excellence in transfer-learning tasks and especially in cases where labeled data are limited. This attribute is in consensus with the goal of the task, which is to examine the accuracy, in terms of detection, and the ability to generalize in cases where the number of labeled data is insufficient to train a larger model from scratch.

On the other end, the SegFormer is merging the Transformer's architectural backbone with a lightweight decoder. In contrast to ViT, its *Mix Transformer* encoder (MiT) does not use any positional encodings and can generate multi-level feature maps (both high-resolution fine features and low-resolution coarse features) due to its hierarchical structure. Furthermore, a series of lightweight (few number of parameters) Multi-Layer Perceptrons (MLP) is used as a decoder, that exhibits the attribute of combining both local and global attention, and eventually creates powerful and meaningful representations accompanied by strong generalization capabilities. This last aspect, along the dominant multi-scale feature learning ability of the SegFormer's encoder, can be proven to be crucial for our method; in specific we use a pre-trained on ImageNet-1K SegFormer-B0 and replace the classification head for fine-tuning on our dataset.

As a mechanism to enrich the semantic information, before training, we concatenate the RGB feature vectors extracted from both the ViT and the SegFormer. Furthermore, the MHSA module has displayed remarkable adaptability in fusing

modalities due to its unique characteristic of computing the attention weights for the input and producing an output vector with encoded information on how each segment of information should be combined with the rest. Therefore, driven by this unique property of the MHSA module, we aim to fuse RGB and laser features as a means to exploit the source of semantic and spatial information of the encountered scene.

For the supervised fine-tuning structure, we use a simple projection head consisting of two fully-connected (FC) layers, each followed by a ReLU activation function. The first FC layer is trained using the combined weights from the two pre-trained networks, ViT and SegFormer respectively. In particular, we replace the classifier heads of the pre-trained ViT and SegFormer with two layers which dimensions are 768x1 and 256x1 respectively, before concatenating them and producing one 1x1024 RGB feature vector. On the other end, the raw LRF measurements are fed to a 1x1081 tensor. RGB and laser input are passed through two FC layers (1024x512 and 1081x512 respectively), before fed to the MHSA module that uses 8 heads and performs the fusion. Ultimately, the GO/NO-GO scene classification is performed by a 512x2 linear FC layer (Figure 6.1).

6.3 Experimental Setup

6.3.1 Data Collection & Annotation

As in Chapter 4 we are using the dataset gathered by the experimental process tele-operating the robotic platform around university buildings and hallways. However the labelling method differentiates as a means to include spatial information. Especially, by using the PU method that learns from positive and unlabeled data [111], we aim to divide each source set corresponding to each domain, into two

subsequent subsets filled with positive and unlabeled image instances respectively. In our previous work 4, we noticed that relying only on the wheels' velocity could result in an error-prone automated annotation. To enhance the reliability of the automated annotation process, we incorporated information from the laser scanner. This addition allows for a more comprehensive understanding of the surroundings and helps address challenges arising from situations where humans appear unexpectedly or when odometry data alone is insufficient.

The labelling process is similar to the one described by Hirose et al. [64], yet supplemented with laser information. Specifically, we examined the velocities corresponding to all four wheels of the robot and define a heuristic threshold value of 1 m/s and a time window of 2.5 seconds. We set the following three conditions to be concurrently satisfied:

- If within this window, for each wheel, the velocity value v_t is constantly above the threshold value of 1 m/s
- The robot is moving forwards (or turning)
- There is no obstacle within a range of 1.2 meters from the laser sensor. This is a heuristic value that takes into consideration the size of the robot

If all the above conditions are true, we determine that the central frame, during this window, portrays a traversable scene and it is labelled as positive. For any other cases not adhering to the above conditions, the frame is unlabeled. Initially, using the above process, we labeled the entirety of the dataset. Afterwards, two datasets depicting the positive and the unlabeled respectively, were created. For training the supervised model, we utilized a combination of automatically labeled positive data and carefully selected negative samples. The negative samples were chosen from the unlabeled dataset and represented non-traversable scenes, including static obstacles, narrow paths, staircases, and humans.

6.3.2 Implementation Details

We used the PyTorch[1] framework as the basis of our experiments. Training was done on a machine with 2 Titan RTX GPUs (24GB GDDR6 RAM, 4608 Cuda Cores). Using the standard cross-entropy loss function, we trained for 50 epochs unless an early stopping callback terminated the trial upon observed convergence. As training parameters we used: batch size = 16, learning rate = 0.01 and weight decay = 5e-4. The fine-tuning of the layers was accomplished using stochastic gradient descent (SGD). All images' initial dimensions of 640x480 pixels were re-scaled to 224x224 using the default PyTorch interpolation.

6.4 Results

Table 6.1: Accuracy[%] per method for 5-fold cross-validation

Method	Test on Set 1	Test on Set 2	Test on Set 3	Test on Set 4	Test on Set 5
$\text{RGB}_{ViT,SEG}$ + LRF	84.77	89.22	93.38	91.66	89.73
$\text{RGB}_{ViT,SEG}$	82.48	89.57	90.28	91.85	87.15
RGB_{ViT} + LRF	83.11	80.33	87.94	86.57	85.60
RGB_{SEG} + LRF	80.93	82.28	85.66	84.19	83.55
RGB_{ViT}	79.24	84.19	86.34	85.21	83.42
RGB_{SEG}	78.76	85.29	84.64	83.92	82.30

6.4.1 Ablation Study

We implement a 5-fold cross-validation on all five domains of the collected dataset, using 4 folds for training and the remaining one for testing. By folding on the different buildings, we aim to exploit the visual dissimilarity between semantically equivalent classes that occur throughout the dataset. This comparison enables us to

[1] `https://pytorch.org`

evaluate the ability of the proposed method to generalize beyond learning visual representations of specific objects. After experimenting with different numbers of data instances for each set, we noticed that the best performance of the supervised classifier is achieved when using 2800 samples i.e. 1400 for positive and 1400 for negative, except for Set 1 that due to its limited size, 1400 images are used in total (700 positive, 700 negative).

Table 7.1 presents the results of the cross-validation for different RGB and LRF combinations. We observe that, the fused $\text{RGB}_{ViT,SEG}$+LRF ensemble is, overall, the most performant method. Specifically, in domains that include robust and distinct obstacles, such as Set 1 and Set 5, the integration of spatial information (through the use of laser) with RGB features from both pre-trained networks, contributes to the overall detection performance. Additionally, it surpasses by a large margin the efficacy of the methods that use RGB features from only one pretrained RGB network (RGB_{ViT} + LRF, $\text{RGB}_{ViT,SEG}$+LRF, RGB_{ViT}, RGB_{SEG}). This highlights the importance of using both pretrained ViT and SegFormer RGB features. On the other hand, in domains that are mostly described by open paths, such as Set 2 and Set 4, we note that the performances of $\text{RGB}_{ViT,SEG}$+LRF and $\text{RGB}_{ViT,SEG}$ are comparable. As a consequence, the effect of semantic homogeneity in training samples is demonstrated; for some domains, the dependence on purely semantic information seems to be adequate enough for the classification task. This argument reinforces the notion that, data annotation using laser information can genuinely influence and boost the overall detection performance. For the remaining of the paper, we denote "$\text{RGB}_{ViT,SEG}$+LRF" as "RGB+LRF" and "$\text{RGB}_{ViT,SEG}$ "as "RGB_{only}".

