UNIVERSITY OF CALIFORNIA SAN DIEGO

Hybrid semantic models for building smart and robust home robots

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Computer Science

by

Anwesan Pal

Committee in charge:

Professor Henrik I. Christensen, Chair
Professor Manmohan Chandraker
Professor Sicun Gao
Professor Jana Kosecka
Professor Nuno Vasconcelos

2023

The Dissertation of Anwesan Pal is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2023

DEDICATION

To my parents: Dipankar Pal and Lopamudra Pal

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I wish to extend my sincerest gratitude to my advisor, Prof. Henrik I Christensen for his help and guidance towards the work contained in this dissertation. I have learned a tremendous amount through the discussions that we have had together. This dissertation would not have been what it is without him.

I would also like to thank my dissertation committee members, Prof. Manmohan Chandraker, Prof. Sicun Gao, Prof. Jana Kosecka, and Prof. Nuno Vasconcelos for their invaluable insights and constant support throughout my doctoral journey. I want to especially thank Prof. Nikolay Atanasov for supporting me during the initial days of my graduate career while I was trying to decide on a research topic to pursue towards a doctoral degree.

I am also indebted to my fellow Cognitive Laboratory lab members, Dr. Carlos Nieto-Granda, Dr. Vikas Dhiman, Dr. Shengye Wang, Dr. Ruffin White, Dr. Priyam Parashar, Dr. Quan Vuong, Dr. David Paz, Dr. Akanimoh Adeleye, Dr. Christopher Dambrosia, Yiding Qiu, Srirangan Madhavan, Jiaming Hu, Shrutheesh R. Iyer, Abdulaziz Almuzairee, Seth Farrell, Andrea Frank, Jing-yan Liao, Rohan Patil, Julian Raheema, Luobin Wang, Zihan Zhang, Hengyuan Zhang, Narayanan Elavathur Ranganatha, Qinru Li, Shixin Li and James Smith for their helpful comments regarding my work and support over the years.

Outside of the lab, a special thank you goes to Prof. Ahmed Qureshi, Yash Agarwal, Dr. Rohan Pote, Sukanya Salunke, Aditya Sant, Dr. Govind Gopal, Dr. Nadim Ghaddar, Dr. Pranav Suresh, Dr. Sheel Nidhan, Mandar Pradhan, Unnikrishnan Sivaprasad, Ashin George and Shahar for their close friendship throughout my doctoral journey. Spending time with you all always made me feel at home, even though I was physically so far away from home.

In the summer of 2021, I had the opportunity to intern at the Google Android Pixel Camera team. My manager Dr. Ying-Chen Lou, and my mentor Dr. Abhishek Kar gave me immense guidance during my first experience outside academia. Following that, in the summers of 2022 and 2023, I was fortunate to intern at the Amazon Alexa AI team. Over the two years, I got to work with a great group of people – Dr. Yue Wu, Dr. Ayush Jaiswal, Dr. Xu Zhang,

Dr. Xiaofeng Gao, Dr. Qiaozi Gao, Dr. Aishwarya Padmakumar, Prof. Gaurav Sukhakme, Prateek Singhal, and Dr. Abhinav Mathur. I would like to thank them deeply for giving me the opportunity, and freedom to pursue multiple open-ended research projects, while also preparing me for a full-time career in the industry.

I would be failing in my duty if I did not show my gratitude towards all staff members of the ECE and CSE department student affairs office at UC San Diego for aiding me in navigating through the administrative requirements of my degree.

Lastly, I am eternally grateful to my parents, my elder brother, and his wife for their constant motivation and unwavering faith in me. Their love and encouragement have been critical for the successful completion of my degree.

This dissertation, in part, is a reprint of this dissertation author's publications.

Chapters 2 and 6, in part, are a reprint of **A. Pal**, C. Nieto-Granda and H. I. Christensen, "DEDUCE: Diverse scEne Detection methods in Unseen Challenging Environments", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019*. The dissertation author was the primary author of this paper.

Chapters 3 and 6, in part, are a reprint of **A. Pal**, S. Mondal and H. I. Christensen, "Looking at the right stuff - Guided semantic-gaze for autonomous driving", in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020*. The dissertation author was the primary author of this paper.

Chapters 4 and 6, in part, are a reprint of the following papers:

- **A. Pal**[*], Y. Qiu[*] and H. I. Christensen, "Learning hierarchical relationships for object-goal navigation", in *Conference on Robot Learning (CoRL), 2020*. The dissertation author was the joint-first author of this paper.

- S. Madhavan, **A. Pal** and H. I. Christensen, "Role of reward shaping in object-goal navigation", in *Embodied AI Workshop, Conference on Computer Vision and Pattern*

[*]Equal Contribution

Recognition (CVPR), 2022. The dissertation author was a co-author of this paper.

Chapters 5 and 6, in part, are a reprint of S. R. Iyer, **A. Pal**, J. Hu, A. Adeleye, A. Aggarwal and H. I. Christensen, "Household navigation and manipulation for everyday object rearrangement tasks", in *IEEE International Conference on Robotic Computing (IRC), 2023*. The dissertation author was a co-author of this paper.

VITA

2013-2017	Bachelor of Engineering in Electrical Engineering, Indian Institute of Engineering Science and Technology, Shibpur, India
2017-2018	Teaching Assistant, University of California San Diego, USA
2018-2023	Graduate Student Researcher, University of California San Diego, USA
2017-2019	Master of Science in Electrical Engineering (Intelligent Systems, Robotics and Control), University of California San Diego, USA.
2021	Research Intern, Google, USA
2022	Applied Scientist Intern, Amazon, USA
2023	Applied Scientist Intern, Amazon, USA
2019-2023	Doctor of Philosophy in Computer Science, University of California San Diego, USA

PUBLICATIONS

S. R. Iyer, **A. Pal**, J. Hu, A. Adeleye, A. Aggarwal and H. I. Christensen, "Household navigation and manipulation for everyday object rearrangement tasks", in *IEEE International Conference on Robotic Computing (IRC), 2023*

A. Pal, S. Wadhwa, A. Jain, X. Zhang, Y. Wu, R. Chada, P. Natarajan and H. I. Christensen, "FashionNTM: Multi-turn Fashion Image Retrieval via Cascaded Memory", in *IEEE/CVF International Conference on Computer Vision (ICCV), 2023*

S. Madhavan, **A. Pal** and H. I. Christensen, "Role of reward shaping in object-goal navigation", in *Embodied AI Workshop, Conference on Computer Vision and Pattern Recognition (CVPR), 2022*

A. Pal[*], Y. Qiu[*] and H. I. Christensen, "Learning hierarchical relationships for object-goal navigation", in *Conference on Robot Learning (CoRL), 2020*

A. Pal, S. Mondal and H. I. Christensen, "Looking at the right stuff - Guided semantic-gaze for autonomous driving", in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020*

A. Pal, C. Nieto-Granda and H. I. Christensen, "DEDUCE: Diverse scEne Detection meth-

[*]Equal Contribution

ods in Unseen Challenging Environments", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019*

ABSTRACT OF THE DISSERTATION

Hybrid semantic models for building smart and robust home robots

by

Anwesan Pal

Doctor of Philosophy in Computer Science

University of California San Diego, 2023

Professor Henrik I. Christensen, Chair

The creation of home robots that can aid human beings in daily mundane chores has been a long-standing goal of robotics research. Some common indoor tasks that service robots can help with include retrieving objects from different locations, observing and monitoring home environments, and rearranging household objects. Enabling artificial agents to perform such activities requires knowledge gathered from several broad topics in Embodied Artificial Intelligence (EAI), such as localization and mapping, contextual scene understanding, and efficient interaction strategies in realistic environments. A common approach adopted by existing methods is to create dense geometric representations of the environment, typically in the form of point-cloud reconstruction of indoor regions. However, there are two fundamental problems

associated with such metric maps, (i) they are non-trivial to construct and often require constant updates as the surrounding world evolves over time, and (ii), they lack semantic information, thereby making data association and contextual indexing more challenging. As such, planning algorithms that utilize only these representations rarely generalize to complex tasks such as searching for objects in noisy real-world environments. The primary objective for this work has been to develop appropriate semantic models of the world to enable robots to make smart, and robust decisions while solving complex indoor tasks. The first component here is a hierarchical semantic representation of the world. Different levels in this structure can correspond to a variation in granularity of scene understanding - ranging from metric information to reasoning-based context, and topological layout. Given this hierarchical representation, the next step is to formulate a smart planning strategy that can adaptively extract only the necessary context for a particular object-interaction task. The combination of this semantic representation and planning module results in a hybrid semantic model that is inspired by human-level cognitive models, and their ability to generalize across domains. Several methods to estimate contextual models for inferring scene geometry and semantics have been presented in this dissertation, with applications in visual place categorization, object-goal visual navigation, and complete home robot rearrangement tasks. Finally, some of the existing challenges in this domain are mentioned, along with a few future directions for home robotics research.

Chapter 1

Introduction

Embodied Artificial Intelligence, popularly known as Embodied AI or EAI, is a study of artificial systems that demonstrate intelligent behavior by directly interacting with human-centric environments [30]. Within the scientific community, this is often associated with service robotics research. The roots of service robotics research can be traced back to the longstanding human fascination with the integration of machines into the society, as depicted in science-fiction movies. Science fiction has played a crucial role in shaping the collective imagination about the possibilities and implications of robotics. Iconic films such as "Blade Runner," "I, Robot," and "The Jetsons" have not only entertained audiences but have also served as a source of inspiration for researchers and engineers in the development of real-world robotic systems. Even though the complete integration of these complex systems into the daily lives of human beings is still a far-fetched dream of robotics research, significant progress has been made toward that goal in the last decade or so. This dissertation aims to take a step in that direction by discussing ways to combine time-tested classical techniques with recently developed robust learning-based methods, into a hybrid semantic framework for performing home robot tasks.

Figure 1.1 illustrates a futuristic Embodied AI problem setting, where a human being enters a messy kitchen with two robots inside, and enquires about the presence of cereals. To answer such a query, the agent needs to divide and conquer by thinking about two sub-problems – (i) Where to look for the cereal object in this environment? and (ii) Upon detecting the object,

Figure 1.1. Illustration of a classic embodied AI problem. Source [30].

how to best interact with it? The answer to the first question involves performing navigation *with reasoning*. This implies that in addition to simply determining how to move, the robot also needs to figure out what are the likely places to look for in search of an object of interest, and in what order to visit them. The second question mainly falls in the domain of robotic manipulation, which is about how to best grasp an object for picking it up for transportation – popularly known as pick-and-place tasks in the robotics community. This dissertation will primarily focus on the topic of robot navigation using reasoning, while the manipulation component will be used as a tool for interacting with objects in the environment.

Historically speaking, the focus of Embodied AI research was to have robots, fitted with a suite of sensors and armed with end-effectors, work in noisy real-world environments [126, 17]. This contrasts the assumption of clean inputs in a static world, as required by classical artificial intelligence (AI) approaches [167]. However, these methods were mostly limited to robot locomotion in controlled environments. With rapid advancements in algorithms following the deep learning revolution, coupled with the emergence of large-scale datasets and cheaper cost of computational resources, the field of Embodied AI has experienced unprecedented growth within the last decade or so. Empowered by the creation of realistic simulation frameworks

[128, 141, 142, 152, 10, 86], typically derived by scanning models of existing buildings, it has been made possible to tackle more complicated tasks such as goal-oriented navigation, complex manipulation, and human-robot interaction. Given such an environment, a popular approach that has been adopted [188, 171, 5, 34, 14, 15] is to have an EAI agent learn all the steps required for a task, simply by training to map the initial input state to the final output result. This technique, commonly referred to as end-to-end learning, eliminates the need for hand-crafted heuristic features that were inherently part of old classical methods. Despite largely reducing human efforts in designing complex fundamental systems, these learned models are not without problems either. The most fundamental issue arises due to the lack of interpretability in these algorithms, thereby making it harder to predict their outputs. Due to this, purely learning-based methods often have a bias towards their trained setting, and therefore suffer when deployed in a previously unseen environment.

Given the drawbacks of both these types of *homogeneous* systems, as in purely classical, or purely end-to-end learning based, this dissertation argues the need for *hybrid* semantic models for building smart and robust home robots. The key idea with hybrid models is to adaptively switch between classical, and learning-based techniques depending upon the task, to leverage the best of both worlds. For instance, classical techniques can be utilized for generating low-level geometric maps, as they are fast to generate. In contrast, learning-based models have been shown to have a better understanding of semantic context, as they are primarily trained to recognize patterns in data. And finally, reasoning using common-sense knowledge can help to bridge the

Table 1.1. Benefits of using hybrid models over homogeneous models

Homogeneous models	Hybrid models
1. Lack semantic context	1. Learn to recognize task-agnostic context [117, 119]
2. Extensive manual labor needed	2. Primarily utilize online techniques [118, 119]
3. High sample complexity	3. Expand from common-sense priors [119, 101]
4. Memorize the trained settings	4. Transferable across domains [118, 62]

gap between the two. Table 1.1 summarizes some of the advantages of using hybrid models over homogeneous models.

The main objective of the work conducted in this dissertation is to develop appropriate semantic models of the world to enable robots to make smart, and robust decisions while solving complex indoor tasks. The focus is to combine learned semantic representations with classical planning modules to develop a hybrid semantic model that is inspired by human-level cognitive models, and their ability to generalize across domains. Several works have been discussed for estimating contextual models for inferring scene geometry and semantics, with applications in visual place categorization, autonomous driving, object-goal visual navigation, and complete home robot rearrangement tasks. This dissertation follows a bottom-up approach:

- Chapter 2 introduces several methods for detecting diverse scenes in challenging environments, through an integration of both object and scene information from the surrounding.
- Chapter 3 discusses context recognition in autonomous driving scenarios through a novel amalgamation of the human gaze and semantics of common roadside objects.
- Chapter 4 grounds the generic robot navigation task into an object-goal navigation problem, and introduces smart search strategies for solving it.
- Chapter 5 brings all the above components together into an integrated hybrid semantic model for home robot rearrangement tasks.
- Chapter 6 summarizes the dissertation, and describes some of the challenges faced by the dissertation author during the doctoral journey. Finally, some recommendations for future work in Embodied AI are mentioned.

1.1 Visual Place Categorization of indoor scenes

In robotics, visual place categorization is defined as the problem of predicting the semantic category of a place based on image measurements acquired from an autonomous

platform [173]. For instance, a scene comprising of a refrigerator, stove, and microwave is likely to be a kitchen, while another scene with a bed, nightstand, and alarm clock is most likely a bedroom. In recent years, there has been a rapid increase in the number of service robots deployed to aid people in their daily activities. Unfortunately, most of these robots require human input for training to do tasks in indoor environments. Successful domestic navigation often requires access to semantic information about the environment, which can be learned without human guidance.

In Chapter 2 of this dissertation, several algorithms, together called Diverse scEne Detection methods in Unseen Challenging Environments, or DEDUCE, are proposed to tackle the challenge of visual place categorization. These methods incorporate deep fusion models derived from scene recognition systems and object detectors. The five methods described here have been evaluated on several popular recent image datasets, as well as real-world videos acquired through multiple mobile platforms including a real robot system. The final results show an improvement over the existing state-of-the-art approaches for visual place recognition. Supplementary material including code and the videos of the different experiments are available at https://sites.google.com/eng.ucsd.edu/deduce.

1.2 Context recognition in autonomous driving

As learning-based algorithms become more and more prevalent within the research community, it becomes necessary to interpret how the solutions found by these models can be better understood by humans. For indoor environments, this amounts to identifying characteristic objects that define a particular scene. However, for outdoor autonomous driving scenes, this is more complicated due to two reasons – (i) There might be multiple overlapping scenes, as a large part of the surrounding falls within the field-of-view, and (ii) All the context may not be relevant to the current driving task. For instance, when driving a car, only a narrow region of the scene in front of the vehicle determines the next actions to be taken. In this scenario, estimating

an expert driver's gaze is a common way to identify important regions around the vehicle.

Chapter 3 describes an approach for identifying crucial regions that require attention during the task of driving. In recent years, predicting drivers' focus of attention has been a very active area of research in the autonomous driving community. Unfortunately, existing state-of-the-art techniques achieve this by relying only on human gaze information, thereby ignoring scene semantics. In this dissertation, a novel Semantics Augmented GazE (**SAGE**) detection approach has been proposed that captures driving specific contextual information, in addition to the raw gaze. Such a combined attention mechanism serves as a powerful tool to focus on the relevant regions in an image frame to make driving both safe and efficient. Using this, a complete saliency prediction framework - **SAGE-Net** is designed, which modifies the initial prediction from SAGE by taking into account vital aspects such as distance to objects (depth), ego vehicle speed, and pedestrian crossing intent. Exhaustive experiments conducted through four popular saliency algorithms show that on **49/56 (87.5%)** cases - considering both the overall dataset and crucial driving scenarios, SAGE outperforms existing techniques without any additional computational overhead during the training process. Additional information is available as part of the supplementary material at https://sites.google.com/eng.ucsd.edu/sage-net.

1.3 Object-goal navigation utilizing hierarchical relationship

Object-goal navigation is a specific instance of the general robot navigation task. The main objective here is to navigate through the environment in search of an object of a specific category, drawn from a predefined set, primarily using visual features. This is in contrast to two other types of goal-driven robot navigation tasks [7] – (i) Point-goal navigation, where the objective is to navigate to a specific location in the environment, typically represented in the form of a 2D cartesian coordinate of a point, and (ii) Area-goal navigation, where the EAI agent needs to navigate to an area of a specified category. For example, "kitchen", "garage", or

"foyer". This task also relies on prior knowledge about the appearance and layout of different areas. While navigating to a point, or broad area of the environment are themselves interesting research problems in Embodied AI, object-goal navigation particularly has enormous real-world applications as humans typically look for and reason about their surroundings using object information.

Chapter 4 describes the object-goal navigation task with a special focus on developing smart search strategies identifying target objects. Direct search for objects as part of navigation poses a challenge for small items. Utilizing context in the form of object-object relationships enables hierarchical search for targets efficiently. Most of the current methods tend to directly incorporate sensory input into a reward-based learning approach, without learning about object relationships in the natural environment and thus generalize poorly across domains. In this dissertation, Memory-utilized Joint hierarchical Object Learning for Navigation in Indoor Rooms (MJOLNIR), a target-driven navigation algorithm is presented, which considers the inherent relationship between target objects, and the more salient contextual objects occurring in their surroundings. Extensive experiments conducted across multiple environment settings show an 82.9% and 93.5% gain over existing state-of-the-art navigation methods in terms of the success rate (SR), and success weighted by path length (SPL), respectively. It is also shown that the proposed model learns to converge much faster than other algorithms, without suffering from the well-known overfitting problem. Additional details regarding the supplementary material and code are available at https://sites.google.com/eng.ucsd.edu/mjolnir.

1.4 Complete home robot rearrangement task

Enabling artificial agents to efficiently interact with the environment and perform day-to-day tasks has been a longstanding goal of Embodied AI [9, 165]. In recent years, navigation and instruction following tasks have received a lot of attention within the research community. Such tasks constitute the building blocks of interactive embodied agents. While remarkable progress

in the development of algorithms has been observed in recent years, a typical assumption of this task is that of a static environment. This means that even though the EAI agents can move within the environment in search of objects, they cannot interact and/or change the state of those objects. This limits the scope of the developed algorithms to transfer to real-world room rearrangement tasks, which are often dynamic, and necessitate moving objects from one place to another.

Chapter 5 considers the problem of building an assistive robotic system that can help humans in daily household cleanup tasks. Creating such an autonomous system in real-world environments is inherently quite challenging, as a general solution may not suit the preferences of a particular customer. Moreover, such a system consists of multi-objective tasks comprising – (i) Detection of misplaced objects and prediction of their potentially correct placements, (ii) Fine-grained manipulation for stable object grasping, and (iii) Room-to-room navigation for transferring objects in unseen environments. The work described in this dissertation systematically tackles each component and integrates them into a complete object rearrangement pipeline. To validate the proposed system, multiple experiments are conducted on a real robotic platform involving multi-room object transfer, user preference-based placement, and complex pick-and-place tasks. Additional details including video demonstrations of the work are available at https://sites.google.com/eng.ucsd.edu/home-robot.

1.5 Acknowledgements

Chapter 1, in part, is a reprint of the following papers:

- **A. Pal**, C. Nieto-Granda and H. I. Christensen, "DEDUCE: Diverse scEne Detection methods in Unseen Challenging Environments", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019*. The dissertation author was the primary author of this paper.

- **A. Pal**, S. Mondal and H. I. Christensen, "Looking at the right stuff - Guided semantic-gaze for autonomous driving", in *IEEE/CVF Conference on Computer Vision and Pattern*

Recognition (CVPR), 2020. The dissertation author was the primary author of this paper.

- **A. Pal**[1], Y. Qiu[1] and H. I. Christensen, "Learning hierarchical relationships for object-goal navigation", in *Conference on Robot Learning (CoRL), 2020*. The dissertation author was the joint-primary author of this paper.