6.4.2 Domain transferability

In order to determine the transferability potential of the RGB+LRF method, we perform one-versus-one domain evaluation. We choose Sets 1 and 4 as the training sets, since they exhibit the largest and smallest concentration of obstacles respectively. Thereupon, we carry out individual testing on each of the remaining four sets, using the RGB+LRF and the RGB_{only} methods. Since the dataset becomes less balanced, the F1-score is used as a metrics to take into account both the false positives and false negatives. Results are shown in Figure 6.3 and both methods, display relatively adequate transferability capacity. We observe that the inclusion of spatial information is yielding improved results especially between domains of spatial homogeneity, including obstacles and compact structures (Set 1). In addition, the use of laser is proving its efficiency in addressing different domain illumination. This is noted by the performance of RGB+LRF, for instance when 1) training on Set 1 (dark illumination) and testing on Set 5 (bright illumination) and 2) training on Set 4 (bright illumination) and testing on Set 1 (dark illumination).

6.4.3 Pre-trained features and training dataset size

To examine the performance of RGB+LRF with respect to the proportion of used training data, we evaluate on Set 3 (best accuracy) and Set 1 (worst accuracy). In Figure **??**, we observe that RGB+LRF exhibits decent efficiency for both sets, even when the fraction of available training data is significantly low. These findings assert that the proposed method's performance is mostly influenced by the utilization of the pre-trained RGB features, extracted from the ViT and the SegFormer. Therefore, it becomes apparent that only a small number of task-specific labels is required. The aforementioned argument is aligned with the idea of exploiting a pre-trained ViT with

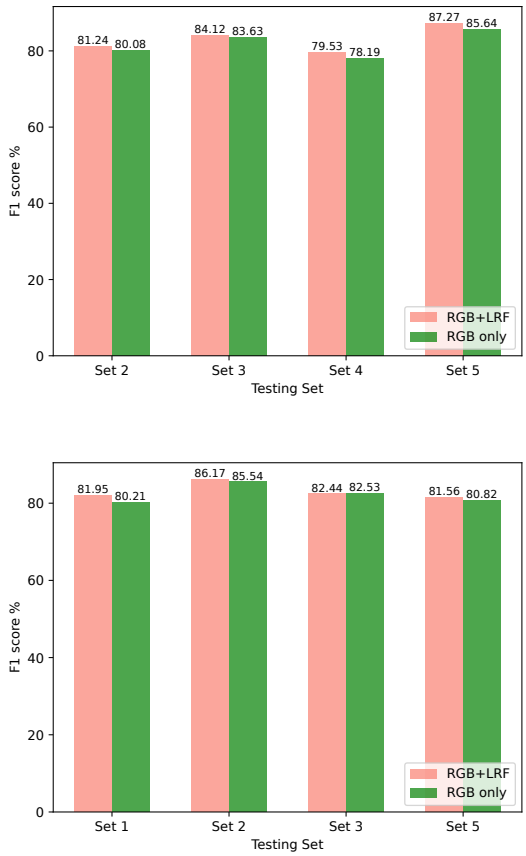


Figure 6.3: Transferability performance when training on Set 1 (left) and Set 4 (right)

the MSN technique and also a SegFormer, to achieve adequate generalizability with scarce training samples.

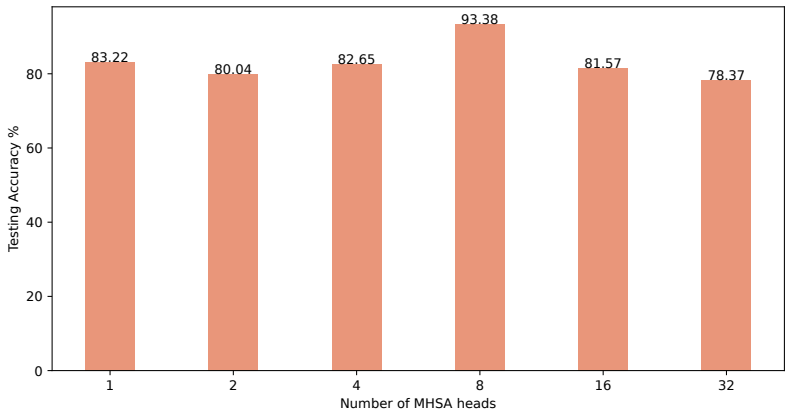


Figure 6.4: Relationship between testing accuracy and different number of MHSA heads when testing on Set 3

6.4.4 Optimal selection of the number of MHSA heads

We experiment with a different number of heads (h) for the MHSA module. The value of h should divide the embedded dimension of the Multi-Head Self-Attention layer, which in our case is 512. As illustrated in Figure 6.4, the RGB+LRF method achieves best results on Set 3, for h = 8. This is in accordance with the results presented in literature [112], and state that a larger number of heads can lead to increased performance. Nonetheless, we notice a degradation in performance, due to the fact that a sharper increase from 8 heads and above, is accompanied by a significant rise of the model's learnable parameters.

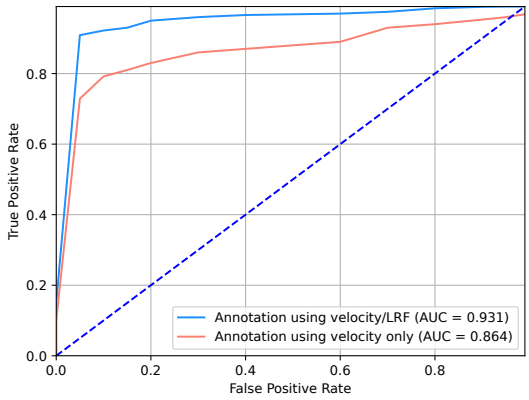


Figure 6.5: ROC analysis curves for the RGB+LRF method when used for different annotation techniques on Set 3

6.4.5 Significance of laser annotation

Figure 6.5 shows the performance given two automated annotation techniques: velocity and laser against solely velocity, as noted on Set 3 (best performance). We observe that the RGB+LRF fusion method achieves better results when using both laser and velocity for annotation, resulting in higher true positive rates given the false positive rate. What is more, AUC (Area under the ROC Curve) reaches a value of 0.931, which is equivalent to a likelihood of 93.1% for the fine-tuned RGB+LRF model to distinguish between a GO and a NO-GO class when using laser for the annotation.

6.5 Discussion & Challenges

Initially, we show that by combining the collected velocity readings with laser scans, a compact automated annotation framework is created. Moreover, we show that the feature fusion of RGB and laser modalities, through the use of two strong pre-trained networks (ViT and SegFormer) and the Multi-Head Self-Attention (MHSA)

module, can lead to better detection efficacy and transferability among different domains. Enriching our dataset with instances from additional domains of different illumination and objects' structures, even including outdoor conditions, may serve as a further experimental point of comparison. The findings suggest that this fusion strategy significantly improves detection efficacy and the adaptability of models across diverse domains.

While the architecture described in this chapter offers merits in terms of prediction, it should be noted that the overall architecture demands a significant amount of computational resources. This can pose a challenge when deploying the algorithm, depending on the available hardware. Furthermore, while a high-level understanding of estimating traversable scenes may be an important step in grasping the scene's semantics, real-time navigation demands that the agent possesses spatial insight about the environment. Therefore, one pending challenge that needs to be addressed is determining specific traversable segments within complex scenes.