- S. Madhavan, **A. Pal** and H. I. Christensen, "Role of reward shaping in object-goal navigation", in *Embodied AI Workshop, Conference on Computer Vision and Pattern Recognition (CVPR), 2022*. The dissertation author was a co-author of this paper.

- S. R. Iyer, **A. Pal**, J. Hu, A. Adeleye, A. Aggarwal and H. I. Christensen, "Household navigation and manipulation for everyday object rearrangement tasks", in *IEEE International Conference on Robotic Computing (IRC), 2023*. The dissertation author was a co-author of this paper.

[1]Equal Contribution

Chapter 2

Visual Place Categorization of indoor scenes

2.1 Introduction

Scene recognition and understanding have been an important area of research in the Robotics and computer Vision community for more than a decade now. Programming robots to identify their surroundings is integral to building autonomous systems for aiding humans in household environments.

Kostavelis *et al*. [80] provided a survey of previous work in semantic mapping using robots in the last decade. According to their study, scene annotation augments topological maps based on human input or visual information about the environment. Bormann *et al*. [13] pointed out that the most popular approaches in room segmentation involve segmenting floor plans based on spatial regions.

An essential aspect of any spatial region is the presence of specific objects in it. Some examples include a bed in a bedroom, a stove in a kitchen, a sofa in a living room, etc. Niko *et al*. [137] formulated the following three reasoning challenges that address the semantics and geometry of a scene and the objects therein, both separately and jointly: 1) Reasoning About Object and Scene Semantics, 2) Reasoning About Object and Scene Geometry, and 3) Joint Reasoning about Semantics and Geometry. The content in this thesis focuses on the first reasoning challenge and uses Convolutional Neural Networks (CNNs) as feature extractors for

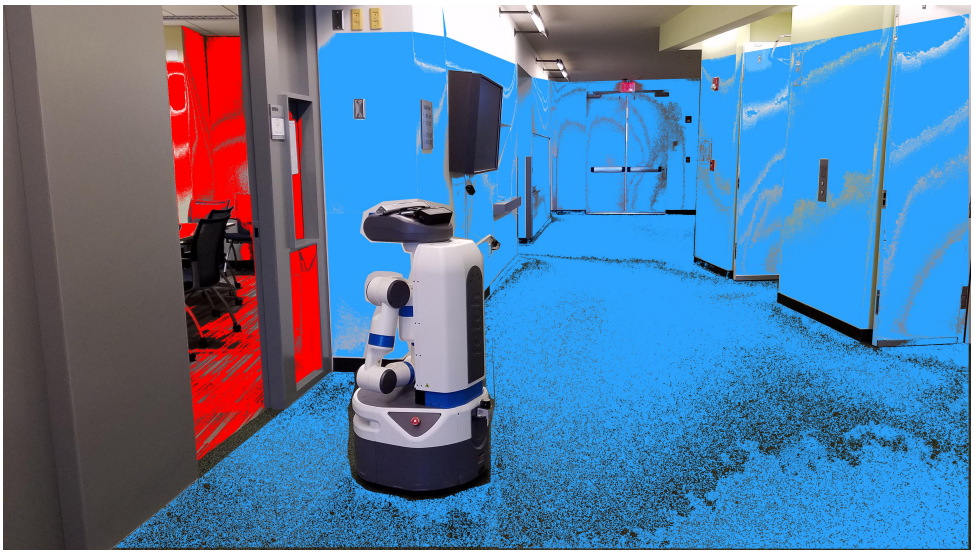


Figure 2.1. Visualization of the semantic mapping performed while the Fetch robot is navigating through the environment.

both scenes and objects. The goal is to design a system that allows a robot to identify the area where it is located using visual information in a manner similar to how a human being would.

In this chapter, five different models of scene prediction have been considered, each developed through the integration of object and scene information from a surrounding to perform place categorization. To evaluate the robustness of these models, extensive experiments have been conducted on state-of-the-art still-image datasets, real-world videos captured via different hand-held cameras, and those recorded using a mobile robot platform in multiple challenging environments. One such environment is shown in Figure 2.1, where the segmented regions correspond to the scenes detected (in this case, blue refers to "corridor", and red refers to "conference room"). The results obtained from the experiments demonstrate that the proposed system can be generalized beyond the training data, in addition to being impervious to object clutter, motion blur, and varying light conditions.

2.2 Background

Semantic place categorization using only visual features has been an important area of research for robotic applications [157, 173]. In the past, many robotics researchers focused on place recognition tasks [127, 144] or the problem of scene recognition in computer vision [116, 129].

Quattoni and Torralba [129] introduced a purely vision-based place recognition system, which improves the performance of the global gist descriptor by detecting prototypical scene regions. However, the size, shape, and location of up to ten object prototypes have to be labeled and learned in advance for this system to work. Also, the labeling task is very work-intensive, and the approach of having fixed regions is only useful in finding objects in typical views of the scene. This makes the system ill-suited for robotics applications. To deal with the flexible positions of objects, a visual attention mechanism is applied that can locate important regions in a scene automatically.

A number of different approaches have been proposed to address the problem of classifying environments. One popular approach adopted is to use Feature Matching with Simultaneous Localization and Mapping (SLAM). Ekvall *et al*. [40] and Tong *et al*. [156] demonstrated a strategy for integrating spatial and semantic knowledge in a service environment using SLAM and object detection & recognition based on Receptive Co-occurrence Histograms. Espinace *et al*. [41] presented an indoor scene recognition system based on a generative probabilistic hierarchical model using contextual relations to associate objects to scenes. Kollar *et al*. [76] utilize the notion of object-object and object-scene context to reason about the geometric structure of the environment to predict the location of the objects. The performance of the object classifiers is improved by including geometrical information obtained from a 3D range sensor that facilitates a focus of attention mechanism in addition. However, these approaches only identify the place based on the specific objects detected and the hierarchical model used to link the objects with the place. In contrast to their method, the proposed algorithm is not limited to a small number of

recognizable objects.

Recently, there have been several approaches to Scene Recognition using Neural Networks. Liao *et al*. [88, 89] used Convolutional Neural Networks (CNNs) to recognize the environment based on object occurrence for semantic reasoning, but their system information and mapping results are not provided. Sun *et al*. [149] proposed a Unified Convolutional Neural Network which performs Scene Recognition and Object Detection. Luo *et al*. [100] developed a semantic mapping framework utilizing spatial room segmentation, CNNs trained for object recognition, and a hybrid map provided by a customized service robot. Niko *et al*. [150] proposed a transferable and expandable place categorization and semantic mapping system that requires no environment-specific training. Mancini *et al*. [104] addressed Domain Generalization (DG) in the context of semantic place categorization. They also provide results of state-of-the-art algorithms on the VPC Dataset [173] that were compared with in this work. However, most of these results do not test their algorithms on a wide variety of platforms. This is the main focus of the work shown in this chapter. The experiments are conducted, both for static images, and dynamic real-world videos captured using hand-held cameras and robots.

2.3 Proposed Methodology

A set of five different models are considered, abbreviated as Diverse scEne Detection methods in Unseen Challenging Environments (DEDUCE), for place categorization. Each model is derived from two base modules, one based on the PlacesCNN [186] and the other being an Object Detector-YOLOv3 [134]. The classification model can be formulated as a supervised learning problem. Given a set of labeled training data $X^{tr} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2)...(\mathbf{x}_N, \mathbf{y}_N)\}$, where $\mathbf{x}_i$ corresponds to the data samples and $\mathbf{y}_i$ to the scene labels, the classifier should learn the discriminative probability model

$$p(\hat{\mathbf{y}}_j | \Phi(X^{tr})) \tag{2.1}$$

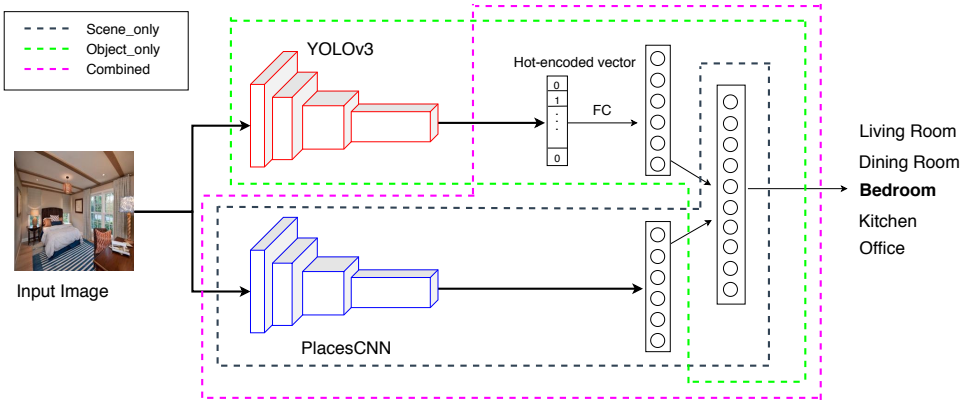


Figure 2.2. Model Architecture. The highlighted regions represent the portion of the network that was trained for the respective models.

where $\hat{\mathbf{y}}_j$ corresponds to the j-th predicted scene label and $\Phi = \{\phi_1, \phi_2 ... \phi_t\}$ are the set of different feature representations obtained from the $\mathbf{x}_i$. This trained model should be able to correctly classify a set of unlabelled test samples $X^{te} = \{\mathbf{x}_1, \mathbf{x}_2 ... \mathbf{x}_M\}$. It is to be noted that while the goal of each of the proposed five models is to perform place categorization, it is the Φ that varies across them. Now the two base modules are described, along with how the five models are derived and trained from them. The complete network architecture is given in Figure 2.2.

2.3.1 Scene Recognition

For scene recognition, the PlacesCNN model is used. The base architecture is that of Resnet-18 [57] which has been pre-trained on the ImageNet dataset [31] and then fine-tuned on the Places365 dataset [186]. Seven classes out of the total 365 classes are chosen, which are integral to the recognition of indoor home/office environments - `Bathroom`, `Bedroom`, `Corridor`, `Dining room`, `Living room`, `Kitchen`, and `Office`. The provided official training and validation split are used for this work. The training set consists of 5000 labeled images for each scene class, while the test set contains 100 images for each scene.

2.3.2 Object Detection

Object detection is a domain that has benefited immensely from the developments in deep learning. Recent years have seen people develop many algorithms for object detection, some of which include YOLO [132, 133, 134], SSD [98], Mask RCNN [56], Cascade RCNN [19] and RetinaNet [91]. In this work, the YOLOv3 [134] detector is used, mainly because of its speed, which makes real-time processing possible. It is a Fully Convolutional Network (FCN) and employs the Darknet-53 architecture which has 53 convolution layers, consisting of successive 3x3 and 1x1 convolutional layers with some shortcut connections. The network used here has been pre-trained to detect the 80 classes of the MS-COCO dataset [92].

2.3.3 Place Categorization models

Scene Only

The first model that was used consists of only the pre-trained and fine-tuned PlacesCNN with a simple Linear Classifier on top of it. This model accounts for a holistic representation of a scene, without specifically being trained to detect objects. Thus, the feature vector for this model is given by $\Phi_{scene} = \phi_s$.

Object Only

The second model acts as a Scene classifier using only the information of detected objects. There is no separate training performed here to identify the individual scene attributes. For this purpose, a codebook of the most common COCO objects seen in all seven scenes was created. This is shown in Table 2.1. It is to be noted that every object has been associated with only one scene, thereby making it a *landmark*. For this model, the feature representation is given by $\Phi_{obj} = \phi_{\{obj\}}$ where $\{obj\}$ is the set of objects detected in the image.

Table 2.1. Top landmark objects (non-human) for the seven different scene classes

Bathroom	Toilet	Sink	-	-
Bedroom	Bed	-	-	-
Corridor	-	-	-	-
Dining Room	Dining Table	Chair	Wine Glass	Bowl
Kitchen	Oven	Microwave	Refrigerator	-
Living Room	Sofa	Vase	-	-
Office	TV-Monitor	Laptop	Keyboard	Mouse

Scene+Attention

In this model, the activation maps for the given image of a scene are computed, and using those, the locations where the network has its focus during scene classification is visualized. From the output of the final block convolutional layer (layer 4) of the WideResnet architecture [183], the 14x14 feature blobs are obtained, which retain the spatial information corresponding to the whole image. The proposed model is similar to the *soft* attention mechanism of [178] in the sense that here too, the weights are assigned to be the output of a softmax layer, thereby associating a probability distribution to it. However, since the classification is not based on a sequence of images, a recurrent network is not employed to compute the sequential features. Instead, simply the weights of the final FC layer are utilized and their dot product is taken with the feature blobs to obtain the heatmap. The final step is to upsample this 14x14 heatmap to the input image size, and then overlay it on top to obtain the activation mask $m(x_n)$ of the input image x_n. Therefore, the feature representation for this model is $\Phi_{attn.} = \phi_{m(x_n)}$. The basic architecture is given in Figure 2.3.

Combined

In this model, the PlacesCNN mentioned above is used as a feature extractor to give the semantics of a scene. In addition, the YOLO detector gives information regarding the objects present in the image. Given an image of a scene, this model creates a hot-encoded vector of 80 dimensions, corresponding to the object classes of MS-COCO, with only the indices of the detected objects set to 1. Then, this vector is concatenated along with that of the output of

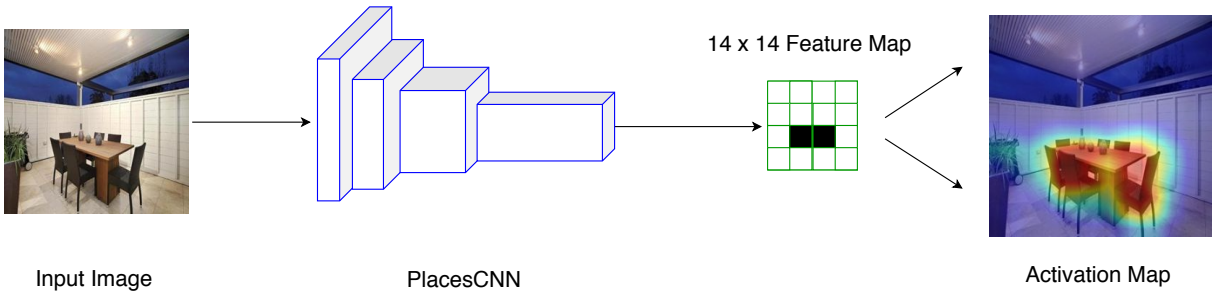


Figure 2.3. Architecture for Generation of the Activation Map. The 14x14 feature maps obtained from block layer 4 of WideResNet are combined with the weights from the final FC layer, and then their dot product is upsampled to the image size and overlaid on top to get the activation maps

the scene feature extractor and a Linear Classifier is trained on top of it. Since two different features of the scene and objects are combined here, the feature representation is given by $\Phi_{comb.} = \{\phi_s, \phi_{\{obj\}}\}$.

Scene+N-best objects

The final proposed model is similar to the above in the sense that here also, both the PlacesCNN and the YOLO detector are used. However, this model does not need to be retrained again and so, it is significantly faster. For this model, a certain confidence threshold on the scene detector is placed, and only when the probability of classification is below this threshold, the information about specific objects in the scene (as obtained from Table 2.1) are searched for. The reason for introducing this as a new model is two-fold. Firstly, the scenario of looking at every object present is eliminated since it is often redundant, given the semantics of the scene. Secondly, this is similar to how human beings operate when they come across an unknown scene. The feature representation for this model is given by $\Phi_{N-best} = \{\phi_s, \phi_{\{N-obj\}}\}$.

2.4 Experiments and Results

The proposed five models described above are evaluated on a number of platforms. First, the training procedure is described, followed by the different experiment settings used for evaluation.

Table 2.2. Accuracy in percentage of DEDUCE on Places365 dataset

Scenes	Φ_{scene}	Φ_{obj}	$\Phi_{attn.}$	$\Phi_{comb.}$	Φ_{N-best}
Dining room	79	**94**	75	79	80
Bedroom	90	74	90	90	**91**
Bathroom	**92**	65	**92**	91	**92**
Corridor	94	90	**99**	96	94
Living Room	**84**	25	68	80	**84**
Office	85	29	76	**94**	83
Kitchen	**87**	62	70	**87**	**87**
Avg	87.3	62.6	81.4	**88.1**	87.3

2.4.1 Training Procedure

As mentioned in Section 2.3, the base architecture for the scene classifier is the ResNet-18 architecture. The data pre-processing and training process is similar to [186]. A stochastic gradient descent (SGD) optimizer is used with an initial learning rate of 0.1, momentum of 0.9, and a weight decay of 10^{-4}. For the Φ_{scene} and the $\Phi_{attn.}$ models, the training was performed for 90 epochs with the learning rate being decreased by a factor of 10 every 30 epochs. The $\Phi_{comb.}$ model converged much faster and so, it was only trained for 9 epochs, with the learning rate reduced by 10 times after every 3 epochs. For all 3 training procedures, the cross-entropy loss function was optimized, which minimizes the cost function given by

$$J(\hat{\mathbf{y}}_j, \mathbf{y}_j) = -\frac{1}{N}(\sum_{j=1}^{N} \mathbf{y}_j \odot log(\hat{\mathbf{y}}_j)) \tag{2.2}$$

The training process was carried out on an NVIDIA Titan Xp GPU using the PyTorch framework. The performance of the five DEDUCE algorithms on the test set of Places365 is shown in Table 2.2 for the seven classes chosen.

2.4.2 Experiment Settings

To check the robustness of the proposed models, their performance is further evaluated on two state-of-the-art still-image datasets.

Table 2.3. Accuracy in percentage of DEDUCE on SUN dataset

Scenes	Φ_{scene}	Φ_{obj}	$\Phi_{attn.}$	$\Phi_{comb.}$	Φ_{N-best}
Dining room	65.2	**83.7**	53.3	67.4	72.8
Bedroom	43.7	36.5	**48.9**	**48.9**	47.3
Bathroom	94.5	87.0	**97.3**	96.6	95.2
Corridor	44.4	**67.6**	**67.6**	44.4	41.7
Living Room	58.8	24.0	43.6	**59.2**	58.8
Office	84.0	12.6	75.8	**90.6**	80.6
Kitchen	77.1	63.5	63.9	**83.8**	77.4
Avg	66.8	53.6	64.3	**70.1**	67.7

SUN Dataset

The SUN-RGBD dataset [147] is one of the most challenging scene understanding datasets in existence. It consists of $3,784$ images using Kinect v2 and $1,159$ images using Intel RealSense cameras. In addition, there are $1,449$ images from the NYUDepth V2 [145], and 554 manually selected realistic scene images from the Berkeley B3DO Dataset [63], both captured by Kinect v1. Finally, it has $3,389$ manually selected distinguished frames without significant motion blur from the SUN3D videos [176] captured by Asus Xtion. Out of this, the seven classes of importance are sampled and the official test split is used to evaluate the presented models. Only the RGB images are considered for this work since the training data doesn't have depth information. The performance is summarized in Table 2.3.

Upon comparison with Table 2.2, which contains the results on the Places365 dataset where the models were fine-tuned, a number of observations can be made that are consistent for both datasets. Firstly, the $\Phi_{comb.}$ model performs the best. This is intuitive since here, the scene classification is done using the combined training of both the information about the scene attributes and the object identity. Secondly, the Φ_{obj} model works the best for the *Dining Room* class, even though its overall performance is the worst. This trend can be attributed to the fact that dining rooms can be easily identified by the presence of specific objects, whereas the scene attributes might throw in some confusion (for instance when the kitchen/living room is partially visible in the image of a dining room). Thirdly, for *Corridor*, the performance of the $\Phi_{attn.}$ model

Table 2.4. VPC Dataset: Average Accuracy across the 6 home environments

Networks	H1	H2	H3	H4	H5	H6	avg.
AlexNet	49.8	53.4	49.2	64.4	41.0	43.4	50.2
AlexNet+BN	54.5	54.6	55.6	69.7	41.8	45.9	53.7
AlexNet+WBN	54.7	51.9	61.8	70.6	43.9	46.5	54.9
AlexNet+WBN*	53.5	54.6	55.7	68.1	44.3	49.9	54.3
ResNet	55.8	47.4	64.0	69.9	42.8	50.4	55.0
ResNet+WBN	55.7	49.5	**64.7**	70.2	42.1	52.0	55.7
ResNet+WBN*	56.8	50.9	64.1	69.3	45.1	51.6	56.5
Ours (Φ_{scene})	**63.7**	57.3	63.7	**71.4**	60.2	65.9	63.7
Ours ($\Phi_{comb.}$)	**63.7**	**60.7**	64.5	70.7	**65.7**	**68.8**	**65.7**

is best for both the datasets. This supports the fact that to classify a scene like a corridor, viewing only a small portion of the image close to the vanishing point is sufficient. Finally, the Φ_{N-best} model performs just as well or better than the Φ_{scene} model. This proves that the presence of objects does indeed improve the scene classification. For the best performance using the Φ_{N-best} model, the threshold was set to 0.5 for the *Places* dataset while it was 0.6 for the *SUN* dataset. The reason for the higher confidence on scene attributes for *Places* dataset is most likely because the scene classifier itself was fine-tuned on it.