CHAPTER 7

Free-space Segmentation

7.1 Overview

Driven by the need to understand traversable regions and deviating from strictly examining high-level traversability estimation techniques, we propose an indoors free-space segmentation method that associates large depth values with navigable regions. This method leverages an unsupervised masking technique that, using positive instances, generates segmentation labels based on textural homogeneity and depth uniformity. Moreover, we generate superpixels corresponding to areas of higher depth and align them with features extracted from a Dense Prediction Transformer (DPT).

7.1.1 Overall Methodology

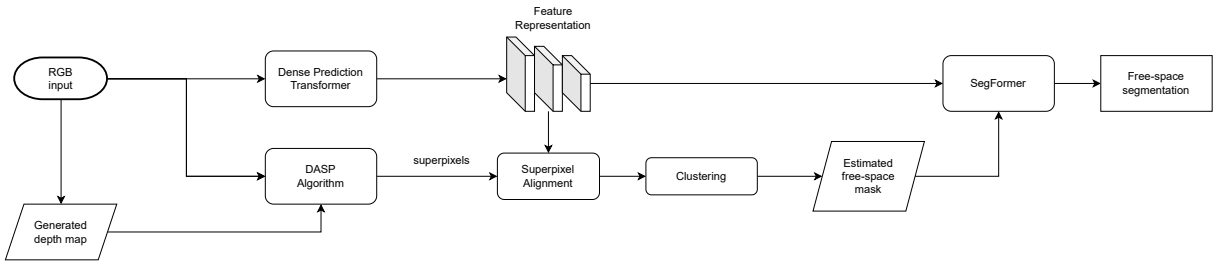


Figure 7.1: Proposed methodology

Figure 7.1 illustrates the proposed methodology's architecture. Our approach is based on the central hypothesis that areas classified as traversable free-space will generally exhibit greater depth values. To validate this hypothesis, we focus on training a refined semantic segmentation model using positive RGB indoor instances that the

robot can safely traverse, such as hallways and wide paths. Our method relies solely on the RGB data we collected, while for generating the depth maps, we utilized the depth estimation model provided by Niantic Labs [113], allowing to extract valuable depth information from the aforementioned RGB instances.

Additionally, we propose an unsupervised mask generation framework that leverages depth information to capture textural homogeneity and depth uniformity as indicators of free space. Subsequently, we evaluate the performance of the model on challenging scenarios where free space is present yet restricted due to the presence of cluttered objects.

7.1.2 Mask Annotation

To generate annotation masks for free-space segmentation, we build upon the following main hypothesis: Free space is expected to be depicted by textural homogeneity in conjunction with depth uniformity. This means, that in order to search for free space in the image, we consider these particular superpixels that are characterized by similarity in both color and depth information. Therefore, we use a superpixel oversegmentation algorithm to generate unsupervised free-space masks by capturing intra-region similarities towards producing meaningful segments: the Depth-Adaptive Superpixels (DASP) algorithm [114].

Overall, our method is influenced by the findings and approaches outlined in [115], however we differentiate in the sense that "seed" superpixels are generated through the guidance of depth information. In lieu of empirically computing the parameters that define the location prior, we use the DASP algorithm to generate the seeds needed for segmentation.

Figure 7.2: Upper row: RGB-D pair Bottom row: On the left, the DASP algorithm performs oversegmentation, dividing the image into superpixels. On the right, the seeds generated by the DASP algorithm, described by the dotted area, represent the area with the greatest depth

7.1.3 Features Extraction

We intend to associate superpixel clusters with features that have been extracted by a Dense Prediction Transformer [116], that uses the *Vision Transformer* (ViT) architecture as a backbone for dense prediction tasks such as semantic segmentation. DPT leverages the strengths of 1) Transformers, known for their ability to model long-range dependencies, and 2) Convolutional Neural Networks (CNNs) that can efficiently process and encode spatial information within the input images. Consequently, DPT allows for efficient capture of both local and global information. We remove the last layer of the pre-trained Intel's DPT-Large[1] model, and the resulting

[1] `https://github.com/isl-org/DPT`

extracted features, in the form of a vector of size 577x1024, exhibit rich high-level semantic information as well as spatial details at a higher resolution.

7.1.4 DASP

DASP utilizes spectral graph theory to calculate segments through graph cuts. Essentially, it creates a graph where each pixel is represented as a node, and the edges represent the similarity between pixels. The similarity computation takes into account both color and depth information and this distinctive property is of fundamental importance for our method i.e. estimating segments that exhibit the largest depth. An additional asset that DASP displays and can be substantial towards estimating the segments of higher importance, is the generation of segmentation seeds that are expected to be associated with free space. DASP achieves this by calculating the depth gradient of the input image and clustering it, to eventually obtain a collection of initial segmentation seeds. Hence, instead of relying on a location prior as in [114], we can utilize the seed coordinates that guide the segmentation towards areas of larger depth (Figure 7.2).

7.1.5 Superpixel alignment

To establish a spatial association between the distinct features derived from the DPT and the coherent superpixels generated by the DASP algorithm, we use a sophisticated technique known as *superpixel alignment* inspired by the work by Tsutsui et al. [115] and influenced by the concept of RoIAlign [116].

Firstly, a selection of ten representative locations is made within each superpixel, forming crucial anchor points within these coherent regions. Next, for each of these chosen anchor points, the algorithm identifies the four nearest neighbors of

each selected pixel for the bilinear interpolation. This process creates a local context around each anchor point, facilitating a focused examination of the surrounding area.

In order to effectively associate the features extracted by the DPT with the representative anchor points within superpixels, the technique of bilinear interpolation is implemented. Specifically, it estimates the features at the anchor points by considering the features of the nearby pixels, and thus seamlessly integrating the two sets of information. By projecting the features extracted from the neighboring pixels onto the representative anchor points using bilinear interpolation, a direct alignment between DPT features and superpixels is achieved.

As a result, this alignment is crucial in ensuring that the information extracted by the DPT accurately corresponds to the meaningful regions represented by the superpixels. Afterwards, we employ average pooling to aggregate the features within each superpixel and therefore generate a comprehensive representation of the superpixels' characteristics.

7.1.6 Superpixel clustering

We aim to guide the segmentation towards traversable areas that exhibit the largest depth. Therefore, we cluster superpixels with respect to the semantic features and additionally, we select the cluster that corresponds to the area with the largest depth. As highlighted by Weikersdorfer et al. [114], the density of superpixel clusters in the image space is computed from the depth image. For detailed information on how this density is computed, the reader is referred to the original article.

The steps of the DASP algorithm can be summarized as follows: 1) the density of the depth-adaptive superpixel clusters is calculated based on the depth input image 2) an initial set of cluster centers is obtained by applying a Poisson disc sampling method 3) these sampled cluster centers are utilized in a density-adaptive local

iterative clustering algorithm to assign pixels to their respective cluster centers. We anticipate that the superpixels representing open areas will have lower density values.This is because there are fewer superpixels/clusters required to represent those areas due to their relatively uniform or homogeneous nature.

Consequently, our objective is to identify the clusters within the Poisson disk distribution that are most likely to define free space. For a given depth image, we estimate the superpixel density values, and using the k-means clustering technique, we single out the clusters. In a similar fashion to [115], the initial cluster is encouraged to include pixels associated with free space by positioning its cluster center as the average of features weighted by their spatial distribution. In contrast, the rest of the clusters have a repelling weight assigned to their members, prompting them to spread out spatially from their previous location.