VPC Dataset

The Visual Place Categorization dataset [173] consists of videos captured autonomously using an HD camcorder (JVC GR-HD1) mounted on a rolling tripod. The data has been collected from 6 different home environments and three different floor types. The advantage of this dataset is that the collected data closely mimics that of the motion of a robot - instead of focusing on captured frames or objects/furniture in the rooms, the operator recording the data just traversed across all the areas in a room while avoiding collision with obstacles. For comparison with the state-of-the-art algorithms, the methods are tested only on the five classes that are present in all the homes - `Bathroom`, `Bedroom`, `Dining room`, `Living room`, and `Kitchen`. Table 2.4 contains the results for the individual home environments for these five classes. For the AlexNet [84] and ResNet [57] models, the same training procedure is adopted as in [104]. It can be seen from the

Table 2.5. VPC Dataset: Comparison with State Of The Art

Method	[173]				[44]	[179]		AlexNet			ResNet		**Ours**	
Config.	SIFT	SIFT+BF	CE	CE+BF	HOUP	G+BF	G+O(SIFT)+BF	Base	BN	WBN*	BN	WBN*	**Scene-only**	**Combined**
Acc.	35.0	38.6	41.9	45.6	45.9	47.9	50	50.2	53.7	54.3	55.0	56.5	**63.7**	**65.7**

table that the proposed models perform better than the rest in all but one of the home environments and much better in the overall performance.

Table 2.5 further compares the models with all other baseline algorithms tested on the VPC dataset. The reported accuracies are the average over all the six home environments. First, the methods described in [173] are considered, which use SIFT and CENTRIST features with a Nearest Neighbor Classifier and also exploit temporal information between images by coupling them with Bayesian Filtering (BF). Next, the approach of [44] is looked at, where Histogram of Oriented Uniform Patterns (HOUP) is used as input to the same classifier. [179] proposed the method of using object templates for visual place categorization, and reported results for the Global configurations approach with Bayesian Filtering (G+BF), and that combined with the object templates (G+O(SIFT)+BF). Ushering the deep learning era, AlexNet [84] and ResNet [57] architectures give better results, both with their base models, as well as the Batch Normalized (BN) and the Weighted Batch Normalized versions [104]. However, comparisons with the Φ_{scene} and $\Phi_{comb.}$ models show that the proposed methods beat all the other results by significant margins.

Do objects play an important role in prediction?

Figure 2.4 shows the attention maps obtained from the Scene+Attention model for six different scenes of the Places365 dataset [186]. The hotpots present in each scene are typically around characteristic objects, such as near the toilet and basin in `Bathroom`, bed in `Bedroom`, or stovetop and shelves in `Living room`. This shows the importance of identifying objects for predicting scene labels.

Another interesting result is observed in Figure 2.5, where the scene prediction accuracy of the Φ_{scene}, and $\Phi_{comb.}$ models are plotted as a bar graph. It is evident that for nearly every scene category, the combined model, which jointly uses object, and scene information outperforms the scene only model. The only comparable results are for `Corridor` – possibly because there are no characteristic objects for this scene category, and `Living room` – as living rooms typically

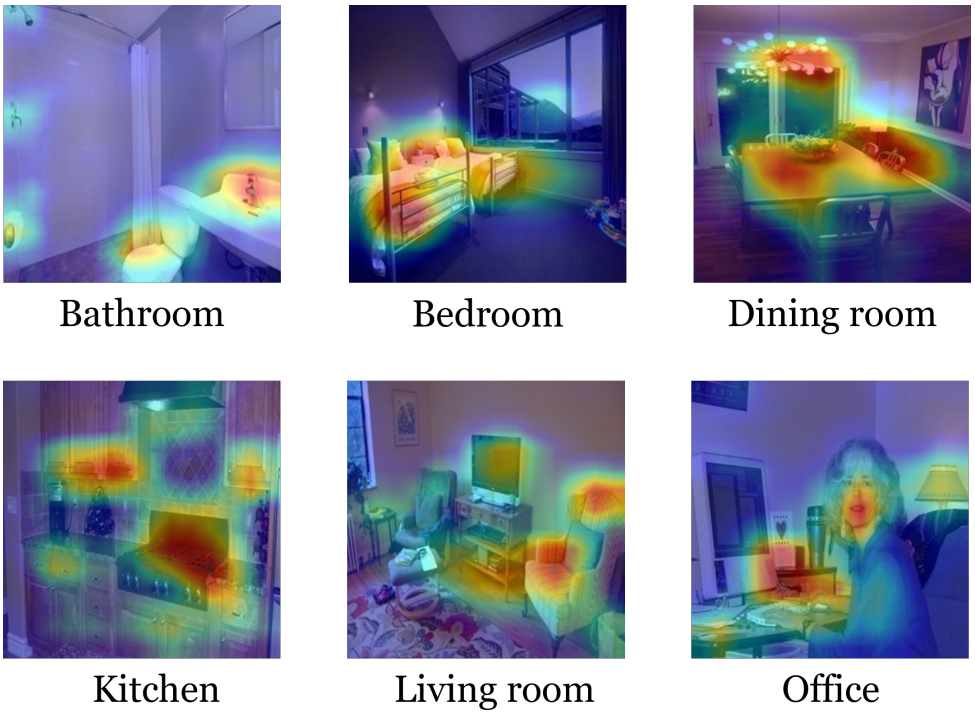


Figure 2.4. Visualization of attention maps for different scenes of the Places365 [186] dataset

contain a very noisy distribution of objects from multiple rooms.

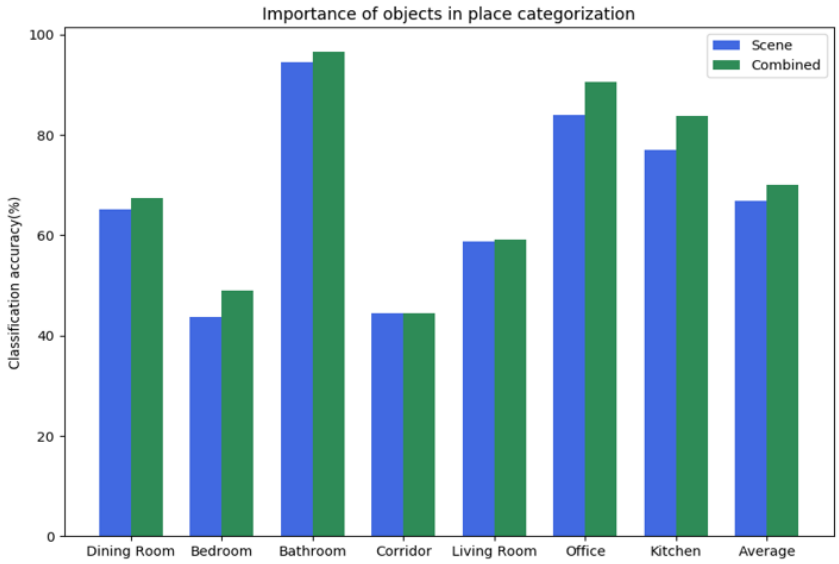


Figure 2.5. Comparison of scene-only, and combined models for different scenes of the Places365 [186] dataset

Real-World Scene Recognition

To test the robustness of the presented DEDUCE models, the domain of test cases is expanded beyond the aforementioned still image data sets. Figure 2.6 does this by showing the results of scene recognition on real-world data recorded using hand-held cameras. The Φ_{N-best} model is deployed for these cases due to its ability to mimic the natural behavior of humans, whereby an initial prediction is made based on the scene attributes, and if unsure, more information related to specific objects are gathered to update/reinforce the initial prediction. The top row corresponds to the tour of a semi-furnished real estate home obtained from YouTube which only has the relevant objects in the scene. Although it is not a sequential tour of the house, it does contain all the rooms. Also, professional photographers captured this video and hence, the image quality and white balance of the camera are pretty good. The next two rows pose a more challenging case as they correspond to homes currently inhabited by people. Two examples of these houses are considered, one which is a standard bungalow residence, while the other is a student apartment. From experience, the bungalow is a much cleaner home, whereas student apartments are prone to the presence of cluttered objects and overlapping scene boundaries. Moreover, the videos were recorded by the inhabitants using their cellphone cameras. This inherently brings motion blur into the picture, especially during scene transitions. Finally, the last row depicts the settings of a house from the movie *"Father of the Bride"*. This ensures that the proposed model is robust enough to classify scenes even when the focus of the recording is on people instead of the background settings. All the detection results mentioned in this chapter are available as individual videos in the following link https://goo.gl/sYyVZ2.

Semantic Mapping

The experimental setting for semantic mapping involves running the presented algorithm on a mobile robot platform in two different environments. The platform is a Fetch Mobile Manipulator and Freight Mobile Robot Base by Fetch Robotics[1]. Figure 2.1 shows the robot

[1] https://fetchrobotics.com/robotics-platforms/

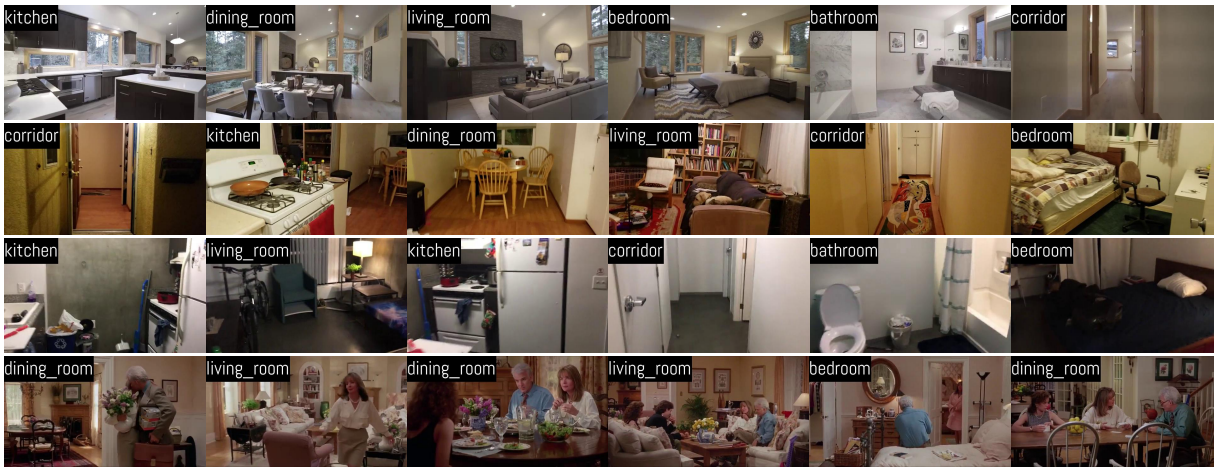


Figure 2.6. Detection Results on Real-World Videos. The top row corresponds to the video of a Real-Estate model house. The next two rows are from the houses of the authors and their friends. The bottom row is obtained from a house in the movie *"Father of the Bride"*.

performing scene classification in one of the environments.

The semantic maps for the experiments were constructed using Omnimapper [159]. It utilizes the GTsam library to optimize a graph of measurements between robot poses along a trajectory, and between robot poses and various landmarks in the environment. The measurements of simple objects like points, lines, and planes are data associated with mapped landmarks using the joint compatibility branch and bound (JCBB) technique [114]. The regions for color segmentation are acquired by the Gaussian Region algorithm of [115]. However, in [115], the map partitions were built through human guidance, whereby the robot was taken on a tour of the space (either by driving the robot manually or using a person following behavior) and the respective scene labels were taught to it. This is in contrast to the approach in this work, where the labels are learned from the visual place categorization system. Thus, the robot is capable of identifying the scenes by itself without any human guide. The Φ_{N-best} model is used for this task and retrained the scene classifier to exclude the *Bedroom*, *Dining Room* & *Bathroom* scenes, and instead include *Conference Room* as it is more likely to occur in an academic building environment.

Figure 2.7a shows the navigation of the robot in the Computer Science and Engineering

(CSE) Building. The developed system in this work was able to classify the seven regions of the floor map. However, there are some regions detected by Omnimapper using the laser range finder. These are painted in white to denote their invisibility to the camera. The second test environment is the Contextual Robotics Institute (CRI) building, which has a very different floor map in comparison to CSE. The result of the run made here is shown in Figure 2.7b.

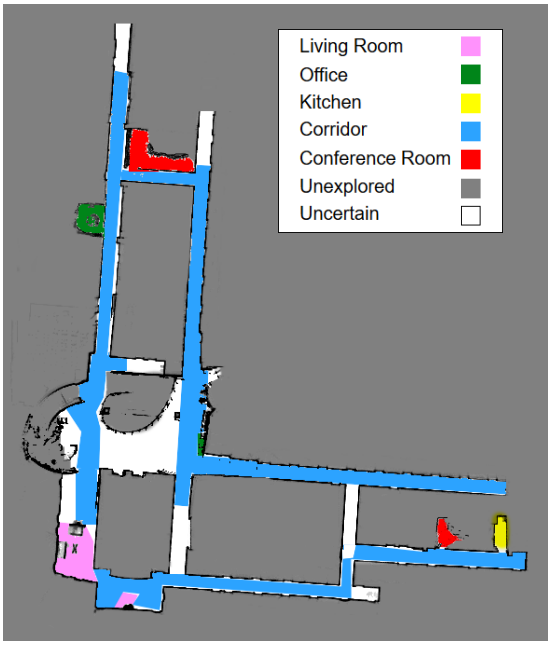
2.5 Conclusion

In this chapter, five different models are considered for performing place categorization, which is derived mainly from two base modules - a scene recognizer, and an object detector. The effectiveness of these algorithms is demonstrated through a series of experiments, ranging from scene recognition in still-image data sets to real-world videos captured from different sources, and finally via the generation of labeled semantic maps using data gathered by multiple mobile robot platforms.

It has been shown that (i) different models are favorable for different scenes (Table 2.2 and 2.3), and thus the ideal scene recognition system would likely be a combination of these five models, (ii) the proposed methods give successful results on many different types of video recordings, even when they are affected by object clutter, motion blur, and overlapping boundaries and (iii) the proposed models are robust enough to be tested on data gathered by mobile robotic platforms on multiple building scenarios which are affected by occlusions and poor lighting conditions.

2.6 Acknowledgements

Chapter 2, in part, is a reprint of **A. Pal**, C. Nieto-Granda and H. I. Christensen, "DEDUCE: Diverse scEne Detection methods in Unseen Challenging Environments", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019*. The dissertation author was the primary author of this paper.



(a) Semantic Map of the CSE building.

(b) Semantic Map of the CRI.

Figure 2.7. Place categorization experiments with mobile robots. Each color represents one of the seven classes of visual place categorization that the proposed system classified.

Chapter 3

Context recognition in autonomous driving

Cameras are one of the most powerful sensors in the world of robotics as they capture detailed information about the environment, and thus can be used for object detection [164, 97] and segmentation [161, 162] - something that is much harder to achieve with a basic range sensor. However, an image/video may contain some irrelevant information. Therefore, there is a need to filter out these unimportant regions and instead, learn to focus the "attention" on parts of the image that are necessary to solve the task at hand. This is crucial for autonomous driving scenarios, where a vehicle should pay more attention to other vehicles, pedestrians, and cyclists present in its vicinity while ignoring inconsequential objects like trees or buildings far away from the road. Upon successfully identifying the objects of interest, the controller driving the vehicle only needs to attend to them to make optimal decisions.

In this work, a novel framework is proposed for predicting the driver's focus of attention through a learned saliency map by taking into consideration the semantic context of an image. Typical saliency prediction algorithms [121, 122, 175, 154] in driving scenarios rely only on human-gaze information, either through an in-car [6], or in-lab [175] setting. However, gaze by itself does not completely describe everything a driver should attend to, mainly due to the following reasons:

1. **Peripheral vision:** Humans tend to rely on peripheral vision, thus giving them the ability

to fixate their eyes on one object while attending to another. This cannot be captured by an eye-tracking device. Thus, only in-car driver gaze [6] does not convey sufficient information. While the in-lab annotation does alleviate this problem to some extent [175] by aggregating the gazes of multiple independent observers, it does not completely remove it since that relies on real human gaze too.

2. **Single focus:** When a human driver realizes that the trajectory of an incoming car or pedestrian is not likely to collide with that of the ego vehicle, they tend to shift their gaze away from the oncoming traffic as it approaches. This is a major cause of accidents. To address this, a method is proposed for tracking the motion of every driving-relevant object by detecting its instances until it goes beyond the field of view of the camera. This is possible because the limitation of a human's ability to single focus does not apply to an autonomous vehicle system.

3. **Distracted gaze:** A human driver while driving the car might often get distracted by some roadside object - say a brightly colored building or some attractive billboard advertisement. To tackle this issue, the proposed method is only trained to detect those objects that influence the task of driving. The in-lab gaze [175] also eliminates this noise by averaging the eye movements of independent observers. However, they assume that the people annotating are positioned in the co-pilot's seat, and therefore cannot realistically emulate a driver's gaze.

4. **Center-bias:** For the majority of a driving task, the human gaze remains on the road in front of the vehicle as this is where the vehicle is headed. When deep learning models are trained on this gaze map, they invariably recognize this pattern and learn to keep the focus there. However, this is not enough since there might be important regions away from the center of the road that demand attention - such as when cars or pedestrians approach from the sides. Thus, relying only on gaze data does not help capture these important cues.

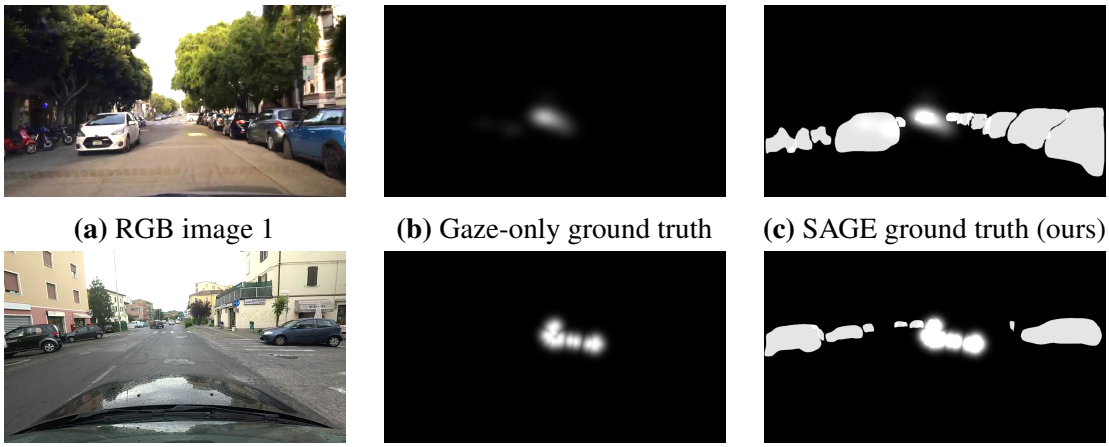
(a) Input image **(b) SAGE-Net (our)**

(c) BDD-A [175] **(d)** DR(eye)VE [6]

Figure 3.1. Predicted saliency map for different models (Best viewed in color). The bounding box shows a pedestrian illegally crossing the road and is prone to accidents. While other models only capture the car ahead (partially), the proposed model can **completely** learn to detect both the car and the crossing pedestrian.

Figure 3.1 shows an example of an accident-prone situation, where the predicted saliency maps from an algorithm trained using different target labels are shown. Gaze-only models were able to detect the car ahead, but completely missed the pedestrian jaywalking. In contrast, the proposed approach successfully detects both objects since it has learned to predict the semantic context in an image.

It is important to note, however, that semantics alone does not completely provide insights into the action that a driver might take at run-time. This is because a saliency map obtained only from training on semantics will give equal-weighted attention to all the objects present. Also, when there is no object of relevance, say on an empty road near the countryside, this saliency map will not provide any attention. In reality, here the focus should be on road boundaries, lane dividers, curbs, etc. These regions can be effectively learnt through gaze information which is



(a) RGB image 1 **(b)** Gaze-only ground truth **(c)** SAGE ground truth (ours)

(d) RGB image 2 **(e)** Gaze-only ground truth **(f)** SAGE ground truth (ours)

Figure 3.2. Comparison of SAGE with the existing gaze-only ground truths. The top row [a-c] is for the BDD-A dataset [175] while the bottom row [d-f] is for the DR(eye)VE dataset [6]. The gaze-only maps indicate the heading of the ego-vehicle but completely ignore the nearby and incoming cars. In contrast, SAGE captures both the driver's intent and the relevant objects.

an indicator of a driver's intent. Thus, a Semantics Augmented GazE (SAGE) *ground-truth* is designed in this work, which successfully captures both gaze and semantic context. Figure 3.2 shows how the proposed ground truth looks as compared to the existing gaze-only ground truths.