7.1.7 Fine-tune a SegFormer

Using the masks produced at the previous step, we fine-tune a SegFormer model on our custom-collected dataset. The SegFormer is merging the Transformer's architectural backbone with a lightweight decoder. In contrast to ViT, its *Mix Transformer* encoder (MiT) does not use any positional encodings and can generate multi-level feature maps (both high-resolution fine features and low-resolution coarse features) due to its hierarchical structure. Furthermore, a series of lightweight (fewer number of parameters) Multi-Layer Perceptrons (MLP) is used as a decoder, that exhibits the attribute of combining both local and global attention, and eventually creates powerful and meaningful representations accompanied by strong generalization capabilities. This last aspect, along the dominant multi-scale feature learning ability of the SegFormer's encoder, can be proven to be crucial for our method; specifically, we use

a SegFormer-B0 pre-trained on ImageNet-1K, and replace the classification head for fine-tuning on our dataset.

7.1.8 Implementation Details

We used the PyTorch[2] framework as the basis of our experiments. Training was done on a machine with 2 Titan RTX GPUs (24GB GDDR6 RAM, 4608 Cuda Cores). Using the standard cross-entropy loss function, we trained for 50 epochs unless an early stopping callback terminated the trial upon observed convergence. As training parameters we used: batch size = 16, learning rate = 0.01 and weight decay = 5e-4. The fine-tuning of the layers was accomplished using stochastic gradient descent (SGD). All images' initial dimensions of 640x480 pixels were re-scaled to 224x224 using the default PyTorch interpolation. In the following sections, we present the best results achieved when using 1800 of positive instances for training, and 453 challenging instances for testing.

7.1.9 Performance Analysis

We train our method on the positive instances and test on the challenging ones. For our experiments, the standard semantic segmentation metrics, Intersection over Union (IoU) is used and we found that the best results were achieved for k=5 clusters.

Table 7.1 provides a summary of the performance of different methods under various input and model configurations in terms of IoU scores. Overall, the results show that incorporating depth information improves the accuracy of free-space segmentation compared to using only RGB images. We observe that our method, using the aforementioned unsupervised annotation framework, outperforms the corresponding SegNet-based method from [3] while also maintaining comparable performance

[2] `https://pytorch.org`

versus its supervised counterparts. Moreover, our approach demonstrates considerable adaptability, by performing well when fine-tuning with different model architectures i.e. U-Net and SegNet.

7.1.10 Impact of the number of training data used

Figure 7.3 illustrates the correlation between the number of training instances and the accuracy in free-space segmentation. The number of positive data used, plays a crucial role in the learning process of our method since, it helps the algorithm learn a wider range of patterns and variations associated with traversable areas. We notice that initially, as the number of instances increases, the IoU scores show a significant improvement. However, beyond the threshold of 1800 images (84%), it is observed that a further increase of the number of instances has a diminishing impact on the IoU scores. These findings indicate that even with a limited dataset, the proposed method achieves satisfactory performance, reinforced by the substantial ability of the SegFormer to capture complex spatial dependencies.

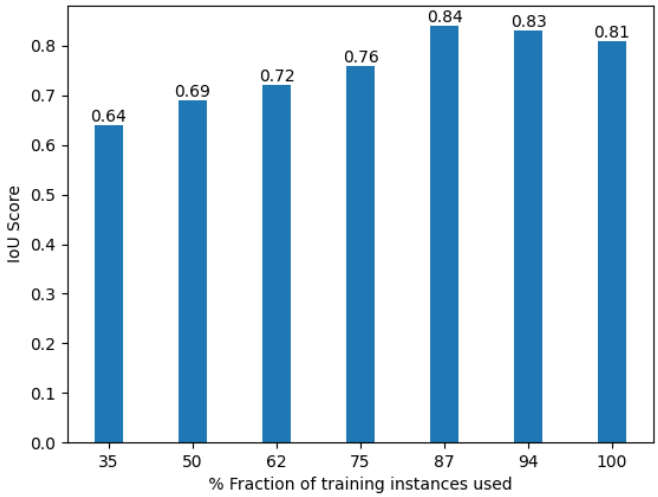


Figure 7.3: Method's performance for different number of training instances

7.1.11 Effect of the number of clusters

Figure 7.4 illustrates the relationship between the IoU performance and the number of clusters used in k-means. Increasing the number of clusters leads to improved segmentation performance, ought to the gradually increasing capability of the algorithm to capture depth and texture variations. It should also be remarked that after a certain number of clusters, for instance k = 6, we note a steady degradation in performance due to over-segmentation caused by the fine-grained clusters, capturing noise and redundant details.

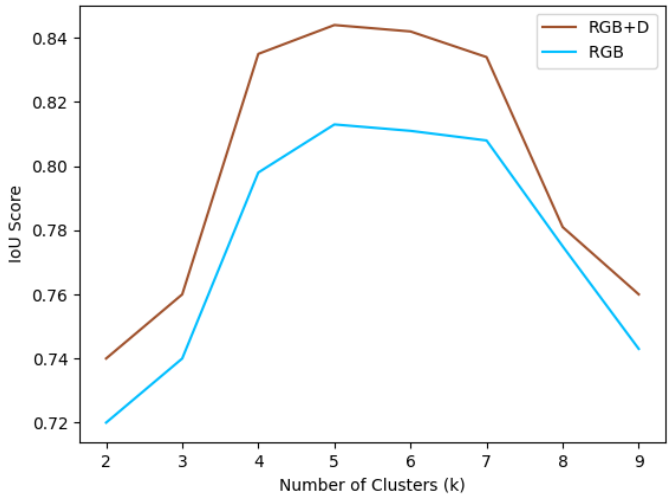
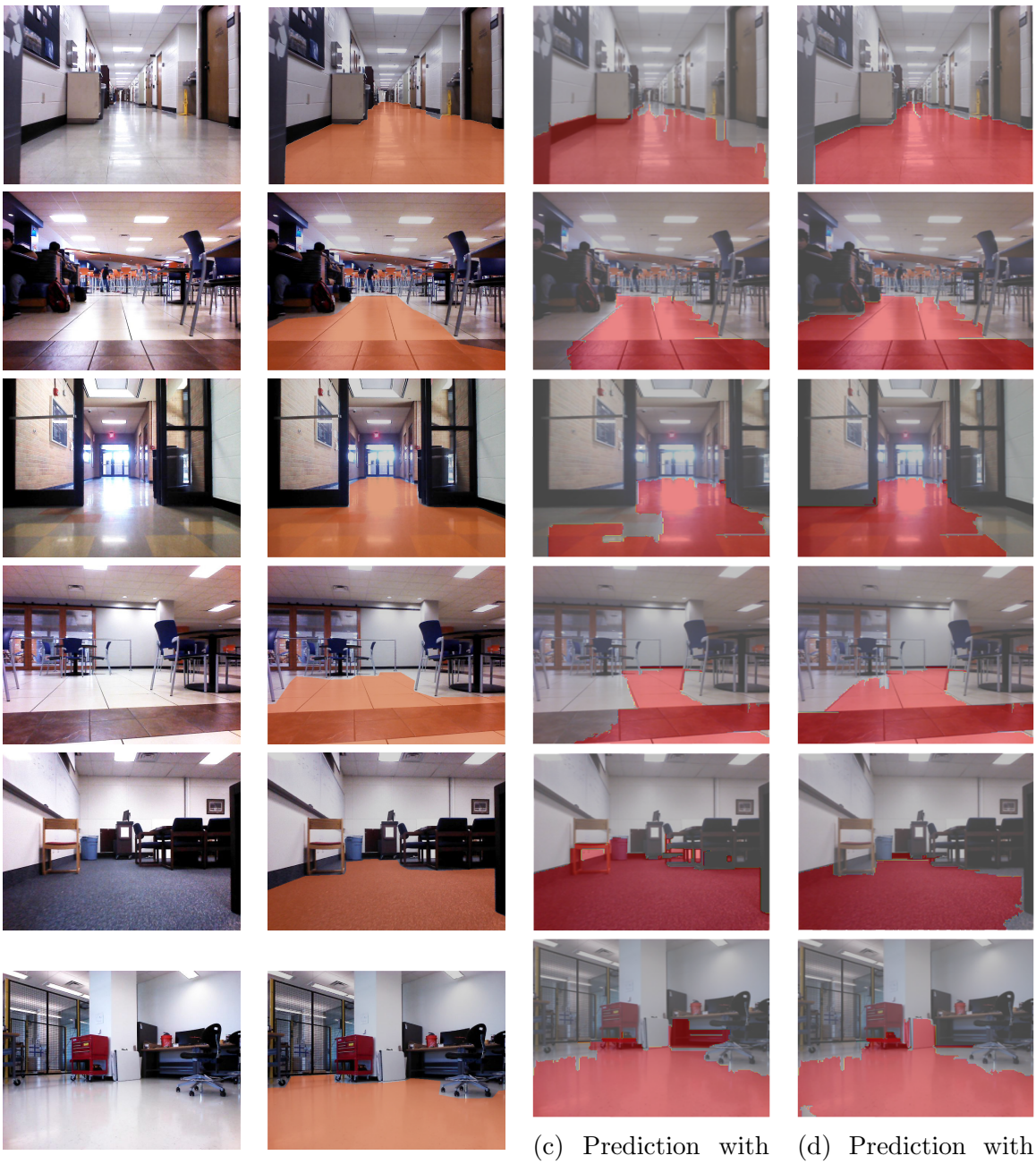


Figure 7.4: Method's performance for different number of clusters

7.1.12 Qualitative results

Figure 7.5 presents some qualitative results on the test dataset. We compare the performance of our method, that uses RGB information during training to generate masks, against the method presented by [3] and only uses RGB. Overall, it is noted that incorporating depth information enhances the accuracy of free-space prediction



(a) Input Images (b) Ground truth (c) Prediction with RGB (d) Prediction with RGB+Depth

Figure 7.5: Illustrative examples of the method's performance, last two rows depict erroneous results

compared to relying solely on RGB data. That is because depth information can provide valuable cues for free-space segmentation in the narrow passages along with some depth discontinuities, that help distinguishing the objects' edges and boundaries from the surrounding free-space regions. SegFormer's proficiency to effectively capture the long-range dependencies and spatial information, along with the features of the pretrained DPT within in the input data, is of instrumental importance. It enables the proposed algorithm to accurately differentiate between free-space regions and obstacles, even in challenging scenarios with objects, doors, and varying textures.

Besides, the use of positive instances during training, allowed the algorithm to prioritize the distinctive features and areas depicting free-space regions. However, some erroneous, false positive predictions were noted (last two rows in Figure 7.5) as well. This can be attributed to the presence of certain objects, which creates visual similarities with free space regions, consequently leading to misclassifications.

Upon examining the corresponding depth maps in Figure 7.6, it becomes evident that the accurate information provided by the depth map does not have an impact on the observed misclassifications. Hence, we can confidently conclude that these errors are a result of our method's performance. More specifically, in the example of row 5/Figure 7.5, the error can be attributed to the fact that the algorithm does not consider the height of the encountered furniture (in this case the instructor's lectern), which highlights the challenge of ambiguous semantics. Also, with respect to the sixth row of Figure 7.5, the tile positioned against the wall leads to a misunderstanding of texture, because it resembles a traversable surface that the algorithm has been previously trained on.

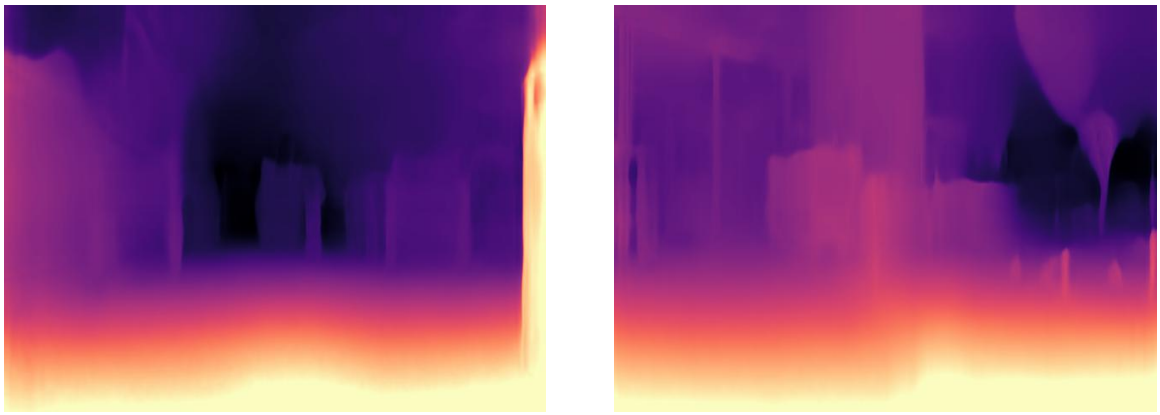


Figure 7.6: Depth maps for the wrong predictions (for rows 5 and 6 respectively) of Figure 7.5

Table 7.1: Performance of each method given different inputs

Method	**RGB**	**Depth**	**Training Data**	**Model**	**IoU Score**
[3]	✓	×	Automated	SegNet	0.773
[3]	✓	✓	Automated	SegNet	0.801
Ours	✓	×	Automated	SegFormer	0.813
Ours	✓	✓	Automated	SegFormer	0.844
Ours	✓	✓	Hand-labeled	SegFormer	0.878
Ours	✓	✓	Automated	U-Net	0.813
Ours	✓	✓	Hand-labeled	U-Net	0.849
Ours	✓	✓	Automated	SegNet	0.824
Ours	✓	✓	Hand-labeled	SegNet	0.862

7.1.13 Discussion

Erroneous classifications established an avenue for future research. One approach could to tackle this error would be leveraging the additional context provided by 3D point clouds by considering both visual cues and the physical positions and heights of objects. Also, experimenting with larger and more diverse datasets of indoor environments could contribute to further refining the algorithm's performance and generalizability.

Our error analysis shows that future research should be directed towards addressing misclassifications due to misleading texture. The area under the chair (row

5, Fig. 7.5) and the tile propped against the wall (row 6, Fig. 7.5) are characteristic examples: The algorithm correctly correlates the texture of tiles with traversable space, but fails to take into account the placement and position of the tile. This is an instance of the more general problem of how to combine common-sense (and, generally, structured) knowledge with machine learning. Future work can explore a more explicit combination between textural and geometric features. Such an approach would allow the system to represent findings such as "tiles propped up are obstacles" and "depending on height, overhead obstacles might make the area under them non-traversable" without jeopardizing the generalization that "tiles are normally laid over traversable areas".