There are three novel contributions made in this work. Firstly, a combined attention mechanism called Semantics Augmented GazE (SAGE) is proposed, which can be used to train saliency models for accurately predicting an autonomous vehicle's (hereafter termed as a driver) focus of attention. Secondly, a thorough saliency detection framework called SAGE-Net has been provided by including important cues in driving such as distance to objects (depth), speed of ego vehicle, and pedestrian crossing intent to further enhance the initial raw prediction obtained from SAGE. Finally, a series of experiments have been conducted using multiple saliency algorithms on different driving datasets to evaluate the flexibility, robustness, and adaptability of SAGE - both over the entire dataset, and also specific important driving scenarios such as intersections and busy traffic regions. The remainder of the chapter is organized as follows. Section 3.1 discusses the existing state-of-the-art research in driver saliency prediction. Section 3.2 then provides details of the proposed framework, followed by the extensive experiments conducted in Section

3.3. Finally, Section 3.4 concludes the discussion and mentions the real-world implications of the conducted research.

3.1 Background

Advances in Salient Object Detection: Detection [164, 97] and segmentation [161, 162] of salient objects in the natural scene has been a very active area of research in the computer vision community for a long time. One of the earliest works in saliency prediction by Itti *et al*.[61] considered general computational frameworks and psychological theories of bottom-up attention, based on center-surround mechanisms [158, 169, 73]. Subsequent behavioral [123] and computational investigations [16] used "fixations" as a means to verify the saliency hypothesis and compare models. The proposed approach differs from theirs due to the incorporation of both a bottom-up strategy by scanning through the entire image and detecting object features that are relevant for driving, as well as a top-down strategy by incorporating the human gaze which is purely task-driven. Some later studies [97, 3] defined saliency detection as a binary segmentation problem. This work adopts a similar strategy, but instead of using handcrafted features that do not generalize well to real-world scenes, deep-learning techniques are used for robust feature extraction. Since the introduction of Convolutional Neural Networks (CNNs), several approaches have been developed for learning global and local features through varying receptive fields, both for 2D image datasets [164, 96, 25, 45], and video-based saliency predictions [163, 99, 43, 118]. However, these algorithms are either too heavily biased toward image datasets or involve designs of complicated architectures which make them difficult to train. In contrast, the proposed approach helps to improve existing architectures without any additional training parameters, thereby keeping the complexity unchanged. This is very important for an autonomous system since it needs to be as close to real-time as possible. For a detailed survey of salient object detection, please refer to the work by Borji *et al*.[12].

Saliency for driving scenario: Lately, there has been some focus on driver saliency

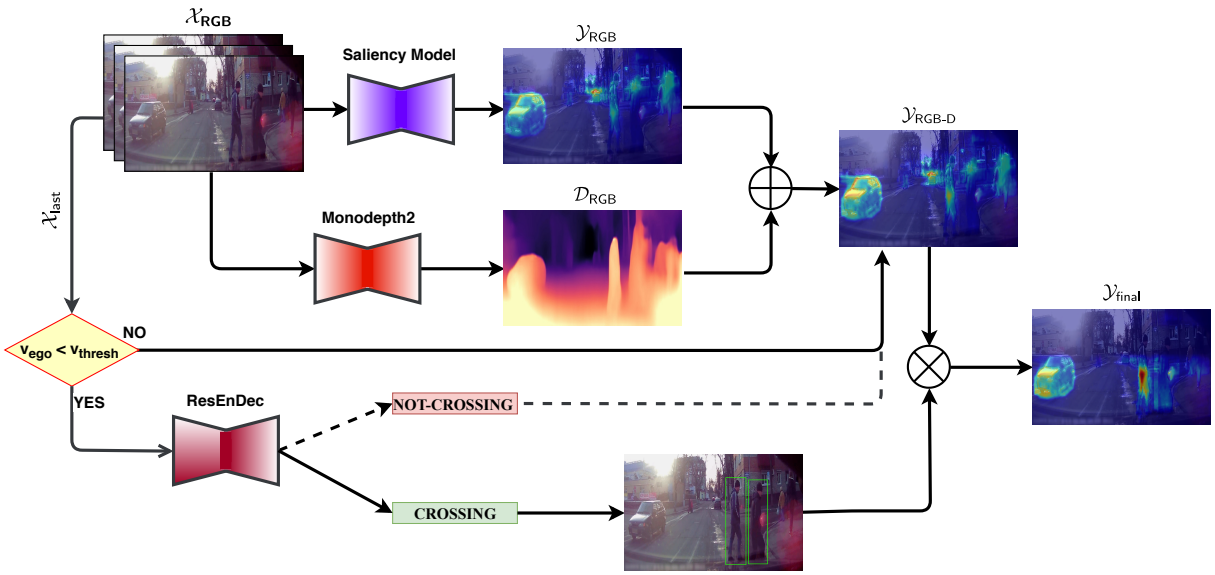


Figure 3.3. The complete **SAGE-Net** framework (Best viewed in color), comprising of a saliency model trained on **SAGE** groundtruth, and added parallel modules for depth estimation and pedestrian intent prediction based on ego-vehicle speed (v_{ego}).

prediction due to rise of the number of driving [82, 182, 131, 124, 113] and pedestrian tracking [33, 42, 82] datasets. Most saliency prediction models are trained using human gaze information, either through in-car eye trackers [6, 121], or through in-lab simulations [175, 154]. However, as discussed above, these methods only give an estimate of the gaze, which is often prone to center bias, or distracted focus. In contrast, the proposed approach involves combining scene semantics along with the existing gaze data. This ensures that the predicted saliency map can effectively mimic a real driver's intent, with the added feature of also being able to successfully detect and track important objects in the vicinity of the ego vehicle.

3.2 SAGE-Net: Semantic Augmented GazE detection Network

Figure 3.3 provides a simplified illustration of the entire SAGE-Net framework, which comprises three components: a SAGE detection module, a distance-based attention update module, and finally a pedestrian intent-guided saliency module. Firstly, a description of how the SAGE maps are obtained is given in Section 3.2.1. Next, Section 3.2.2, describes how relative

distances of objects from ego vehicle should impact saliency prediction. Lastly, in Section 3.2.3, the importance of pedestrian crossing intent detection is highlighted along with how it influences the focus of attention.

3.2.1 SAGE saliency map computation

In this work, a new approach is proposed for predicting driving attention maps which not only uses raw human gaze information but also learns to detect the scene semantics directly. This is done using the Mask R-CNN (M-RCNN) [56] object detection algorithm, which returns a segmented mask around an object of interest along with its identity and location.

For the instance segmentation task, the Matterport implementation of M-RCNN [2] is used, which is based on Feature Pyramid Network (FPN) [90] and adopts ResNet-101 [57] as backbone. The model is trained on the MS-COCO dataset [92]. However, out of the total 80 objects in [92], the following 12 categories which are most relevant to driving scenarios are selected - `person`, `bicycle`, `car`, `motorcycle`, `bus`, `truck`, `traffic light`, `fire hydrant`, `stop sign`, `parking meter`, `bench` and `background`. For each video frame, M-RCNN provides an instance segmentation of every detected object. However, as the relative importance of different instances of the same object is not a significant cue, a binary classification approach is used where all objects are segmented against the background. This object-level segmented map is then superimposed on top of the existing gaze map provided by a dataset, to preserve the gaze information. This gives out the final saliency map as seen in Fig 3.2. Upon inspection, it can be seen that the proposed ground truth has managed to capture a lot more semantic context from the scene, which gaze-only maps have missed.

3.2.2 Does relative distance between objects and ego vehicle impact focus of attention?

Depth estimation through supervised [39, 94, 107] and unsupervised [46, 160] learning methods, as a measure of relative distance between objects and ego vehicle has been a long

studied problem in the autonomous driving community [102, 48, 49]. Human beings inherently react and give more attention to vehicles and pedestrians that are "closer" to them as opposed to those at a distance, since chances of collision are much higher for the former case. Unfortunately, this crucial information is yet to be exploited for predicting driving saliency maps to the best of the author's knowledge. Therefore, this work considers it through the recently proposed self-supervised monocular depth estimation approach - Monodepth2 [49]. It should be noted that SAGE-Net is not restricted to just this algorithm, but can effectively inherit stereo or LiDAR-based depth estimators into its framework as well.

Two methods of incorporating depth maps into the presented framework have been considered. The first involves taking a parallel depth channel which does not undergo any training but is simply used to amplify nearby regions of the predicted saliency map. The second method is to use it as a separate trainable input to the saliency prediction model along with the raw image, like how optical flow and semantic segmentation maps are trained in [121]. Eventually, the first strategy was decided because, in addition to being much simpler and faster to implement, it also removes the issue of training a network only on a depth map which has a lot less variance in data, thus leading to overfitting towards the vanishing point in the image.

Given an input clip of 16 RGB image frames, $\mathbf{X}_{\text{RGB}} \in \mathbb{R}^{16\times3\times h\times w}$, the raw saliency map prediction $Y_{\text{RGB}} \in \mathbb{R}^{h\times w}$ is obtained. In addition, for each frame, the depth map $D_{\text{RGB}} \in \mathbb{R}^{h\times w}$ is also computed. Finally, the raw prediction is combined with the depth map to obtain $Y_{\text{RGB-D}}$ using the $\oplus$ operator, which is defined as

$$Y_{\text{RGB}} \oplus D_{\text{RGB}} = Y_{\text{RGB}} * D_{\text{RGB}} + Y_{\text{RGB}} \tag{3.1}$$

where $*$ and $+$ denote element-wise multiplication and addition respectively.

3.2.3 Does extra attention need to be paid to pedestrians crossing at intersection scenarios?

Accurate pedestrian detection in crosswalks is a vital task for an autonomous vehicle. Thus, an additional module is included in this framework which focuses solely on the crossing intent of pedestrians at intersections, and correspondingly updates the saliency prediction. It should be noted that even though SAGE does capture information about pedestrians in its raw prediction in general driving scenarios, it does not distinguish between them and other objects in crowded traffic conditions such as intersections. This is critical since the chances of colliding with a pedestrian are much higher around intersection regions than on other roads. However, this is a slow process since it involves detecting pedestrians and predicting their pose at run-time. Fortunately, this situation only occurs when the speed of the ego vehicle itself is less. Thus, specialized detection of pedestrians is only included in this framework when the speed of the ego vehicle ($\mathtt{v_{ego}}$) is below a certain threshold velocity $\mathtt{v_{thresh}}$. It is not very difficult to obtain $\mathtt{v_{ego}}$ since most driving datasets provide this annotation [6, 175]. Also, for an autonomous vehicle, the odometry reading contains this. $\mathtt{v_{thresh}}$ is a tunable hyper-parameter which can vary as per the road and weather conditions. When $\mathtt{v_{ego}} < \mathtt{v_{thresh}}$, a check is done to see if pedestrians are crossing the road. This is done using the recently proposed algorithm ResEnDec [53] which predicts the intent I of pedestrians as "`crossing`" or "`not crossing`" through an encoder-decoder framework using a spatiotemporal neural network and ConvLSTM. This algorithm is trained on the JAAD [82] dataset, considering 16 consecutive frames to be the temporal strip while predicting the last frame X_{last}. The proposed framework is designed such that if the prediction is "`crossing`", an object detector O, such as YOLOv3 [134], is used to get the bounding box of the pedestrians from that last frame. Consequently, the predicted attention for pixels inside the bounding boxes is amplified, while leaving the rest of the image intact. This

is given by the $\otimes$ operator, defined as follows

$$Y_{\text{RGB-D}} \otimes \textbf{bbox} = \begin{cases} Y_{\text{RGB-D}}[x,y] * k \;\forall\; (x,y) \in \textbf{bbox} \\ Y_{\text{RGB-D}}[x,y] * 1/k, \text{ else} \end{cases} \tag{3.2}$$

where k is an amplification factor (> 1) by which the predicted map is strengthened, while $*$ and $+$ denote element-wise multiplication and addition respectively. If the predicted intent is "`not crossing`", the original prediction $Y_{\text{RGB-D}}$ remains the final output saliency map. The summary of the entire SAGE-Net algorithm is depicted in Algorithm 1.

Algorithm 1: The complete SAGE-Net pipeline

Input: RGB image $\mathbf{X}_{\text{RGB}}$, threshold velocity $\texttt{v}_{\texttt{thresh}}$, ego vehicle velocity $\texttt{v}_{\texttt{ego}}$
Object detector: O
$Y_{\text{RGB}} \leftarrow$ Saliency model$(\mathbf{X}_{\text{RGB}})$
$X_{\text{last}} \leftarrow \mathbf{X}_{\text{RGB}}[-1]$
$D_{\text{RGB}} \leftarrow$ Monodepth2(X_{last})
$Y_{\text{RGB-D}} \leftarrow Y_{\text{RGB}} \oplus D_{\text{RGB}}$
if $\texttt{v}_{\texttt{ego}}(X_{last}) > \texttt{v}_{\texttt{thresh}}$ **then**
  return $Y_{\text{RGB-D}}$
else
  $I_{X_{\text{last}}} \leftarrow$ ResEnDec$(\mathbf{X}_{\text{RGB}})$
  if $I_{X_{last}} =$ `not crossing` **then**
    return $Y_{\text{RGB-D}}$
  else
    bbox $\leftarrow O(X_{\text{last}})$
    $Y_{\text{final}} \leftarrow Y_{\text{RGB-D}} \otimes$ **bbox**
    return Y_{final}

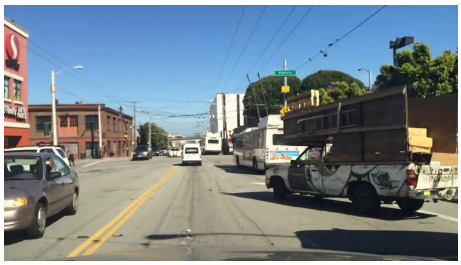
3.3 Experiments and Results

Due to the simplicity of computation of the proposed ground truth, several experiments can be run using it. These experiments can be split into a two-stage hierarchy - (i) conducted over the entire dataset comprising of multiple combinations in driving scenarios - day vs night, city vs countryside, intersection vs highway, etc. and (ii) those over specific important driving conditions such as intersection regions and crowded streets. The reason for the latter set of experiments

is that averaging out the predicted results over all scenarios is not always reflective of the most important situations requiring maximum human attention [175]. For all the experiments, the evaluation metrics used for comparison are described, and using those, the results of the gaze-only ground truth and the proposed SAGE ground truth are compared for different algorithms and datasets.

3.3.1 Some popular saliency prediction algorithms

Four popular saliency prediction algorithms are selected from an exhaustive list for training with SAGE ground truth and their performances are compared against those trained with gaze-only maps. The first set of algorithms, DR(eye)VE [122] and BDD-A [175], were created exclusively for saliency prediction in the driving context. For DR(eye)VE, only the image branch is considered for the analysis instead of the multi-branch network [121], due to two main reasons that make real-time operation possible. Firstly, it has a fraction of the number of trainable parameters and hence is faster to train and evaluate. Secondly, the latter assumes that the optical flow and semantic segmentation maps are pre-computed even at test time, which is difficult to achieve online. The BDD-A algorithm is more compact and it consists of a visual feature extraction module [84], followed by a feature and temporal processing unit in the form of 2D convolutions and Convolutional LSTM (Conv2D-LSTM) [177] network respectively. However, both these algorithms combine the features extracted from the final convolution layers to make the saliency maps. This mechanism ignores low-level intermediate representations such as edges and object boundaries, which are important detections for driving scenarios. Thus, another algorithm called ML-Net [27] is considered, which achieved the best results on the largest publicly available image saliency dataset SALICON [64]. It extracts low, medium, and high-level image features and generates a fine-grained saliency map from them. Finally, PiCANet [95] extends this notion further by generating an attention map at each pixel over a context region and constructing an attended contextual feature to further enhance the feature representability of ConvNets.



(a) RGB Image

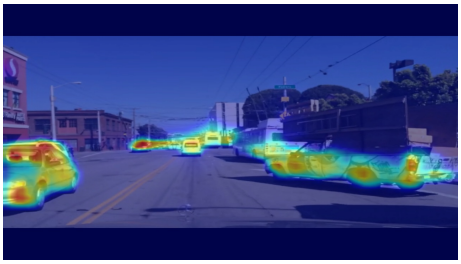
(b) DR(eye)VE [122] with BDDA gt

(c) DR(eye)VE [122] with SAGE gt

(d) BDDA [175] with BDDA gt

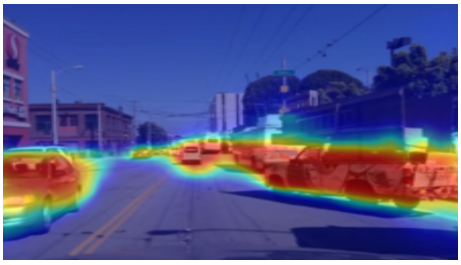
(e) BDDA [175] with SAGE gt

(f) ML-Net [27] with BDDA gt



(g) ML-Net [27] with SAGE gt

(h) PiCANet [95] with BDDA gt



(i) PiCANet [95] with SAGE gt

Figure 3.4. Comparison of the prediction of four popular saliency models trained on the BDD-A ground-truth (middle row) and our SAGE groundtruth (bottom row). It can be seen that for each model, SAGE trained results can capture more detailed semantic context (Best viewed in color).

Figure 3.4 shows a comparison of the predicted saliency maps trained on gaze-only ground-truth, and those obtained from SAGE. For nearly every gaze-only model, the focus of attention is entirely towards the center of the image, thereby ignoring other cars. In contrast, SAGE-trained models have managed to successfully capture this vital information. Please refer to Section 3.5.2 of the supplementary material for implementation details of these four algorithms.

3.3.2 Evaluation metrics

For evaluation, a set of metrics that are suitable for evaluating saliency prediction in the driving context are considered, as opposed to general saliency prediction. More specifically, for driving purposes, it is important to be more careful about identifying "False Negatives (FN)" than "False Positives (FP)", since the former error holds a much higher cost. As illustrated in Section 3.2, the proposed ground truth has both a gaze component and a semantic component. Thus, the set of metrics is broadly classified into two categories - (i) fixation-centric and (ii) semantic-centric.

For the first category, two distribution-based metrics are chosen - Kullback-Leibler Divergence (D_{KL}), and Pearson's Cross Correlation (CC). D_{KL} is an asymmetric dissimilarity metric, that penalizes FN more than FP. CC, on the other hand, is a symmetric similarity metric that equally affects both FN and FP, thus giving an overall information regarding the misclassifications that occurred. Other variants of fixation metrics are the location-based metrics, such as Area Under ROC Curve (AUC), Normalized Scanpath Saliency (NSS), and Information Gain (IG), which operate on the ground-truth being represented as discrete fixation locations [18]. But for the driving task, it is crucial to identify every point on a relevant object, especially their boundaries, to mitigate risks. Thus, continuous distribution metrics are more appropriate here as they can better capture object boundaries.

In the second category, again two metrics are considered - namely F-score, which measures region similarity of detection, and Mean Absolute Error (MAE), which gives pixel-

wise accuracy. F-score is given by the formulae,

$$F_\beta = \frac{(1+\beta^2) * precision * recall}{\beta^2 * precision + recall} \tag{3.3}$$

where β^2 is a parameter that weighs the relative importance of precision and recall. In most literatures [163, 4, 87], β^2 is taken to be 0.3, thus giving a higher weightage to precision. However, following the earlier discussion regarding varying costs associated with FN and FP for the driving purpose, β^2 is considered to be 1, thereby assigning equal weightage to each. For formal proof of this, please refer to Section 3.5.1 of the supplementary material.

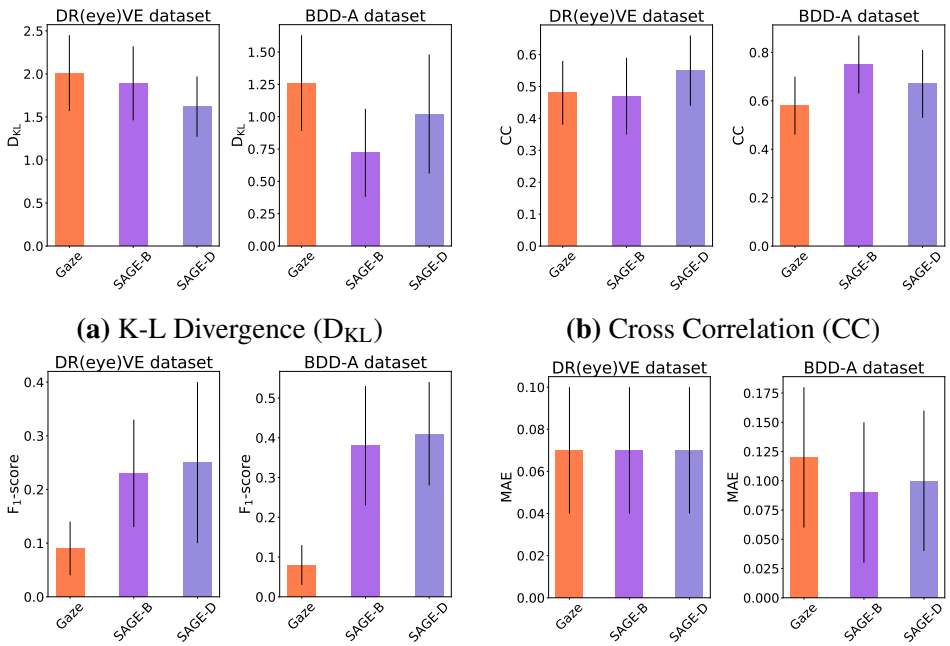
3.3.3 Results and Discussion

In this section, the experiments and results of algorithms trained on the proposed SAGE ground truth are discussed, along with how they compare to the performance of the same algorithms, when trained on existing gaze-only ground truths [6, 175]. The results of the proposed method are compared with that of the BDD-A gaze in most of the experiments since it is more reflective of scene semantics than the DR(eye)VE gaze. For a fair comparison, different strategies for evaluating the fixation-centric and semantic-centric metrics are adopted. Since both the traditional gaze-only approach and SAGE contain gaze information, respective ground-truths are used to evaluate the fixation metrics (i.e. gaze for the gaze-only trained model, and SAGE for the proposed trained model). However, for the semantic metrics, the segmented maps generated by Mask RCNN are used as ground truth to evaluate how well each of the methods can capture semantic context. The first set of comparisons, given by Table 3.1 and Figure 3.5, are calculated by taking the average over the entire test set, while the remaining comparisons are for a subset of the test set involving two important driving scenarios, namely - pedestrians crossing at an intersection in Table 3.2, and cars approaching towards the ego vehicle in Table 3.3.

Overall comparison - In Table 3.1, the four algorithms described in Section 3.3.1 are trained on the BDD-A dataset [175]. The results shown are obtained when evaluating the

Table 3.1. Comparison of different saliency algorithms trained on BDD-A gaze gt and SAGE gt. All experiments are conducted on the BDD-A dataset.

	Fixation-centric metrics				Semantic-centric metrics			
	D_{KL}		CC		F_1 score		MAE	
Model	Gaze gt	**SAGE gt**	Gaze gt	**SAGE gt**	Gaze gt	**SAGE gt**	Gaze gt	**SAGE gt**
DREYEVE [121]	1.28±0.43	**0.73±0.38**	0.58±0.13	**0.75±0.13**	0.1±0.06	**0.37±0.14**	0.11±0.06	**0.08±0.05**
BDDA [175]	1.34±0.67	**1.02±0.49**	0.54±0.23	**0.6±0.18**	0.12±0.11	**0.46±0.19**	**0.12±0.09**	0.13±0.07
ML-Net [27]	**1.1±0.32**	1.35±0.51	**0.64±0.13**	0.6±0.14	0.12±0.07	**0.43±0.14**	0.12±0.06	**0.1±0.06**
PiCANet [95]	1.11±0.28	**0.83±0.31**	0.64±0.11	**0.73±0.11**	0.15±0.08	**0.64±0.15**	0.11±0.06	**0.11±0.05**



(a) K-L Divergence (D_{KL})

(b) Cross Correlation (CC)

(c) F_1 score

(d) Mean Absolute Error (MAE)

Figure 3.5. Cross-evaluation of SAGE-gt by considering the gaze of two different datasets. [6] and BDD-A [175] have been used for comparison. SAGE-B/D refers to the combination of semantics with the gaze of BDD-A/DR(eye)VE dataset.

algorithms trained on the gaze-only data, and then on SAGE data generated by combining semantics with the gaze of [175]. As observed from the table, the D_{KL} and F_1 values obtained on SAGE are optimal for almost all the algorithms, while for CC and MAE, it either performs better or is marginally poorer in performance. Overall, this analysis shows that the proposed SAGE ground truth performs well on a diverse set of algorithms, thus proving its flexibility and robustness.

Next, in Figure 3.5, a cross-evaluation of the proposed method for different driving datasets is performed. For this set of experiments, one algorithm, namely DR(eye)VE [122] is fixed, while the dataset is varied. Two variants of SAGE are evaluated - first, by combining scene semantics with the gaze of [6], and second, with the gaze of [175]. For each of these, the results are compared with the respective gaze-only ground truth of the respective datasets. Like before the performance of predicted saliency maps is evaluated using the same fixation-centric and semantic-centric metrics. The results show that the proposed SAGE models are not strongly tied

to a dataset and can adapt to different driving conditions. It is important to note that even though the cross-evaluation (SAGE-D tested on [175], and SAGE-B tested on [6]) is slightly unfair, the results for SAGE still significantly outperforms those of the respective gaze-only models.

Comparison at important driving scenarios - In Table 3.2, the scenarios of pedestrians crossing at intersections are considered. For this purpose, a subset of the JAAD dataset [82] was used that contained more than five pedestrians (not necessarily as a group) crossing the road. The same four algorithms described in Section 3.3.1 have been reconsidered for this case. Using M-RCNN, the segmented masks of all the crossing pedestrians were computed and the predicted saliency maps from the models were evaluated against this baseline. Upon comparison, it can be seen that models trained on SAGE surpass those trained on the gaze-only ground truth. It is to be noted that even though none of the models were trained on the JAAD dataset [82], the results are still pretty consistent across all the algorithms. This shows that learning from SAGE indeed yields a better saliency prediction model that can detect pedestrians crossing at an intersection more reliably.

Finally, in Table 3.3, another important driving scenario is taken into account where the detections of the number of cars approaching the ego vehicle are considered as a metric. The evaluation set was constructed from different snippets of the DR(eye)VE [6] and the BDD-A [175] datasets, where a single or a group of cars is/are approaching the ego vehicle from the opposite direction in an adjacent lane. Once again, the four algorithms were evaluated on this evaluation set. Like in Table 3.2, here too, the detections were analyzed with respect to those made by M-RCNN. The results from Table 3.3 show that for almost every experiment the performance of algorithms trained on SAGE is consistent in detecting the vehicles more accurately compared to the models trained by gaze-only ground truth.

To summarize, the experiments clearly show that the proposed SAGE ground-truth can be easily trained using different saliency algorithms and the obtained results can also operate well across a wide range of driving conditions. This makes it more reliable for the driving task as compared to existing approaches which only rely on raw human gaze. Overall, the

Table 3.2. Comparison of SAGE with the gaze models for pedestrian crossing at intersection scenario. The clips are taken from the JAAD [82] dataset.

	Fixation-centric metrics				Semantic-centric metrics			
	D_{KL}		CC		F_1 score		MAE	
Model	Gaze gt	**SAGE gt**	Gaze gt	**SAGE gt**	Gaze gt	**SAGE gt**	Gaze gt	**SAGE gt**
DREYEVE [121]	3.36±0.76	**1.56±0.62**	0.19±0.09	**0.55±0.15**	0.07±0.06	**0.21±0.09**	0.08±0.04	**0.07±0.04**
BDDA [175]	2.37±0.78	**1.87±0.81**	0.28±0.16	**0.43±0.16**	0.2±0.13	**0.37±0.17**	**0.09±0.05**	0.12±0.04
ML-Net [27]	2.44±0.58	**2.27±0.67**	0.29±0.11	**0.41±0.15**	0.15±0.07	**0.31±0.13**	0.09±0.04	**0.08±0.04**
PiCANet [95]	2.97±0.68	**1.81±0.72**	0.20±0.11	**0.50±0.14**	0.13±0.07	**0.44±0.16**	**0.07±0.04**	0.11±0.03

Table 3.3. Comparison of SAGE with the gaze models for detecting multiple cars approaching the ego-vehicle from the opposite direction. The clips are taken from the DR(eye)VE [6] and BDD-A [175] datasets.

	Fixation-centric metrics				Semantic-centric metrics			
	D_{KL}		CC		F_1 score		MAE	
Model	Gaze gt	**SAGE gt**	Gaze gt	**SAGE gt**	Gaze gt	**SAGE gt**	Gaze gt	**SAGE gt**
DREYEVE [121]	3.87±0.79	**1.28±0.71**	0.18±0.11	**0.62±0.19**	0.08±0.08	**0.33±0.16**	0.08±0.05	**0.07±0.05**
BDDA [175]	2.95±0.96	**1.92±1.01**	0.19±0.16	**0.42±0.18**	0.14±0.13	**0.34±0.19**	**0.09±0.09**	0.12±0.07
ML-Net [27]	2.72±0.6	**1.94±0.9**	0.21±0.1	**0.5±0.18**	0.12±0.07	**0.37±0.14**	0.09±0.05	**0.08±0.05**
PiCANet [95]	3.17±0.6	**1.69±0.88**	0.18±0.1	**0.55±0.17**	0.12±0.07	**0.49±0.2**	**0.08±0.05**	0.1±0.04

performance of the proposed method is better than gaze-only ground truth on **49/56 (87.5%)** cases, not only when averaged over the entire dataset, but more importantly, in specific driving situations demanding higher focus of attention.

3.4 Conclusion and Future Work

This work introduces SAGE-Net, a novel deep-learning framework for successfully predicting "where an autonomous vehicle should look" while driving, through predicted saliency maps that learn to capture semantic context in the environment, while retaining the raw gaze information. With the proposed SAGE-ground truth, saliency models have been shown to pay attention to the important driving-relevant objects while discarding irrelevant or less important cues, without having any additional computational overhead to the training process. An extensive set of experiments demonstrates that the proposed method improves the performance of existing saliency algorithms across multiple datasets and various important driving scenarios, thus establishing the flexibility, robustness, and adaptability of SAGE-Net. The authors hope that the research conducted in this work will motivate the autonomous driving community into looking at strategies, that are simple but effective, for enhancing the performance of currently existing algorithms.

A possible future work will involve incorporating depth in the SAGE-ground truth and then having the entire framework trained end-to-end. Currently, this could not be achieved due to low variance in the depth data, leading to overfitting. Another possible direction that is being considered is to explicitly add motion dynamics of segmented semantic objects in the surroundings in the form of SegFlow [24]. Work in this area is in progress as a campus-wide dataset is currently being built with these kinds of annotations through visual sensors and camera-LiDAR fusion techniques.

3.5 Supplementary Material

3.5.1 Appendix A: Derivation of β for F-score

For saliency prediction in driving, False Negatives (FN) are more of a concern as compared to False Positives (FP). This is because it is probably still fine to detect a pedestrian, even if they are not crossing the road anytime soon. On the contrary, it is a much bigger cost to not detect a person crossing. Thus, the metrics need to be tuned to penalize FN more in comparison to FP. As discussed in the paper, D_{KL} and CC already do that. Here, the derivation of F-score in terms of its hyper-parameter β is provided. It is known that:

$$\text{Precison} = \frac{\text{True Positive } (TP)}{\text{True Positive } (TP) + \text{False Positive } (FP)} \tag{3.4}$$

and,

$$\text{Recall} = \frac{\text{True Positive } (TP)}{\text{True Positive } (TP) + \text{False Negative } (FN)} \tag{3.5}$$

Now, the F-score is given by:

$$\text{F-score } (\beta) = \frac{(1+\beta^2) * \text{Precision} * \text{Recall}}{\beta^2 * \text{Precision} + \text{Recall}} \tag{3.6}$$

Replacing for Precision and Recall from 3.4 and 3.5 respectively,

$$\text{F-score } (\beta) = \frac{(1+\beta^2) * \frac{TP}{TP+FP} * \frac{TP}{TP+FN}}{\beta^2 * \frac{TP}{TP+FP} + \frac{TP}{TP+FN}} \tag{3.7}$$

$$= \frac{(1+\beta^2) * TP}{\beta^2 * (TP+FN) + (TP+FP)} \tag{3.8}$$

$$= \frac{(1+\beta^2) * TP}{(1+\beta^2) * TP + \beta^2 * FN + FP} \tag{3.9}$$

In equation 3.9, it can be clearly seen that the numerator has no FP or FN terms and they are present only in the denominator in the additive form. Thus it can be concluded that with an

increase in FP or FN, the F-score (β) value decreases. That is,

$$\text{F-score } (\beta) \downarrow = \frac{(1+\beta^2) * TP}{(1+\beta^2) * TP + \beta^2 * FN \uparrow + FP} \tag{3.10}$$

$$\text{F-score } (\beta) \downarrow = \frac{(1+\beta^2) * TP}{(1+\beta^2) * TP + \beta^2 * FN + FP \uparrow} \tag{3.11}$$

Also in equation 3.9, it can be seen that FN has a weight of β^2 and FP has a weight of 1. Thus, when β^2 is lower than 1, FN has smaller influence on F-score (β) compared to FP , and when β^2 is greater than 1, FN has greater influence on F-score (β) compared to FP. As discussed in the paper and above, FN is more dangerous compared to FP for autonomous driving tasks, and thus the value of β^2 must NOT be chosen lower than 1. Therefore, to give equal weightage to FN and FP, it was decided to keep β^2 equal to 1.

3.5.2 Appendix B: Algorithm description

In Table 3.4, the hyperparameters for each model are mentioned. Additionally, in Table 3.5, the details of the architecture of each of the four algorithms that were considered are provided. All four models were trained on both the gaze-only data and the proposed SAGE data. The hyperparameters were kept the same during both the training process.

3.5.3 Appendix C: Miscellaneous

In Figure 3.5, a cross-evaluation experiment was conducted where two variants of SAGE were considered and compared with the respective gaze-only ground truths. In Table 3.6, a similar result is shown where two variants of the gaze-only results are considered along with that of SAGE. As seen from the results, SAGE outperforms the former in almost every case.

Table 3.4. Summary of Hyperparameters

Parameters	**DR(eye)VE** [122]	**BDD-A** [175]	**ML-Net** [27]	**PiCANet** [95]
Input image size	448×448	576×1024	480×640	224×224
Initial Learning rate	0.0001	0.001	0.001	0.001
Learning rate decay	-	-	0.0005/step	0.1/7000 steps
Non-Linearity in feedforward network	ReLU, Leaky ReLU (α=0.001)	ReLU	ReLU	ReLU
Training Loss function	K-L Divergence	Cross-Entropy	K-L Divergence	Binary Cross-Entropy
Optimizer	Adam	Adam	SGD	SGD
Batchsize	8	10	8	4
#Training Epochs	20	20	32	20

Table 3.5. Network Architectures

Algorithms	Model Architecture
DR(eye)VE [122]	COARSE: 6 layer 3D ConvNet (C3D architecture) with Bilinear Upsampling REFINE: 5 layer 2D ConvNet (for resized input), 1 layer 2D ConvNet (for cropped input)
BDD-A [175]	AlexNet feature extractor + Upsampling + 3 layer 2D ConvNet (visual processing) + Conv2D-LSTM (temporal processing)
ML-Net [27]	Feature Extraction Network: 13 layer Fully Convolutional Network (FCN) Encoder Network: 1 layer 2D ConvNet Decoder Network: Bilinear Upsampling
PiCANet [95]	Encoder Network: 16 layer FCN (VGG-16) Decoder Network: 6 layer Deconvolution with bilinear interpolation

Table 3.6. Comparison of SAGE with two variants of the gaze truth.

	Fixation-centric metrics						Semantic-centric metrics					
	D_{KL}			CC			F_1 score			MAE		
Dataset	DR(eye)VE gt	BDD-A gt	**SAGE gt**	DR(eye)VE gt	BDD-A gt	**SAGE gt**	DR(eye)VE gt	BDD-A gt	**SAGE gt**	DR(eye)VE gt	BDD-A gt	**SAGE gt**
DR(eye)VE	2.02±0.47	2.26±0.55	**1.67±0.41**	0.48±0.1	0.45±0.11	**0.55±0.11**	0.17±0.09	0.13±0.05	**0.36±0.09**	0.07±0.03	0.07±0.03	0.07±0.03
BDDA	1.74±0.43	1.28±0.43	**0.73±0.38**	0.42±0.14	0.58±0.13	**0.75±0.13**	0.09±0.06	0.1±0.06	**0.37±0.14**	0.12±0.06	0.11±0.06	**0.08±0.05**

3.6 Acknowledgements

Chapter 3, in part, is a reprint of **A. Pal**, S. Mondal and H. I. Christensen, "Looking at the right stuff - Guided semantic-gaze for autonomous driving", in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020*. The dissertation author was the primary author of this paper.

Chapter 4

Object-goal navigation utilizing hierarchical relationship

Human beings can perform complex tasks such as object-goal navigation efficiently, with implicit memorization of the relationships between the different objects. For example, to navigate to a target such as a `toaster` in the kitchen, a natural thing to do is to start by looking around a set of larger candidate objects that are likely to be nearby, such as a `stove` or `microwave`. Unfortunately, this type of *hierarchical relationship* is rarely used in robot navigation. As a result, the agent usually fails to develop any intuition about the goal location when they are further away. In this chapter, the object goal is referred to as the `target` objects, and those larger candidates that have either a spatial or semantic relationship with the target are called `parent` objects. This is depicted in Figure 4.1. Two main challenges are addressed here - (i) Correctly associating the robot's current observation with some prior intuition about object relationships into the model. (ii) Efficiently utilizing this hierarchical relationship for the visual navigation problem.

Existing research in this area tends to aggregate sensory input into a meaningful state, before sending it to a Reinforcement Learning (RL) framework, with the expectation that the robot can implicitly learn the navigation problem through recursive trials. Zhu *et al*.[188] proposed to solve this problem by finding the similarity between the current observation and the target observation through a trained Siamese Network [74]. The work of Wortsman *et al*.[171]

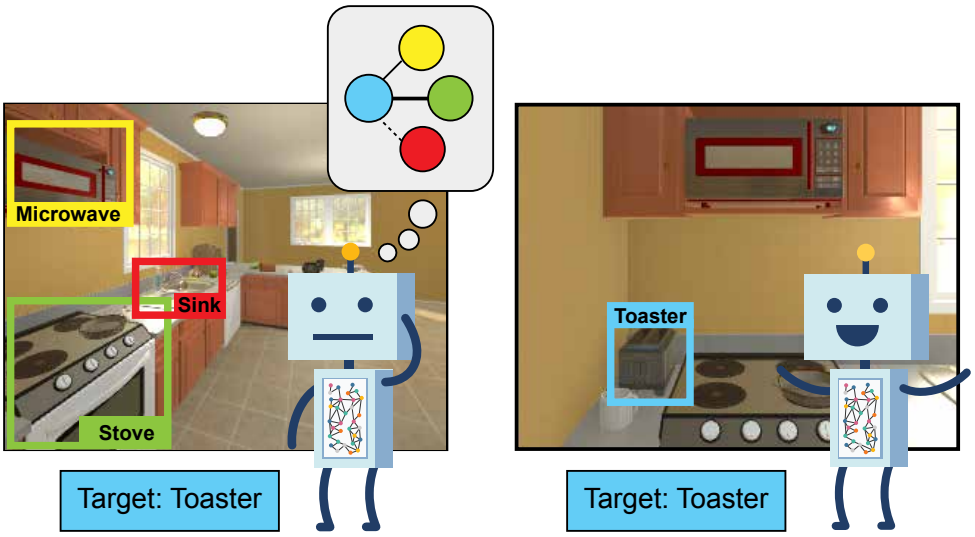


Figure 4.1. Illustration of the parent-target relationship. Upon seeing a set of parent objects (left image), the agent learns to associate the correct parent (`Stove` or `Microwave`, here) to the target object (`Toaster`, here) from the knowledge graph and tries to search in its neighborhood to successfully locate the target object (right image). [Best when viewed in color]

incorporated a meta-learning approach where the agent learns a self-supervised interaction loss during inference to avoid collision. However, none of these methods use any prior information or semantic context. The work by Sax *et al.*[139] highlights that a set of mid-level visual priors such as depth and edge information, surface normals, key points, etc. can be useful for the navigation task. However, a different encoder is required for learning each of these representations, and thus, the model does not scale very well. Moreover, the learned features are expected to vary with the test setting, thereby limiting the scope of the method in unseen dissimilar environments. Instead, using the knowledge about object relationships is more robust since they are domain-independent. Two approaches that are similar to that proposed here, are from Yang *et al.*[180], and Druon *et al.*[35]. In [180], semantic priors in the form of a knowledge graph are used to capture object-object connectivity, but this connectivity is defined only in terms of their spatial proximity, not the inter-object dependency. For instance, an object such as `pillow` in the bedroom might be visible next to both a `bed` and an `alarmclock`. Yet, one knows from experience that to find a `pillow`, one should always start by looking around the `bed`. This type of common-sense knowledge is missing from their method. In contrast, the proposed approach in this work assigns a sub-goal

reward to force the agent to learn this key hierarchical relationship. Also, important information about an object's spatial location in the scene is missing as they learn a single context for the whole scene. Instead, this work utilizes an object detector to capture this information explicitly. Recently, Druon *et al*.[35] introduced the concept of a *context grid* where they modeled the spatial (through an object detector) and language (through a word-embedding) similarity between the target and other objects as a 16×16 grid. However, this relationship does not update during the training stage, thereby making the model less adaptable. Additionally, their action space is much bigger, thus simplifying the navigation task.

A major issue with RL algorithms in a real-world setting is efficient modeling of the continuous high-dimensional state space in the agent's surroundings [38]. While implementing other algorithms [188, 180], it was observed that even though the convergence during the training stage was fast, they gave a relatively poor performance during testing. A possible reason for this is that due to their non-representative state space, the agent is perhaps simply learning to memorize the training set after a certain number of episodes rather than understanding the underlying object relationships, and thus it fails to generalize to a novel scene. The adaptive gradient of Wortsman *et al*.[171] alleviates this problem to some extent. Nevertheless, it doesn't remove it completely since it does not take any prior memory into account. In contrast, the representative state in this work leverages the correct balance of prior context and current observation to provide sufficient information about the surroundings during training, while simultaneously being abstract enough to generalize to a different room layout during testing.