CHAPTER 8

Conclusion and Future Directions

In this chapter, we elaborate on the research findings presented in this thesis and discuss potential future research directions in the field of improving indoors traversability estimation.

8.1 Summary of Findings

This study aims to explore various aspects of estimating indoors traversability estimation. At a higher level, we demonstrate that relying solely on semantic information cannot ensure secure navigation within the environment, primarily because of the RGB sensor's constrained situational awareness. This underscores the importance of incorporating spatial and geometric information, which can complement each other in a collaborative manner to enhance overall perception and safety.

We initially gathered a dataset comprising of various scenes including static and dynamic objects that can pose a threat to the robot's safety. Afterwards, different approaches and perspectives regarding the sensory input used to infer the traversability of indoor scenes, were examined. One of the key findings is the effectiveness of fusing multiple modalities, particularly combining visual data with Laser Range Finder (LRF) information. This fusion of modalities has shown promise in improving the accuracy of traversability estimation especially in scenes with cluttered objects and when examining the transferability of the method between two domains that were governed by levels of different ambience lighting.

The Vision Transformer (ViT) has played a crucial role in our research, showcasing its potential for feature extraction. By fine-tuning a pre-trained ViT model on our down-stream task (either image classification or semantic segmentation), we have witness improved algorithm's performance. With respect to free-space segmentation, we utilized a method that assumes that larger depth values correspond to areas with higher traversability. We employ an efficient automated masking technique that leverages textural homogeneity, depth uniformity, and positive scenes to create meaningful segments. Finally, we fine-tuned a SegFormer model on our custom-collected dataset, training on positive instances and testing on challenging ones, achieving satisfactory performance.

Nevertheless, the experiments we conducted, exposed certain limitations, including susceptibility to sensor noise, issues arising from occlusions, and the dynamic nature of indoor environments. These challenges underscore the need for the development of robust and adaptable traversability estimation methods. Future research in this domain could focus on several promising avenues:

- Integration of 3D Point Clouds: Leveraging richer spatial information from 3D point clouds in conjunction with visual data can help mitigate challenges related to occlusions and object heights. This approach can enhance the algorithm's ability to infer traversability accurately.
- Larger and More Diverse Datasets: Expanding the dataset to include a wider variety of indoor environments, encompassing various lighting conditions, layouts, and scenarios, can enhance the algorithm's generalizability. A more diverse dataset will better prepare the algorithm for real-world indoor navigation challenges.
- Real-Time Feedback and Learning: Implementing mechanisms for real-time feedback and learning can enable the algorithm to continuously adapt and im-

prove its traversability estimation prediction based on the robot's interactions with the environment.

- Multi-Robot Collaboration: Exploring how multiple robots can collaborate to share information and collectively estimate traversability in complex indoor environments. This collaborative approach might also include aerial information from drones using various modalities such as infrared information.

8.2 Published Implementations

All published implementations described in Chapters 3-7 are publicly available, along with the datasets used, and can be accessed at the following github repository https://github.com/ChristosSev.

REFERENCES

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[2] H. Seraji, "Traversability index: A new concept for planetary rovers," in *Proceedings of the 1999 IEEE International Conference on Robotics and Automation (ICRA)*, 1999.

[3] G. Ishigami, K. Nagatani, and K. Yoshida, "Path planning and evaluation for planetary rovers based on dynamic mobility index," in *Proceedings of 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[4] P. Papadakis, "Terrain traversability analysis methods for unmanned ground vehicles: A survey," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 4, pp. 1373–1385, 2013.

[5] D. Langer, J. Rosenblatt, and M. Hebert, "A behavior-based system for off-road navigation," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 6, 1994.

[6] I. Kostavelis and A. Gasteratos, "Semantic mapping for mobile robotics tasks: A survey," *Robotics and Autonomous Systems*, vol. 66, pp. 86–103, 2015.

[7] C. Sevastopoulos and S. Konstantopoulos, "A survey of traversability estimation for mobile robots," *IEEE Access*, vol. 10, pp. 96 331–96 347, 2022.

[8] Y. Pan, X. Xu, Y. Wang, X. Ding, and R. Xiong, "Gpu accelerated real-time traversability mapping." in *Proceedings of 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2019.

[9] J. Sock, J. Kim, J. Min, and K. Kwak, "Probabilistic traversability map generation using 3d-lidar and camera," in *Proceedings of 2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[10] L. Wellhausen, R. Ranftl, and M. Hutter, "Safe robot navigation via multimodal anomaly detection," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1326–1333, 2020.

[11] D. D. Fan, K. Otsu, Y. Kubo, A. Dixit, J. Burdick, and A.-A. Agha-Mohammadi, "STEP: stochastic traversability evaluation and planning for risk-aware off-road navigation," in *Proceedings of Robotics: Science and Systems XVII (RSS 2021)*, 2021. [Online]. Available: arXiv:2103.02828[cs.RO]

[12] S. Li, R. Song, Y. Zheng, H. Zhao, and Y. Li, "Rugged-terrain traversability analyzing for quadruped robots," in *Proceedings of the 2nd International Conference of Intelligent Robotic and Control Engineering (IRCE)*, 2019.

[13] D. Kim, D. Carballo, J. Di Carlo, B. Katz, G. Bledt, B. Lim, and S. Kim, "Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot," in *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[14] V.-G. Loc, I. M. Koo, D. T. Tran, S. Park, H. Moon, and H. R. Choi, "Improving traversability of quadruped walking robots using body movement in 3D rough terrains," *Robotics and Autonomous Systems*, vol. 59, no. 12, pp. 1036–1048, 2011.

[15] L. Xie, S. Wang, A. Markham, and N. Trigoni, "Towards monocular vision based obstacle avoidance through deep reinforcement learning," in *Proceedings*

of Robotics: Science and Systems Workshop 2017: New Frontiers for Deep Learning in Robotics, Boston, July 2017, 2017. [Online]. Available: arXiv:1706.09829[cs.RO]

[16] S. Martin, L. Murphy, and P. Corke, "Building large scale traversability maps using vehicle experience," in *Experimental Robotics*, ser. Springer Tracts in Advanced Robotics. Springer, 2013, vol. 88.

[17] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[18] A. Huertas, L. Matthies, and A. Rankin, "Stereo-based tree traversability analysis for autonomous off-road navigation," in *Proceedings of 2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION '05)*, 2005.

[19] A. Rankin, A. Huertas, and L. Matthies, "Evaluation of stereo vision obstacle detection algorithms for off-road autonomous navigation," JPL, Tech. Rep., June 2005, http://hdl.handle.net/2014/38362.

[20] M. Bajracharya, J. Ma, M. Malchano, A. Perkins, A. A. Rizzi, and L. Matthies, "High fidelity day/night stereo mapping with vegetation and negative obstacle detection for vision-in-the-loop walking," in *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[21] C. Castejon, B. L. Boada, D. Blanco, and L. Moreno, "Traversable region modeling for outdoor navigation," *Journal of Intelligent and Robotic Systems*, vol. 43, no. 2, pp. 175–216, 2005.

[22] I. Bogoslavskyi, O. Vysotska, J. Serafin, G. Grisetti, and C. Stachniss, "Efficient traversability analysis for mobile robots using the Kinect sensor," in *Proceedings of the 2013 European Conference on Mobile Robots*, 2013.