Several contributions are made in this work. Firstly, a hierarchical object relationship learning approach is presented for the object-goal navigation problem by understanding the role of semantic context. This is done through the proposed novel *Context Vector* as a node embedding in the graph convolutional neural network. Secondly, the role object detection plays as opposed to a scene representation from traditional image classification networks is also emphasized. Finally, the mentioned parent-target object relationship is incorporated through a new reward-shaping mechanism.

4.1 Background

Map vs map-less navigation approaches - Traditional approaches in visual navigation involved formalizing this as an obstacle avoidance problem, where the agent learns to navigate in its environment through a collision-free trajectory. This is done either in the form of offline maps [11, 70, 118], or online maps [29, 146, 155, 170] generated through Simultaneous Localization and Mapping (SLAM) techniques. Given this map as input, the typical approach was to employ a path planning algorithm such as A* [55] or RRT* [69] to generate a collision-free trajectory to the goal. The limitation of these algorithms is that it might not be possible to have a pre-computed map of the environment. Generating a rich semantic map online is also a non-trivial task. With the advent of deep learning, the focus has instead shifted towards methods that are map-less [23, 54, 47, 93, 136], meaning that the representations can be learned over time through interactions. However, most of these algorithms are not suited for finding specific target objects in a previously unseen environment. In contrast, the proposed approach solves the target-driven navigation problem entirely using only visual inputs without the need for a pre-computed map.

Point-goal navigation vs object-goal navigation - Point-goal navigation [7] refers to the problem where an agent starts from a randomly chosen pose and learns to navigate to a specific target point, usually specified in terms of 2D/3D coordinates relative to the agent. Lately, there has been some research [110, 75] in this area. However, it is still an ill-defined problem in a realistic setting, thereby making comparison difficult [138]. Object-goal navigation refers to the problem where the agent instead learns to navigate to a specified target object while successfully avoiding obstacles. These tasks usually require some prior knowledge about the environment which can be useful for navigation [188, 171, 180, 35]. The work proposed here approach falls in this category but involves learning a robust contextual object-object relationship.

Role of learning semantic context - Learning the semantic context of the surrounding world is an important research topic in the computer vision and robotics community. However, most of the existing work [157, 59, 130, 112, 143, 106] surrounds static settings such as object

detection, semantic segmentation, activity recognition, etc. Recently, object-object relationship modeling has been studied for tasks such as image retrieval [67] using scene graphs, visual relation detection [185], visual question-answering [66, 105], place categorization [118], and driver saliency prediction [117]. This work proposes a novel algorithm that successfully learns to exploit hierarchical object relationships for object-goal visual navigation.

Reward shaping for policy networks - Reward shaping is a method in reinforcement learning for engineering a reward function to provide more frequent feedback on appropriate behaviors [166]. In general, defining intermediate goals or sub-rewards for an interaction-based learning algorithm is non-trivial, since the environment model is not always known. However, a divide and conquer approach is often imperative to exploit the latent structure of a task to enable efficient policy learning [153, 8, 50]. Specifically, for the object-goal visual navigation problem, it is important to learn the inherent "parent-target" object relationship for providing meaningful feedback to the end-to-end training. For the policy network, the Asynchronous Advantage Actor-Critic (A3C) [111] algorithm is used to sample the action and the value at each step as per the approach of other models [188, 171, 180].

4.2 Task Definition

The object-goal navigation problem aims to find a target object, defined through a set $T = \{t_1, \ldots, t_N\}$, in a given environment. The problem is defined purely from a vision perspective, and therefore any information about the environment in the form of a semantic or a topological map is not provided. The agent is spawned at a random location in the environment at the beginning of an episode. The input to the model is current observation in the form of RGB images, and the target object's word-embedding, rolled over each time-step. Using this, the agent has to sample an action a from its trained policy given a set of actions A, where $a \in A = \{$`MoveAhead`, `RotateLeft`, `RotateRight`, `LookUp`, `LookDown` and `Done`$\}$. The `MoveAhead` action takes the agent forward by 0.25 meters, while the `RotateLeft` and `RotateRight` actions rotate it by

45 degrees. Finally, the `Look` action tilts the camera up/down by 30 degrees. An episode is considered a "success", if the target object is `visible`, meaning the agent can detect it in the current frame, and is within a distance of 1.5 meters from it. When the "`Done`" action is sampled by the agent, the episode ends and the model checks if this criterion is met.

4.3 Memory-utilized Joint hierarchical Object Learning for Navigation in Indoor Rooms (MJOLNIR)

Parent-Target object relationship - In addition to the target objects, a new set of object classes is introduced, defined by the set $P = \{p_1, \ldots, p_M\}$. These "parent objects" consist of the larger objects present in a room, which also happen to be spatially/semantically related to the target object. For example, `CounterTop` is a parent object in the `Kitchen` and `Bathroom` scenes, while `Shelf` is a parent object in `Living room` and `Bedroom`. The set of parent objects, P, is manually picked for each room based on the strong correspondences they have with the target object list, T, in the knowledge graph (explained below). The navigation agent aims to start by exploring the area around $p \in P$, eventually leading to the target object $t_i \in T$.

Construction of Knowledge Graph and the Context vector - Similar to [180], the proposed knowledge graph is also constructed using the objects and relationships extracted from the image-captions of the Visual Genome (VG) dataset [83]. However, by pruning a lot of the object (for instance, "`armchair`" vs "`arm chairs`") and relationship (for instance, "`near`" vs "`next to`") aliases, it is possible to build a cleaner adjacency matrix for the graph convolution network, containing strong relationship correspondences between those VG objects, which also appear in the current experimental setting.

In addition to the newly constructed graph, a novel *context vector* $\mathbf{c}_j$ for each object $o_j \in O$ is also introduced, where O is the list of all the 101 objects in the environment. This 5-D vector gives information regarding the state of o_j in the current frame and can be represented as $\mathbf{c}_j = [b, x_c, y_c, bbox, CS]^T$. The first element, b, is a binary vector specifying whether o_j can be

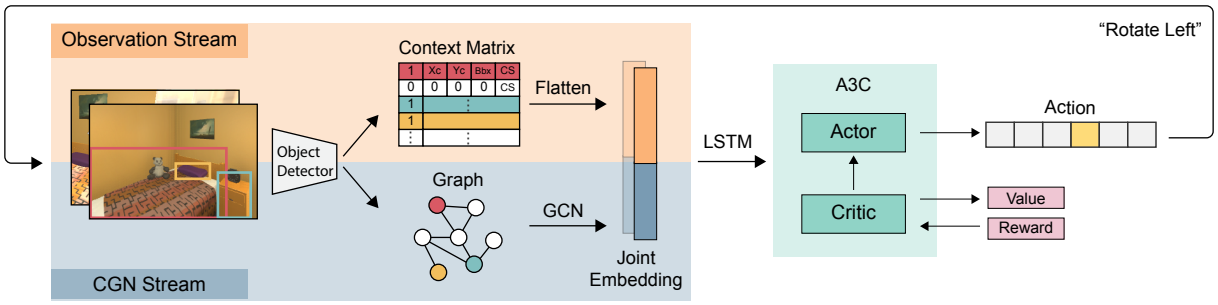


Figure 4.2. The entire MJOLNIR architecture. In the Observation stream, the ground-truth object detector is used to construct the context vector for all objects in the environment. This observation vector (orange) is concatenated with the graph embedding (blue) from the CGN stream to form a joint embedding. This is then sent to an LSTM cell and finally fed to the A3C model. [Best when viewed in color]

detected in the current frame. The next two elements, (x_c, y_c), and *bbox* correspond to the center (x, y) coordinates of the bounding box of o_j, and its covered area, both normalized with respect to the image size. Finally, *CS* is a number giving the cosine similarity between the respective word embeddings of o_j, and the target object $t \in T$. This is expressed as:

$$CS(\mathbf{g}_{o_j}, \mathbf{g}_t) = \frac{\mathbf{g}_{o_j} \cdot \mathbf{g}_t}{||\mathbf{g}_{o_j}|| . ||\mathbf{g}_t||}$$

where **g** denotes the word embeddings in the form of GloVe vectors [125].

End-to-end model - The entire network for the task is shown in Figure 4.2. A two-stream network approach is adopted, consisting of (i) the Observation stream, which encodes the agent's current observation in the environment, and (ii) the Contextualized Graph Network (CGN) stream, which embeds the prior memory obtained through a knowledge graph $G = (V, E)$.

For the Observation stream, two variants have been tried out - (i) The ResNet-18 [57] `conv` features obtained from the current frame are used to give a holistic representation of the scene. This feature map is then combined with the target object word embedding using point-wise convolution and flattened to obtain the observation vector. (ii) The second variant is to replace the `conv` features with the 5-D *context vector* (described above) for every object in the environment. The resulting *context matrix* $\in R^{|O| \times 5}$ is then flattened and forms the observation

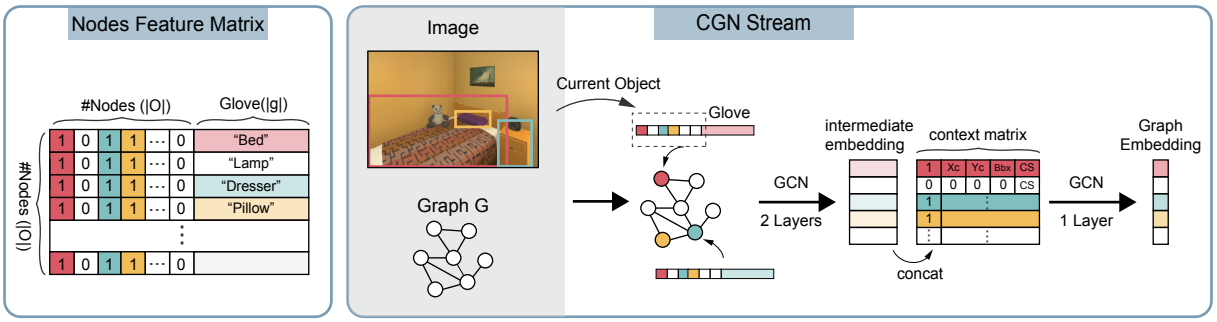


Figure 4.3. The novel CGN architecture. The input node feature constitutes the current detected object list and the node object's glove embedding. This is passed through two layers of GCN. This intermediate embedding is concatenated with the context vector and another layer of GCN is trained on it. [Best when viewed in color]

vector.

Even though the input knowledge graph, G, provides strong initial guidance to the agent, this information, by itself, can be insufficient due to the domain difference between VG [83] and the target navigation environment. The CGN stream is used to diminish this gap. A graph convolution network (GCN) [71] is used to learn the node embeddings. Given G, the node feature vector $X \in R^{|O|+|\mathbf{g}|}$ is designed which is given by the concatenation of the output of an object detector on the current frame, specified by the $|O| = 101$-dimensional vector having 1 for the current frame objects, and 0 for others, along with each node object's word-embedding. "101" is the length of the simulator's exhaustive list of object types, which is analogous to the number of object classes in a trained object detector. This is different from the 1000-dimensional class probability used by Yang *et al.*[180]. The reason for this change is two-fold - (i) the probability vector obtained from ResNet-18 [57] is pre-trained on the 1000 classes of ImageNet [31], which are quite different from the object list present in the environment, and (ii) since the pre-trained network primarily learns to classify a single object, in the multi-object setting, it is more likely to generate noisy labels[1]. The combined input node features are passed through two layers of GCN, to generate the intermediate embedding. This new feature is then concatenated with the *context vector* and then fed to another layer of GCN to generate the final graph node embedding. The

[1]An illustration of this is shown in the supplementary videos in Appendix A

concatenation of the observation vector and the graph embedding results in the joint embedding (shown in Figure 4.2), which is the input to the A3C model. The CGN stream is detailed in Figure 4.3.

To separately highlight the contribution of the changes made to each of the streams, two variants of the algorithm are presented. MJOLNIR-r uses ResNet and word embedding in the current observation stream while using CGN as the graph stream. MJOLNIR-o uses the flattened *context matrix* as the observation vector, along with the CGN stream.

Reward - The reward function is tuned to correctly learn to utilize the parent-target relationship for the navigation task. The agent in the model receives a "partial reward", R_p, when a parent object $p \in P$ is `visible`. This is given by $R_p = R_t * Pr(t|p) * k$, where R_t is the target reward and $k \in (0,1)$ is a scaling factor. $R_t = 5$ and $k = 0.1$ are chosen for the experiments. $Pr(t|p)$ is taken from the partial reward matrix M [2], where each row has the probability distribution of the relative "closeness" of all the parent objects, to a given target object. The closeness is defined based on the relative spatial distance (measured in terms of the L2 distance) between a pair of objects in the floorplans. If multiple parent objects are `visible`, only the object with the maximum R_p is chosen. Moreover, the agent does not get this reward the next time it sees the same parent object. This encourages it to explore different parent objects in the room until the target is located. If the "`Done`" action is sampled, *i.e.* the termination criterion occurs, and t is `visible`, the agent gets the goal reward, which is the sum of R_t and R_p. In this way, it learns to associate parent objects with a target, as well as the current state. Since the entire network is trained end-to-end, this shaped reward propagates back to the GCN layers, and tunes them to correctly learn the "parent-target" hierarchical relationship from the input knowledge graph. Finally, if neither the parent nor the target object is `visible`, the agent gets a negative step penalty of 0.01. The reward for the state s, and action a, is therefore given by:

[2] Please refer to Appendix B of the supplementary for details about the construction of M.

$$R(s,a) = \begin{cases} R_p, & \text{if p is } \texttt{visible} \\ R_t, & \text{if t is } \texttt{visible} \text{ at termination} \\ R_t + R_p, & \text{if both are } \texttt{visible} \text{ at termination} \\ -0.01, & \text{otherwise} \end{cases}$$

Algorithm 2 summarizes the above explained process.

Algorithm 2: Reward Shaping for MJOLNIR

Input: state s, action a, target $t \in T$, `SeenList`
Data: target reward R_t, partial reward matrix M
Function `Judge(`s, a, t`)`**:**
  if $a \neq$ "`DONE`" **then**
    reward = `Partial` (s,t)
  else if $a ==$ "`DONE`" and t is `visible` **then**
    `SeenList` = [];
    reward = R_t + `Partial` (s,t);
return reward;
Function `Partial(`s,t`)`**:**
  foreach *parent* $p_i \in P$ **do**
    if p_i *is* `visible` and $p_i \notin$ `SeenList` **then**
      $p \leftarrow argmax\ M[t]$;
      `SeenList` $\leftarrow p$;
      $R_p = M[t][p] * R_t * k$
  end
return R_p;

4.4 Experiments and Results

4.4.1 Experimental setting

The AI2-THOR (The House Of inteRactions) [78] simulator is used as the environment for the navigation tasks. It is a challenging simulation platform, consisting of 120 photo-realistic floorplans categorized into 4 different room layouts - `Kitchen`, `Living room`, `Bedroom`, and `Bathroom`. Each scene is populated with real-world objects that the agent can observe and interact with, thereby enabling algorithms trained here to be easily transferable to real-robot

settings. Out of the 30 floorplans for each scene layout, the first 20 rooms from each scene type are considered for the training set, and the remaining 10 rooms as the test set for the experiments. The list of target and parent objects can be found in Appendix C of the supplementary.

4.4.2 Comparison Models

Random - In this model, at each step, the agent randomly samples its actions from the action space with a uniform distribution. **Baseline** - This model closely resembles that of Zhu *et al*.[188], as it comprises the current observation (in the form of the ResNet features of the current RGB frame) and the target information (in the form of glove embedding of the target object) as its state. **Scene Prior** (SP) - The publicly available implementation of Yang *et al*.[180] has been used here. This uses the prior knowledge in the form of a knowledge graph but does not utilize the hierarchical relationships between objects. **SAVN** - In this model [171], the agent keeps learning about its environment through an interactive loss function even during inference time.

Metrics - For fair comparison with other state-of-the-art algorithms, the evaluation metrics proposed by [7] are used. This is consistent with the metrics adopted by other target-driven visual navigation algorithms [188, 171, 180, 35]. The Success Rate (SR) is defined as $\frac{1}{N}\sum_{i=1}^{n} S_i$, while the Success weighted by Path Length (SPL) is given by $\frac{1}{N}\sum_{i=1}^{n} S_i \frac{l_i}{\max(l_i,e_i)}$. Here, N is the number of episodes, and S_i is a binary vector indicating the success of the i-th episode. e_i denotes the path length of an agent episode, and l_i is the optimal trajectory length to any instance of the target object in a given scene from the initial state. The performance of all the models is evaluated on the trajectories where the optimal path length is at least 1 ($L >= 1$), and at least 5 ($L >= 5$).

4.4.3 Implementation details

The proposed model is built on the publicly available code of [171], using the PyTorch framework. The agent was trained for 3 million episodes on the offline data from AI2-THOR

Table 4.1. Comparison with state-of-the-art visual navigation algorithms on the unseen test set

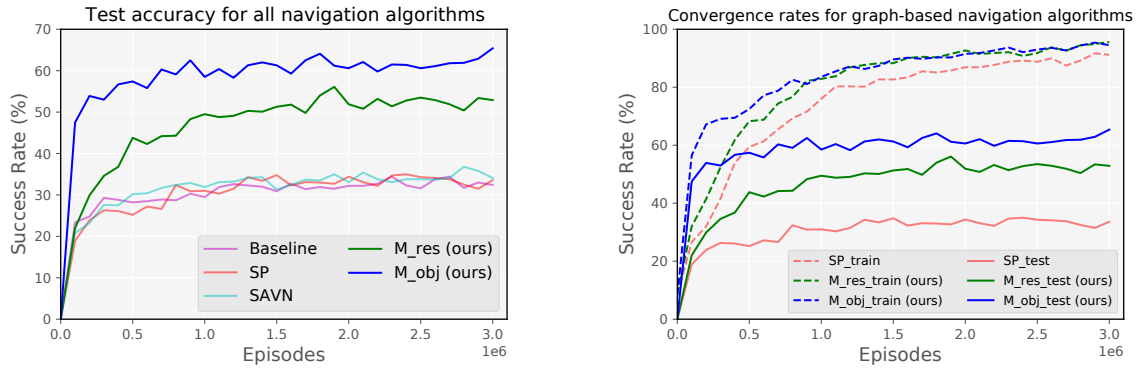
	$L >= 1$		$L >= 5$	
Model	SR(%)	SPL(%)	SR(%)	SPL(%)
Random	11.2	5.1	1.1	0.50
Baseline [188]	35.0	10.3	25.0	10.5
Scene-prior [180]	35.4	10.9	23.8	10.7
SAVN [171]	35.7	9.3	23.9	9.4
MJOLNIR-r (our)	54.8	19.2	41.7	18.9
MJOLNIR-o (our)	**65.3**	**21.1**	**50.0**	**20.9**

*v*1.0.1 [78]. During evaluation, 250 episodes were used for each of the 4 room types, resulting in 1000 episodes in total. In each episode, the floorplan, target, and initial agent position were randomly chosen from the test set defined in Section 4.4.1. Additional implementation details can be found in Appendix D of the supplementary material.

4.4.4 Results

Table 4.1 and Figure 4.4a show the performance of each of the models on the test environments. It is to be noted that the test environments consist of rooms that the agent has not previously seen during training, and therefore the location of the different objects is completely unknown. It can be seen that both the proposed models significantly outperform the current state-of-the-art. MJOLNIR-o has a 82.9% increase in SR. This supports the hypothesis that incorporating *context vector* into the observation state is indeed a better idea than directly using ResNet and GloVe features. This is because the semantic information extracted from the scene in this manner is more object-centric, thereby making the target-driven navigation problem easier. It is also important to note here, that the graph convolutional network of [180], which does not include the *context vector* as its node feature, performs poorer than even the baseline [188] and SAVN [171].

It is interesting to note that even though MJOLNIR-r cannot beat the performance of



(a) Test accuracy for all models

(b) Convergence rate for graph-based models

Figure 4.4. Test accuracy and convergence rates for all the algorithms

MJOLNIR-o, it is still a substantial improvement over the other state-of-the-art methods (an observed gain of 53.5%). This highlights the importance of the proposed CGN stream which can better capture the contextual information extracted via ResNet `conv` features from the current input image. Moreover, using reward shaping to tune the model parameters ensures that not only is the prior memory preserved, but the current information containing the parent-target object relationship hierarchy is also utilized.

Figure 4.4b shows the convergence rate of MJOLNIR with the model from Yang *et al.*[180]. It can be seen that the training and testing SR of the proposed algorithms rapidly increases within the first 5 million episodes itself, before saturating. This shows that the presented models learn to correctly locate targets much faster than others. In contrast, for [180], even though the training SR is quite high ($\approx 90\%$), there is a huge drop during the testing performance ($\approx 35\%$), signifying severe overfitting. Finally, Table 4.2 gives the comparison of room-wise results.

4.4.5 Ablation study

A number of ablations on MJOLNIR are shown in Table 4.3. In MJOLNIR-o (no_g), the third graph convolution layer of MJOLNIR-o where the intermediate embedding was concatenated with the *context vector* is removed. Instead, here, the intermediate embedding is directly

Table 4.2. Evaluation results on a per-room basis

Models	Bathroom		Bedroom		Kitchen		Living Room		Average	
	SR(%)	SPL(%)	SR(%)	SPL(%)	SR(%)	SPL(%)	SR(%)	SPL(%)	SR(%)	SPL(%)
Baseline [188]	53.2	13.4	28.8	9.0	32.4	10.9	35.2	10.0	37.4	10.8
Scene Prior [180]	41.6	13.3	33.6	10.4	26.4	9.1	36.0	9.9	34.4	10.7
SAVN [171]	47.6	14.6	21.6	6.7	34.8	8.3	40.0	9.0	36.9	9.7
MJOLNIR-r (our)	72.8	24.3	41.2	**16.9**	56.4	21.2	**50.8**	15.9	55.3	19.6
MJOLNIR-o (our)	**82.4**	**25.1**	**43.2**	14.4	**74.8**	**22.9**	50.0	**17.9**	**62.6**	**20.1**

Table 4.3. Ablation study for MJOLNIR

DONE action	Model	$L >= 1$		$L >= 5$	
		SR(%)	SPL(%)	SR(%)	SPL(%)
only sampled	MJOLNIR-r	54.8	19.2	41.7	18.9
	MJOLNIR-o	**65.3**	21.1	**50.0**	**20.9**
	MJOLNIR-o (no_g)	59.0	16.6	41.0	16.9
	MJOLNIR-o (w)	64.7	**21.6**	46.4	20.6
sampled + env	SAVN [171]	54.4	35.55	37.87	23.47
	MJOLNIR-o	**83.1**	**53.9**	**71.6**	**36.9**

fed to the joint embedding. The slightly lower performance shows that the context vector can indeed help in learning meaningful node embeddings for the graph. In MJOLNIR-o (w), a weighted adjacency matrix is used for the graph convolutional network. This does not affect the performance significantly as the authors believe that the object-object relationship is inherently learnt by MJOLNIR-o.

Another evaluation of the proposed model was performed with a different stopping criteria. In this case, the agent does not rely only on its sampled "DONE" action to learn the termination action. Instead, it stops even when the environment gives the signal that the target object has been found. For this, the proposed model is compare with the SAVN [171] model using this stopping criteria. The results show that the presented approach performs significantly better.

Some failure case analysis provides insights into the proposed models, and also opens up research in further directions. Mainly two such cases are identified. Firstly, the agent gets stuck at a particular state, when the target is visible, but the straight path to it blocked by some obstacle. The authors hypothesize that a planner module which checks for collision might help to overcome this. Secondly, since "DONE" is one of the candidate actions, (i) it may be wrongly sampled even when the goal hasn't been reached, or (ii) it might not be sampled even after reaching the goal. A workaround is to have the termination criterion provided by the environment.

As seen from the ablation study, this boosts the success rate from 65.3% to 83.1%. However, it is at the cost of increased episode length as the agent is encouraged to explore more of the environment. Moreover, in a real-robot setting, it might not be possible to have the environment signal the termination.

4.5 Case study: A deeper look on the role of reward shaping

As mentioned in Section 4.1, reward shaping for reinforcement learning is a way to provide localized signals to an agent for encouraging behavior that is consistent with prior knowledge [85]. For the task of indoor robot navigation in search of a target object of interest, an agent needs to obtain intermediate auxiliary signals based on surrounding objects, to ensure that it's heading towards the goal. This is especially true for large environments, where the robot may need to take a number of steps to reach the goal [119]. A popular reward function used in the object-goal navigation literature [188, 180, 171, 36, 37] is binary, where a large positive reward is given at the goal state, while a smaller negative step penalty is assigned for every other state. Unfortunately, this type of signal is quite sparse, thereby discouraging the learning process.

An alternate approach that has gained interest [22, 103] is to use geodesic distance to the closest target as a reward signal. Although this is a denser function compared to the binary reward, absolute knowledge about the closest distance to the goal is a strong assumption that may not be easily available outside certain simulation environments [138]. An alternate approach to this is to propose a method that relies on the estimated distance to objects calculated via different heuristics. Two approaches that are similar to this are that of Druon *et al.*[35] and Ye *et al.*[181]. They both provide auxiliary signals based on the bounding box area of objects. However, these rewards are only assigned to the target object, and therefore, the signals are still quite sparse, especially when targets are smaller in size.

To expand on this idea, one can build on the initial approach described in [119] by defining distance-based heuristics to modify the reward for both target objects and other large,

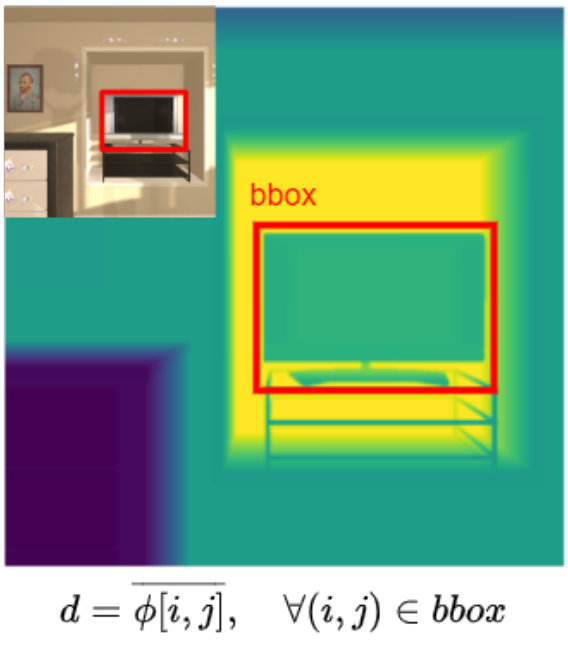
salient objects that have a close relationship with the target (called parent objects).

4.5.1 Methodology

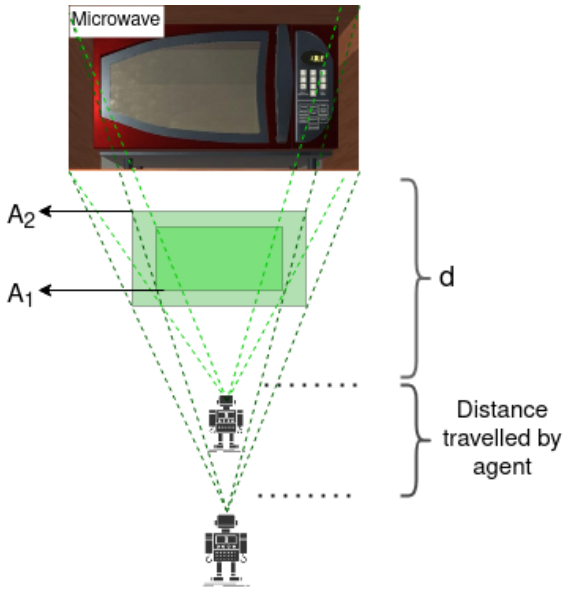
Pal *et al.*[119] introduced a reward shaping mechanism where the agent receives a "partial" reward, R_p when it can identify a parent object with a close relationship to the target. This is given by $R_p = R_t * Pr(t|p) * k$, where R_t is the target reward, and $Pr(t|p)$ is a probability distribution of the relative "closeness" of all the parent objects, p, to a given target object, t. Additional details can be found in [119]. Notably, in that work, the scaling factor, k, is a constant kept fixed at 0.1. Therefore, the partial reward is independent of the distance between the agent and the parent/target objects, d. Moreover, R_p was only provided when the agent is within a distance threshold from the parent (set as 1m in [119]). To overcome this issue, *two* methods have been proposed to address by reformulating k as a factor of d. Furthermore, the R_p formulation is extended towards both parent and target objects. The primary motivations for this are: (i) the agent should be encouraged to identify parent objects whenever they are visible, and (ii) by making the reward a factor of d, the agent is further inspired to explore regions closer to p.

(i) Utilizing metric depth - The first approach involves using metric depth in the form of depth maps obtained directly from the AI2-THOR simulator [77]. Instead of this, an RGB-D sensor can also be used to get the estimated depth. From the depth maps, the metric depth d is computed as the average value of the region, ϕ, bounded by an object's bounding box. This is illustrated in Figure 4.5a. Subsequently, the scaling factor is formulated as a linear function, $k'(d) = k * (m * d + c)$. In the experiments, $m = -0.15$, and $c = 1$ were empirically chosen to ensure $k' \in [0, 1]$.

(ii) Utilizing bounding box area - While the metric depth approach is intuitive, in theory, it is observed that due to the added sensor input in the form of depth maps, the training time increases. Thus, the next approach that was tried was to use a heuristic for relative distance, where the scaling factor is calculated based on the assumption that as the agent moves closer, an object's bounding box (bbox) area should proportionately increase. This method,



(a) Metric distance from depth maps



(b) Relative distance from bbox area

Figure 4.5. The image on the left shows a depth map with a bounding box around the object. The inset contains the RGB image of the object. d is obtained by finding the average distance of each pixel in the bounding box. The image on the right shows the relative increase in the bounding box area of an object (A_1 to A_2) as the agent moves closer. d is object distance when the area is A_2.

apart from being simple to implement, also reduces the dependence on additional sensor data, thereby minimizing the computational load. For this strategy, the scaling factor is given by $k'(d) = k * (1 - (A_1/A_2(d))^{0.5})$, where A_1 and A_2 are bounding box areas of a particular object in the state when it was first seen by the agent and the current state respectively. This is depicted in Figure 4.5b.

A visualization of the reward distribution is provided in Figure 4.6. On the left side, the binary reward r_{bin} is shown, which only activates within a region of the target object, and is absent everywhere else. In the middle, the baseline partial reward from Pal *et al.*[119] is shown, which increases the region of reward to the large "parent" objects near the target objects. It is to be noted that even with the increased coverage, this reward is still not continuous. Finally, on the right, the dense reward from Madhavan *et al.*[101] is shown, which is a continuous function of

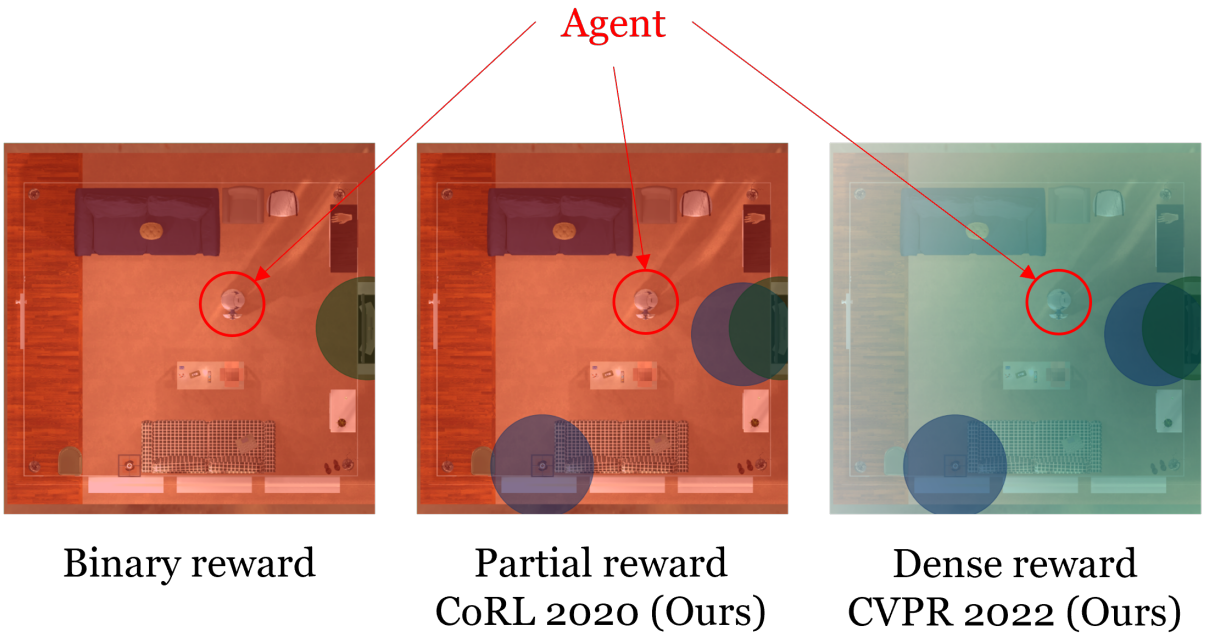


Figure 4.6. Distribution of the three types of reward functions mentioned in this work. [Best when viewed in color]

the distance from the object.

4.5.2 Experiments and Results

Similar to Pal *et al.*[119], the AI2-THOR [77] environment was used for these experiments. The setup and train/test split is consistent with other standard methods - GCN [180], SAVN [171], and MJOLNIR-O/R [119]. The agent was trained for 3×10^6 episodes for each model. Furthermore, for every model, experiments using 4 different reward functions were conducted - binary reward r_{bin}, baseline partial reward from [119], r_{base}, and the two proposed rewards, namely depth-based, r_{depth}, and area-based, r_{area}, respectively. The evaluation metrics are adopted from Anderson *et al.*[7].

Metric 1 discussion: Success rate (SR) - Table 4.4 shows the performance for this metric. For nearly every model, training via the proposed reward mechanism yields the best results, especially for episodes with larger path lengths, i.e. $L \geq 5$, where further exploration of the environment might be needed. This shows the benefits of adding a denser reward signal

Table 4.4. Metric 1: Success rate (%). The mean score over 5 runs is provided with the standard deviation as sub-scripts.

Models	$L \geq 1$				$L \geq 5$			
	r_{bin}	r_{base} [119]	ours		r_{bin}	r_{base} [119]	ours	
			r_{depth}	r_{bbox}			r_{depth}	r_{area}
GCN [180]	$33.1_{(0.8)}$	$33.3_{(1.4)}$	$31.7_{(0.7)}$	$\mathbf{35.3_{(0.5)}}$	$25.0_{(1.4)}$	$23.5_{(1.6)}$	$\mathbf{26.9_{(1.1)}}$	$24.6_{(0.8)}$
SAVN [171]	$34.7_{(0.5)}$	$\mathbf{40.7_{(1.4)}}$	$32.2_{(0.9)}$	$39.6_{(0.8)}$	$25.8_{(0.8)}$	$30.0_{(1.4)}$	$26.8_{1.3}$	$\mathbf{31.7_{(1.5)}}$
MJOLNIR-o [119]	$58.8_{(1.0)}$	$64.1_{(0.7)}$	$\mathbf{66.4_{(0.3)}}$	$66.3_{(1)}$	$40.6_{(0.6)}$	$46.6_{(1.6)}$	$50.5_{(0.7)}$	$\mathbf{51.5_{(1.3)}}$
MJOLNIR-r [119]	$65.5_{(0.6)}$	$68_{(0.9)}$	$\mathbf{77.1_{(0.7)}}$	$69.7_{(0.9)}$	$52.3_{(0.8)}$	$52.3_{(0.5)}$	$\mathbf{69.2_{(0.8)}}$	$57.3_{(1.3)}$

based on distance to objects.

Metric 2 discussion: Success weighted by Path Length (SPL) - As opposed to the results for success rate, the SPL performance drops for the proposed methods. This is shown in Table 4.5. A possible reason for this could be due to the added incentive that the agent now gets to explore regions around parent objects, before heading towards the target. However, this is not necessarily a major drawback, as exploring the environment is an important feature, especially in large and previously unseen environments.

It should also be noted that generally, the denser distance-based reward functions perform better for models that consider object relationships (like GCN [180], and the MJOLNIRs [119]). This supports the intuition that adding auxiliary signals based on surrounding objects can aid in

Table 4.5. Metric 2: SPL (%). The mean score over 5 runs is provided with the standard deviation as sub-scripts.

Models	$L \geq 1$				$L \geq 5$			
	r_{bin}	r_{base} [119]	ours		r_{bin}	r_{base} [119]	ours	
			r_{depth}	r_{bbox}			r_{depth}	r_{area}
GCN [180]	$10.0_{(0.4)}$	$\mathbf{10.8_{(0.5)}}$	$5.5_{(0.2)}$	$8.2_{(0.1)}$	$10.3_{(0.7)}$	$\mathbf{11.2_{0.7}}$	$7.3_{(0.3)}$	$8.7_{(0.3)}$
SAVN [171]	$11.0_{(0.2)}$	$\mathbf{11.1_{(0.3)}}$	$6.6_{(0.3)}$	$10.5_{0.2}$	$11.7_{(0.1)}$	$12.4_{(0.5)}$	$10.5_{0.3}$	$\mathbf{12.8_{(0.6)}}$
MJOLNIR-o [119]	$18.5_{(0.3)}$	$\mathbf{20.7_{(0.2)}}$	$11.6_{(0.1)}$	$15.8_{0.4}$	$17.8_{(0.3)}$	$\mathbf{20.0_{(0.6)}}$	$13.7_{(0.3)}$	$17.3_{(0.5)}$
MJOLNIR-r [119]	$24.4_{(0.3)}$	$\mathbf{26.5_{0.2}}$	$15.0_{(0.3)}$	$16.8_{(0.2)}$	$26.2_{(0.4)}$	$\mathbf{27.2_{(0.3)}}$	$20.3_{(0.4)}$	$19.3_{(0.4)}$

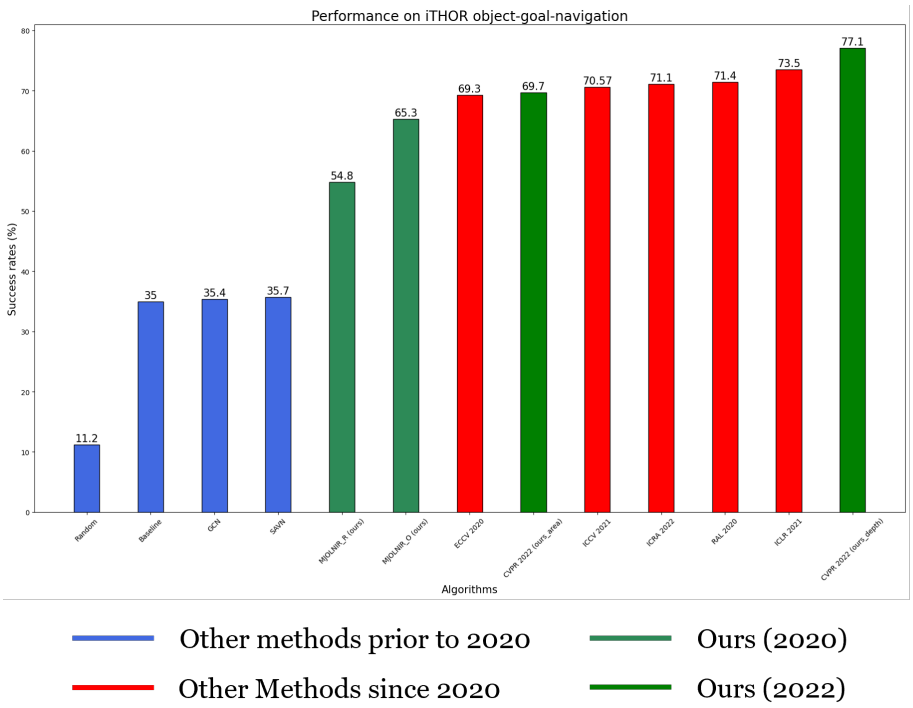


Figure 4.7. Bar graph showing the performance of different object-goal navigation algorithms over the years. [Best when viewed in color]

the search of far-off target objects. Figure 4.7 shows the performance of different object-goal navigation algorithms over the years, and how the two approaches mentioned in this chapter rank among them.

4.6 Conclusion

This chapter introduces MJOLNIR, a novel object-goal visual navigation algorithm that utilizes prior knowledge and also learns to associate object "closeness" in the form of parent-target hierarchy during training. This is done through the proposed *context vector* which can be easily derived from the output of an object detector. It is shown that besides the modified state space, knowledge graph, and reward shaping also play a significant role in guiding the agent to search for the target. Extensive experiments show that the agent can successfully find small target objects using the larger parent object as an anchor. The proposed model's performance can

be generalized across different unseen scenes and current state-of-the-art models. In an extension to this work conducted by Madhavan *et al*.[101], a distance-based reward shaping mechanism was introduced that provides denser feedback to the agent, thereby encouraging it to explore more of the environment. It has been shown that adopting this strategy leads to a higher success rate of reaching the target object for multiple models, especially for cases where the optimal path requires taking a longer sequence of actions.