[23] C. S. Dima, N. Vandapel, and M. Hebert, "Classifier fusion for outdoor obstacle detection," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

[24] D. A. Pomerleau, "Progress in neural network-based vision for autonomous robot driving," in *Proceedings of the Intelligent Vehicles '92 Symposium*, 1992.

[25] K. Ho, T. Peynot, and S. Sukkarieh, "Traversability estimation for a planetary rover via experimental kernel learning in a gaussian process framework," in *Proceedings of 2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[26] R. Oliveira, L. Ott, V. Guizilini, and F. Ramos, "Bayesian optimisation for safe navigation under localisation uncertainty," *Robotics Research*, 2020.

[27] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[28] M. Bellone, G. Reina, L. Caltagirone, and M. Wahde, "Learning traversability from point clouds in challenging scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 296–305, 2017.

[29] S. Zhou, J. Xi, M. W. McDaniel, T. Nishihata, P. Salesses, and K. Iagnemma, "Self-supervised learning to visually detect terrain surfaces for autonomous robots operating in forested terrain." *Journal of Field Robotics*, vol. 29, no. 2, pp. 277–297, 2012.

[30] F. Y. Narvaez, E. Gregorio, E. A., J. R. Rosell-Polo, M. Torres-Torriti, and F. A. Cheein, "Terrain classification using tof sensors for the enhancement of agricultural machinery traversability," *Journal of Terramechanics*, vol. 76, pp. 1–13, 2018.

[31] N. Kingry, M. Jung, E. Derse, and R. Dai, "Vision-based terrain classification and solar irradiance mapping for solar-powered robotics," in *Proceedings of 2018*

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018.

[32] D. Kim, J. Sun, S. M. Oh, J. M. Rehg, and A. F. Bobick, "Traversability classification using unsupervised on-line visual learning for outdoor robot navigation," in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

[33] M. Happold, M. Ollis, and N. Johnson, "Enhancing supervised terrain classification with predictive unsupervised learning," in *Proceedings of Robotics: Science and Systems VI (RSS 2006)*, 2006.

[34] M. Ollis, W. H. Huang, and M. Happold, "A bayesian approach to imitation learning for robot navigation." in *Proceedings of 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[35] B. Suger, B. Steder, and W. Burgard, "Traversability analysis for mobile robots in outdoor environments: A semi-supervised learning approach based on 3d-lidar data," in *Proceedings of 2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[36] J. Sock, J. Kim, J. Min, and K. Kwak, "Probabilistic traversability map generation using 3d-lidar and camera," in *Proceedings of 2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[37] G. Reina and A. Milella, "Towards autonomous agriculture: Automatic ground detection using trinocular stereovision," *Sensors*, vol. 12, no. 9, 2012.

[38] D. Kim, S. M. Oh, and J. M. Rehg, "Traversability classification for UGV navigation: A comparison of patch and superpixel representations," in *Proceedings of 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.

[39] Y. Tang, J. Cai, M. Chen, X. Yan, and Y. Xie, "An autonomous exploration algorithm using environment-robot interacted traversability analysis," in *Proceedings of 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2019)*, 2019.

[40] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3019–3026, 2018.

[41] D. Droeschel, M. Schwarz, and S. Behnke, "Continuous mapping and localization for autonomous navigation in rough terrain using a 3D laser scanner," *Robotics and Autonomous Systems*, vol. 88, pp. 104–115, 2017.

[42] C. A. Brooks and K. D. Iagnemma, "Self-supervised classification for planetary rover terrain sensing," in *Proceedings of the 2007 IEEE Aerospace Conference*, 2007.

[43] R. O. Chavez-Garcia, J. Guzzi, L. M. Gambardella, and A. Giusti, "Learning ground traversability from simulations," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1695–1702, 2018.

[44] W. Zhang, Q. Chen, W. Zhang, and X. He, "Long-range terrain perception using convolutional neural networks." *Neurocomputing*, vol. 275, pp. 781–787, 2018.

[45] B. Rothrock, R. Kennedy, C. Cunningham, J. Papon, M. Heverly, and M. Ono, "Spoc: Deep learning-based terrain classification for mars rover missions," in *Proceedings of AIAA SPACE 2016*, 2016.

[46] L. Tai, S. Li, and M. Liu, "Autonomous exploration of mobile robots through deep neural networks," *International Journal of Advanced Robotic Systems*, vol. 14, no. 4, 2017.

[47] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots." in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[48] S. Palazzo, D. C. Guastella, L. Cantelli, P. Spadaro, F. Rundo, G. Muscato, D. Giordano, and C. Spampinato, "Domain adaptation for outdoor robot traversability estimation from rgb data with safety-preserving loss," in *Proceedings of 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020. [Online]. Available: arXiv:2009.07565

[49] R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak, "Planning to explore via self-supervised world models," in *Proceedings of the 2020 International Conference on Machine Learning (ICML)*, 2020.

[50] R. Hadsell, P. Sermanet, J. Ben, A. Erkan, M. Scoffier, K. Kavukcuoglu, U. Muller, and Y. LeCun, "Learning long-range vision for autonomous off-road driving," *Journal of Field Robotics*, vol. 26, no. 2, pp. 120–144, 2009.

[51] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots." in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[52] L. Wellhausen, R. Ranftl, and M. Hutter, "Safe robot navigation via multimodal anomaly detection," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1326–1333, 2020.

[53] A. Valada, J. Vertens, A. Dhall, and W. Burgard, "Adapnet: Adaptive semantic segmentation in adverse environmental conditions," in *Proceedings of 2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

[55] D. Shah, B. Eysenbach, G. Kahn, N. Rhinehart, and S. Levine, "Ving: Learning open-world navigation with visual goals," in *Proceedings of 2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, 2015. [Online]. Available: arXiv:1412.6980

[57] K. S. Sikand, S. Rabiee, A. Uccello, X. Xiao, G. Warnell, and J. Biswas, "Visual representation learning for preference-aware path planning," in *Proceedings of the 2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022. [Online]. Available: arXiv:2109.08968

[58] J. Zurn, W. Burgard, and A. Valada, "Self-supervised visual terrain¨ classification from unsupervised acoustic feature learning," *IEEE Transactions on Robotics*, 2020.

[59] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning." in *Proceedings of the 2016 International Conference on Machine Learning (ICML)*, 2016.

[60] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, and V. e. a. Tsounis, "ANYmal: A highly mobile and dynamic quadrupedal robot," in *Proceedings of 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 2016. [Online]. Available: doi:10.1109/IROS.2016.7758092

[61] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation." *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, 2017.

[62] M. Shneier, T. Chang, T. Hong, W. Shackleford, R. Bostelman, and J. S. Albus, "Learning traversability models for autonomous mobile vehicles," *Autonomous Robots*, vol. 24, no. 1, pp. 69–86, 2008.

[63] N. Hirose, A. Sadeghian, P. Goebel, and S. Savarese, "To go or not to go? a near unsupervised learning approach for robot navigation," *arXiv:1709.05439*, 2017.

[64] N. Hirose, A. Sadeghian, M. Vázquez, P. Goebel, and S. Savarese, "GONet: A semi-supervised deep learning approach for traversability estimation," in *Proceedings of 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[65] N. Hirose, A. Sadeghian, F. Xia, R. Martín-Martín, and S. Savarese, "VUNet: Dynamic scene view synthesis for traversability estimation using an RGB camera," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2062–2069, 2019.

[66] H. Thanh-Tung and T. Tran, "Catastrophic forgetting and mode collapse in GANs," in *Proceedings of 2020 International Joint Conference on Neural Networks (IJCNN)*, 2020.

[67] D. Saxena and J. Cao, "Generative adversarial networks (gans) challenges, solutions, and future directions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–42, 2021.

[68] E. Goh, J. Chen, and B. Wilson, "Mars terrain segmentation with less labels," *arXiv:2202.00791 [cs.CV]*, 2022.

[69] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 2020 International Conference on Machine Learning (ICML)*, 2020.

[70] B. Gao, S. Hu, X. Zhao, and H. Zhao, "Fine-grained off-road semantic segmentation and mapping via contrastive learning," in *Proceedings of 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[71] D. Shah and S. Levine, "ViKiNG: Vision-based kilometer-scale navigation with geographic hints," *arXiv:2202.11271*, 2022.

[72] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv:1807.03748 [cs.LG]*, 2018.

[73] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2012.

[74] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[75] A. S. Polydoros and L. Nalpantidis, "Survey of model-based reinforcement learning: Applications on robotics," *Journal of Intelligent & Robotic Systems*, vol. 86, no. 2, pp. 153–173, 2017.

[76] L. Tai and M. Liu, "Towards cognitive exploration through deep reinforcement learning for mobile robots," *arXiv:1610.01733*, 2016.

[77] L. Xie, S. Wang, A. Markham, and N. Trigoni, "Towards monocular vision based obstacle avoidance through deep reinforcement learning," in *Proceedings of Robotics: Science and Systems Workshop 2017: New Frontiers for Deep Learning in Robotics, Boston, July 2017*, 2017. [Online]. Available: arXiv:1706.09829[cs.RO]

[78] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," in *Proceedings of NIPS Deep Learning Workshop 2013*, 2013. [Online]. Available: arXiv:1312.5602

[79] K. Zhang, F. Niroui, M. Ficocelli, and G. Nejat, "Robot navigation of environments with unknown rough terrain using deep reinforcement learning," in *Proceedings of the 2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2018.

[80] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proceedings of International Conference on Machine Learning (ICML 2016)*, 2016.

[81] G. Kahn, P. Abbeel, and S. Levine, "BADGR: An autonomous self-supervised learning-based navigation system," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1312–1319, 2021.

[82] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, 2020.

[83] M. Wulfmeier, P. Ondruska, and I. Posner, "Maximum entropy deep inverse reinforcement learning," *arXiv:1507.04888 [cs.LG]*, 2016.

[84] Z. Zhu, N. Li, R. Sun, D. Xu, and H. Zhao, "Off-road autonomous vehicles traversability analysis and trajectory planning based on deep inverse reinforcement learning," in *Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020. [Online]. Available: doi:10.1109/IV47402.2020.9304721

[85] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[86] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," in *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12 299–12 310.

[87] X. Chen, C.-J. Hsieh, and B. Gong, "When vision transformers outperform resnets without pre-training or strong data augmentations," *arXiv preprint arXiv:2106.01548*, 2021.

[88] C.-F. R. Chen, Q. Fan, and R. Panda, "Crossvit: Cross-attention multi-scale vision transformer for image classification," in *Proceedings of the IEEE/CVF Int'l conference on computer vision*, 2021, pp. 357–366.

[89] S. Bhojanapalli, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner, and A. Veit, "Understanding robustness of transformers for image classification," in *Proceedings of the IEEE/CVF Int'l Conference on Computer Vision*, 2021, pp. 10 231–10 241.

[90] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[91] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 299–12 310.

[92] L. Wei, L. Xie, W. Zhou, H. Li, and Q. Tian, "Mvp: Multimodality-guided visual pre-training," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*. Springer, 2022, pp. 337–353.

[93] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transform-

ers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 077–12 090, 2021.

[94] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 7262–7272.

[95] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 077–12 090, 2021.

[96] Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu, and H. Li, "Uformer: A general u-shaped transformer for image restoration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 17 683–17 693.

[97] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[98] Y. Zhang, H. Liu, and Q. Hu, "Transfuse: Fusing transformers and cnns for medical image segmentation," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24*. Springer, 2021, pp. 14–24.

[99] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[100] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

[101] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov, "Multimodal transformer for unaligned multimodal language sequences," in *Proceedings of the conference. Association for Computational Linguistics. Meeting*, vol. 2019. NIH Public Access, 2019, p. 6558.

[102] P. Cerda-Mardini, V. Araujo, and A. Soto, "Translating natural language instructions for behavioral robot navigation with a multi-head attention mechanism," *arXiv preprint arXiv:2006.00697*, 2020.

[103] C. Yu, X. Yang, J. Gao, J. Chen, Y. Li, J. Liu, Y. Xiang, R. Huang, H. Yang, Y. Wu, *et al.*, "Asynchronous multi-agent reinforcement learning for efficient real-time multi-robot cooperative exploration," *arXiv preprint arXiv:2301.03398*, 2023.

[104] S. Yi, X. Liu, J. Li, and L. Chen, "Uavformer: A composite transformer network for urban scene segmentation of uav images," *Pattern Recognition*, vol. 133, p. 109019, 2023.

[105] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.

[106] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

[107] Q. She, F. Feng, X. Hao, Q. Yang, C. Lan, V. Lomonaco, X. Shi, Z. Wang, Y. Guo, Y. Zhang, *et al.*, "Openloris-object: A robotic vision dataset and

benchmark for lifelong deep learning," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 4767–4773.

[108] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[109] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 3, no. 3, pp. 1–13, 2007.

[110] M. Assran, M. Caron, I. Misra, P. Bojanowski, F. Bordes, P. Vincent, A. Joulin, M. Rabbat, and N. Ballas, "Masked siamese networks for label-efficient learning," in *European Conference on Computer Vision*. Springer, 2022, pp. 456–473.

[111] J. Bekker and J. Davis, "Learning from positive and unlabeled data: A survey," *Machine Learning*, vol. 109, pp. 719–760, 2020.

[112] P. Michel, O. Levy, and G. Neubig, "Are sixteen heads really better than one?" *Advances in neural information processing systems*, vol. 32, 2019.

[113] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3828–3838.

[114] D. Weikersdorfer, A. Schick, and D. Cremers, "Depth-adaptive supervoxels for rgb-d video segmentation," in *2013 IEEE International Conference on Image Processing*. IEEE, 2013, pp. 2708–2712.

[115] S. Tsutsui, T. Kerola, S. Saito, and D. J. Crandall, "Minimizing supervision for free-space segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 988–997.

[116] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 179–12 188.

BIOGRAPHICAL STATEMENT

Christos Sevastopoulos was born in Athens, Greece, in 1987. He received his B.S. degree in Physics from the National and Kapodistrian University of Athens, Greece, in 2015, his M.S degree in Robotics from the University of Bristol, United Kingdom, in 2017 and and Ph.D. degree in Computer Engineering from The University of Texas at Arlington in 2023. From 2018 to 2019, he was with the National Centre for Scientific Research (NCSR) 'Demokritos , as a research trainee. He was co-supervised by NSCR during his Ph.D. and his research interests are in the area of Computer Vision for estimating traversable scenes for mobile robots.