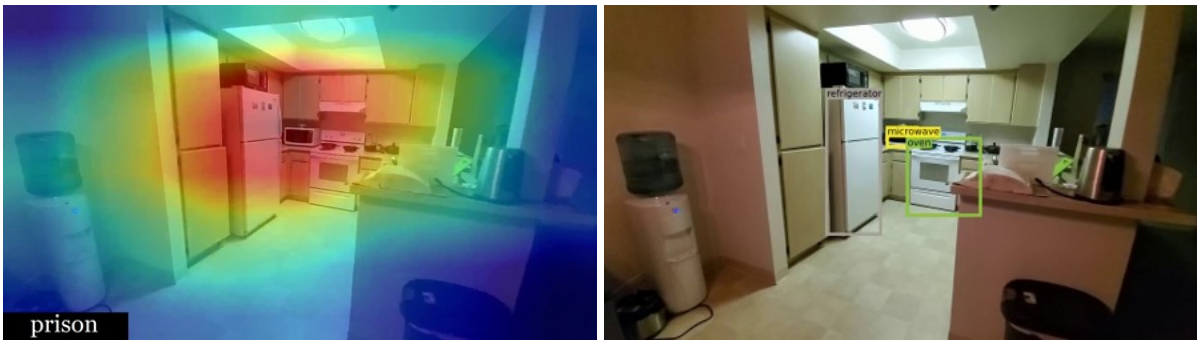
4.7 Supplementary Material

4.7.1 Appendix A: Object detector vs ResNet features

One of the significant changes made in the proposed algorithm from existing works [180] is to use object detection features instead of a classification probability from a CNN such as ResNet as the input node embedding in the graph convolution network. This is because the detection of multiple objects is integral to learning the "parent-target" hierarchical object relationships effectively. To further demonstrate this, a pre-trained YOLOv3 detector is run on an RGB image of a kitchen environment, taken using a cell phone camera, as shown in Figure 4.8b. Side-by-side, a Grad-cam visualization of ResNet-18, `resnet.mp4` on the same image is also depicted in Figure 4.8a. As seen from Figure 4.8, the multi-object detector can generate much fewer false-detections as compared to the 1000 class probability from ResNet, which generates noisy labels such as "`prison`".

4.7.2 Appendix B: Construction of Partial reward matrix M

As mentioned in Section 4.3, the conditional probability of a target object being found, given that a parent object was observed, $Pr(t|p)$, is obtained from the Partial reward matrix M. The training split of the AI2-THOR environment was utilized for creating M. For every floorplan, the 3D position of each object present there was plotted. Then, the occurrence of different target objects that were located within a Euclidean distance of 1 meter from a parent object (from Table



(a) Prediction of scene label from a pre-trained scene encoder

(b) Prediction of object bounding boxes from a pre-trained object detector

Figure 4.8. Comparison of model prediction from a pre-trained scene encoder vs a pre-trained object detector on an RGB image

4.7) was counted. It is to be noted that even though the same parent object might be present in more than one room type, its relationship with the target objects might be different. Thus, M was computed for every room type, as shown in Table 4.6. Normalizing each row provides the probability distribution of a target object $t \in T$, given a parent object $Pr(t|P)$. For the final parent reward R_p, a scaling factor k was used to ensure that the agent receives a lesser reward for the parent object as compared to the target reward $R_t = 5$. For the work in [119], a constant value of $k = 0.1$ was used, while for the follow-up work of [101], k was made to be a function of distance.

4.7.3 Appendix C: Parent and target object list

Table 4.7 provides the list of target objects, T, and parent objects, P, used in the experiments.

4.7.4 Appendix D: Implementation Details

For the word embeddings, the 300-D GloVe vectors were used that are pre-trained on 840 billion tokens of Common Crawl [125]. The A3C model is based on [81]. The model hyperparameters used for the experiments are tabulated below in Table 4.8.

4.8 Acknowledgements

Chapter 4, in part, is a reprint of the following papers:

- **A. Pal**[1], Y. Qiu[1] and H. I. Christensen, "Learning hierarchical relationships for object-goal navigation", in *Conference on Robot Learning (CoRL), 2020*. The dissertation author was the joint-first author of this paper.

- S. Madhavan, **A. Pal** and H. I. Christensen, "Role of reward shaping in object-goal navigation", in *Embodied AI Workshop, Conference on Computer Vision and Pattern Recognition (CVPR), 2022*. The dissertation author was a co-author of this paper.

[1]Equal Contribution

Table 4.6. Partial reward matrices for each of the 4 room-type in AI2-THOR.

Target \ Parent	Fridge	StoveBurner	Microwave	TableTop	Sink	CounterTop	Shelf
Toaster	0.15	**0.29**	0.15	0.04	0.15	0.23	-
Spatula	0.03	**0.31**	0.22	0.02	0.19	0.22	0.02
Bread	-	0.16	0.13	0.16	0.20	**0.36**	-
Mug	-	0.19	0.17	0.11	**0.30**	0.23	-
CoffeeMachine	0.08	0.10	0.10	0.06	**0.38**	0.28	-
Apple	0.12	0.12	0.12	0.12	**0.25**	0.23	0.02

(a) Kitchen

Target \ Parent	Drawer	Shelf	TableTop	Sofa	FloorLamp
Painting	**0.86**	0.14	-	-	-
Laptop	0.27	0.14	**0.36**	0.23	-
Television	**0.40**	0.2	0.35	-	0.05
RemoteControl	**0.25**	0.05	**0.25**	0.4	0.05
Vase	0.21	**0.47**	0.26	-	0.05
ArmChair	0.09	-	0.27	0.18	**0.45**

(b) Living room

Target \ Parent	Shelf	Dresser	NightStand	Drawer	Desk	Bed
Blinds	**1**	-	-	-	-	-
DeskLamp	0.16	0.2	0.12	0.24	**0.28**	-
Pillow	-	0.04	0.21	0.17	-	**0.58**
AlarmClock	0.19	0.11	**0.26**	0.21	0.04	0.19
CD	0.21	0.1	0.08	**0.33**	0.23	0.06

(c) Bedroom

Target \ Parent	CounterTop	Cabinet	Drawer	ShowerDoor	Toilet	Bathtub
Mirror	**0.51**	0.26	0.21	0.03	-	-
ToiletPaper	0.19	0.19	0.10	0.06	**0.46**	-
SoapBar	**0.29**	0.18	0.14	0.04	0.2	0.16
Towel	0.15	0.04	0.15	**0.44**	0.07	0.15
SprayBottle	**0.35**	0.20	0.16	0.02	0.22	0.06

(d) Bathroom

Table 4.7. Parent and target object list

Room type	Target Objects T	Parent Objects P
Kitchen	Toaster, Spatula, Bread, Mug, CoffeeMachine, Apple	Fridge, StoveBurner, Microwave, TableTop, Sink, CounterTop, Shelf
Living room	Painting, Laptop, Television, RemoteControl, Vase, ArmChair	Drawer, Shelf, TableTop, Sofa, FloorLamp
Bedroom	Blinds, DeskLamp, Pillow, AlarmClock, CD	Shelf, Dresser, NightStand, Drawer, Desk, Bed
Bathroom	Mirror, ToiletPaper, SoapBar, Towel, SprayBottle	CounterTop, Cabinet, Drawer, ShowerDoor, Toilet, Bathtub

Table 4.8. Summary of Hyperparameters

Parameters	**Baseline** [188]	**Scene Prior** [180]	**SAVN** [171]	**MJOLNIR-r (our)**	**MJOLNIR-o (our)**
Learning rate	0.0001	0.0001	0.0001	0.0001	0.0001
Optimizer	SharedAdam	SharedAdam	SharedAdam	SharedAdam	SharedAdam
Discount Factor	0.99	0.99	0.99	0.99	0.99
max #workers	8	8	6	8	8
Observation stream encoder	ResNet	ResNet	ResNet	ResNet	Object detector
Max training episodes	3×10^6	3×10^6	3×10^6	3×10^6	3×10^6

Chapter 5

Complete home robot rearrangement task

Creating autonomous agents to aid human beings in everyday household chores has long been considered to be the holy grail of service robotics research. This work takes a step towards that goal by proposing a complete system for an indoor tidy-up task. Usually, this comprises identifying misplaced objects in the environment and transferring them to their desired locations. Several aspects of this inherently long-horizon task make it particularly challenging in a real-world environment. Firstly, recognizing out-of-place objects in a noisy environment is a non-trivial problem. While state-of-the-art open-vocabulary object detectors [184, 52, 109, 187] are quite adept at localizing objects in a zero-shot manner, determining whether they belong in a particular environment is more complicated, as it also involves understanding scene context. Secondly, user preferences for placing objects in the "correct" room and surface (hereafter called *receptacle*), are often subjective, thereby inhibiting the sole use of generic common-sense reasoning models. Thirdly, manipulating unknown objects in a cluttered environment is still an open research problem due to the difficulty of affordance estimation and motion planning. Finally, delivering an object to a previously unlabeled receptacle in the target room is particularly challenging, especially if the precise location of said receptacle is unknown.

This chapter addresses each of the mentioned components for rearranging household objects in a real-world setting utilizing the Fetch [168] mobile manipulation platform. To ensure robustness and scalability within the physical world, a modular system has been proposed that is

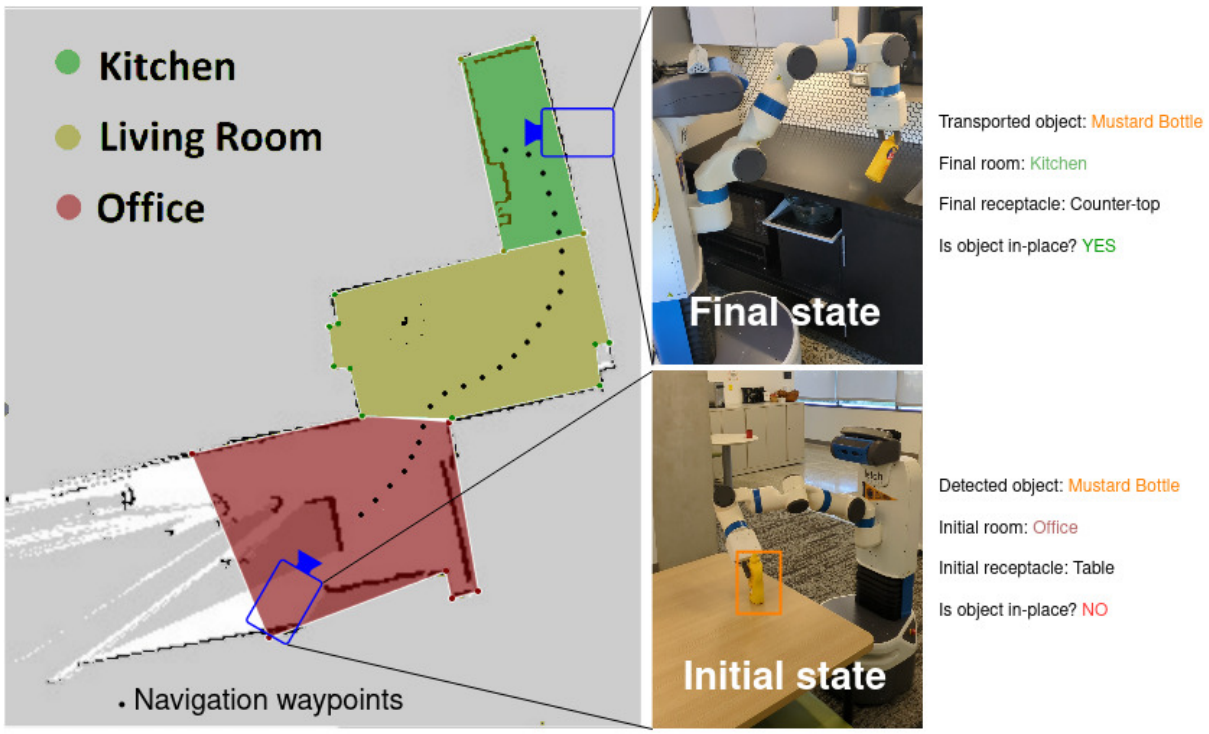


Figure 5.1. An example of a home-robot rearrangement task. At the initial state, the robot identifies the `mustard_bottle` object and determines that it is misplaced in the `office`. Subsequently, the robot transports it to its correct location in the `kitchen` on top of the `counter-top`. The semantic map used for the navigation task is shown on the left with the robot's trajectory.

capable of performing (i) user-preference-based reasoning through collaborative filtering, (ii) fine-grained pick-up of unknown objects and placement on previously unlabeled receptacles, and (iii) multi-room rearrangement. All these functionalities are coordinated by behavior trees that can handle failure at different levels. An example of the operation is shown in Figure 5.1.

The remainder of the chapter is organized as follows. Section 5.1 discusses existing approaches for object rearrangement in home-robot environments. Section 5.2 has a description of each component used to perform the overall task, with a summary of the integrated system in Section 5.3. The conducted experiments are explained in detail in Section 5.4, and a summary of the work with some future goals is provided in Section 5.5.

5.1 Background

Recently, indoor object rearrangement tasks using mobile robots have received a lot of attention from the robotics and computer vision community. Due to the increasing number of Embodied AI platforms available [128, 141, 142, 152, 10, 86], several approaches have been proposed for solving the complete mobile manipulation task in several home environments. However, most of these methods [60, 108, 128, 9, 152] are entirely trained in simulation, and therefore rarely generalize to real-world environments. Other works have adopted the task planning approach, but are either restricted to specific tasks such as folding clothes [148] and rearranging kitchens [172], or follow a pre-defined template [28]. Some approaches [140, 51, 135, 72] focus on the human-robot interaction aspect, but not on autonomy. Lately, large language models (LLMs) have gained popularity for robotic manipulation, both for task planning [174, 32, 21], as well as end-to-end execution [14, 65, 15, 5]. While these large foundational models are proficient at reasoning about object semantics, accurately grounding the offline acquired knowledge in a dynamic physical environment is still considered to be a non-trivial problem. Two efforts closest to the work done here are that of Wu *et al*.[174] and Castro *et al*.[20]. Wu *et al*.[174] use LLMs to infer generalized user preferences and use them to tidy a room. However, they do not handle fine-grained manipulation, need rigorous prompt engineering to understand user preferences, and are limited to within-room navigation. Castro *et al*.[20] do consider room-to-room navigation, but they rely on manually annotated prior semantic maps for querying the exact locations of target rooms and receptacles. In contrast, this work builds a simple 2D geometric map with rough room locations and proceeds to identify receptacles in the environment on the fly.

5.2 Components

The proposed ensemble system for home-robot rearrangement contains four primary modules: scene recognition and mapping, object rearrangement, manipulation, and navigation.

5.2.1 Semantic mapping and visual recognition

The detection module perceives the environment in two stages. The first stage involves the construction of a semantic map of the environment for localization, while the second stage deals with the recognition of objects in the environment. The localization system uses Cartographer mapper [58] to generate a LiDAR-based 2D occupancy-grid environment map. For simplicity, the locations in the map are manually annotated with a semantic label of the room category. This manual annotation step can also be replaced by an automated module such as [118]. The location of receptacles, however, is not annotated, as knowing their exact positions apriori is a strong assumption in dynamic environments. For object recognition, the DETIC [187] model that is trained on twenty-thousand object classes is used. With this detector, one can detect both manipulable objects and receptacle surfaces for the rearrangement task.

5.2.2 Object rearrangement

The rearrangement module involves repositioning objects in the home, using both *common-sense reasoning* (to determine target rooms) and *human preferences* (for selecting target receptacles). A large human-labeled dataset [68] is utilized for object placement preferences in homes, creating a knowledge base to predict likely room locations for objects. Then, user preference is used to capture diversity in human choices for receptacle locations. However, the dataset does not contain a particular user's preference for *all* the objects, leading to a sparse user-preference matrix. Thus, given scores and user-ranked preferences, collaborative filtering [1] is used to fill out the sparse matrix. Subsequently, matrix factorization [79] is performed to predict user ratings $r_{u,i}$ for user u and item i. For this case, an item i refers to an object's placement in a particular room and receptacle. An user's rating is predicted using $f(u,i) = \gamma_u * \gamma_i$. Here, $\gamma_u \in \mathbb{R}^d$ and $\gamma_i \in \mathbb{R}^d$ are latent vectors representing the row of a user in matrix γ_U and column of an item in matrix γ_I, and d is the lower dimensional space. To choose parameters $\gamma = \{\gamma_u, \gamma_i\}$ to closely fit the data, a loss function using Mean Squared Error with an

L2 regularization term is minimized.

$$\arg\min_{\gamma} \frac{1}{|\tau|} \sum_{r_{u,i} \in \tau} w_{u,i}(r_{u,i} - f(u,i))^2 + \lambda \Omega(\gamma) \tag{5.1}$$

where τ is the corpus of ratings and $\Omega(\gamma)$ is $\ell 2$ norm $||\gamma||_2^2$. The approach allows estimation of the full preferences of users' desired correct object placement locations.

Object rearrangement involves two main steps – (i) Identifying misplaced objects by checking if their current location is in the top-k (10 in this work) likely locations from the user-preference matrix, and (ii) Predicting preference-based placement by first determining the target room using common-sense reasoning, and then identifying various potential receptacle locations within that room based on a sampled user identity.

5.2.3 Manipulation of objects

The manipulation module includes planning to understand and construct a scene, analyzing interaction methods with the target object, and planning the required motion for effective interaction, all aligned with the task goal.

Before constructing the planning scene, the robot in this work possesses some prior knowledge of the environment. For instance, it understands that most objects should be positioned on a flat receptacle such as a `table`, or `counter-top`. Therefore, the receptacle serves as a common obstacle during the manipulation tasks, making it beneficial to prioritize its search once an object is detected. Finally, the receptacle is added as a single entity in the planning scene for efficient collision detection, while a voxel set represents the remaining non-target objects, optimizing resource usage.

Even though the robot knows the planning scene, interacting with the target object is crucial. In this work, grasping is the prevailing contact approach. For this, a learning-based grasp prediction [151] model is utilized to estimate a set of possible grasping poses. However, pick-and-place is not the only manipulation action available. The robot must also account for

potential object motions based on the task requirements. For instance, it might need to open a drawer before placing an object. Consequently, the robot must compute the required motion to open it after identifying a set of arm configurations to grasp the drawer handle. The robot may explore alternative approaches if the motion is found before the timeout.

5.2.4 Semantic navigation

The navigation module aims to move the robot between different locations for the rearrangement task and is considered in two stages – (i) room-to-room navigation for planning a path to the target room, and (ii) receptacle navigation for navigating to the correct receptacle in the target room.

For room-to-room navigation, the 2D coordinate of the center of the target room is first computed from the annotated semantic map. Using this destination point, a heuristic point-goal navigation algorithm is adopted to plan a trajectory by avoiding obstacles along the way with the Navfn planner. Upon reaching the target room, the receptacle navigation module is called. First, the entire room is scanned for possible receptacles for the held object, and the position of each candidate receptacle is updated in the map by re-projecting the detected object from the depth map of the camera. Then, the most likely target receptacle is chosen as per the rearrangement module 5.2.2. Finally, a second heuristic planner is called to make the robot move as close to the goal receptacle position as is feasible in collision-free space, which is achieved through the Carrot Planner.

5.3 System Integration

This section outlines the primary structure of the proposed system and then discusses the flow of control using behavior trees.

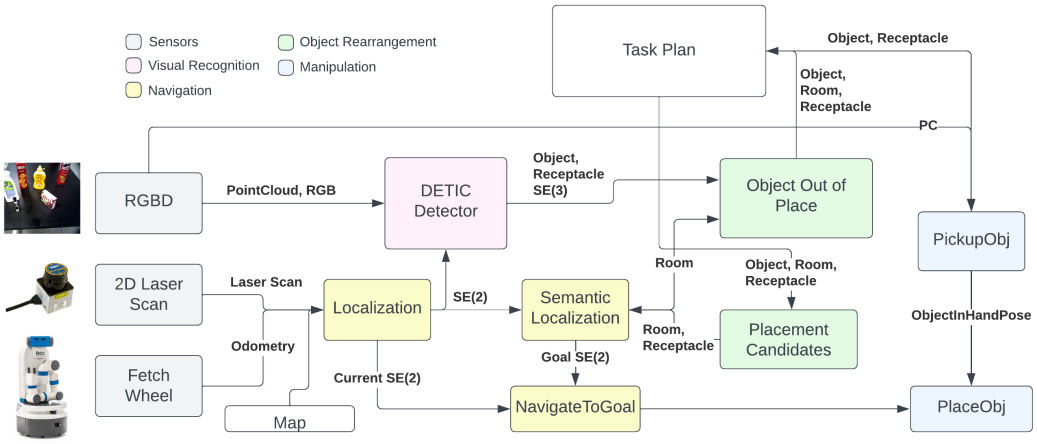


Figure 5.2. The overall architecture of the proposed system, as discussed in 5.3.1

5.3.1 System Architecture

Figure 5.2 depicts the overall architecture of the proposed system. The task plan is provided in the form of behavior trees, as discussed in the next section. The localization module reads the semantic map, along with sensor data, to get the robot's current coordinates in the room. The detector module reads the sensor data, along with the robot's location, and identifies objects in the environment along with their 3D locations on the map. The object rearrangement module obtains a list of $(object, receptacle, room)$ tuples from the perception and localization modules to identify misplaced objects and propose "correct" placements. The manipulation module picks up the misplaced object. The target room for placement provides the goal location for the navigator module, which then calls the perception module to locate the target receptacle and navigate to it. The manipulation module finally places the object either on the receptacle or inside the receptacle, depending on the specified goal from the rearrangement module.

5.3.2 Use of Behavior Trees for Integration

A key component of the complex home-robot system is the composition of the different capabilities of the robot to execute the task robustly and continuously. This calls for a control architecture that is modular and capable of switching between tasks such that the different tasks

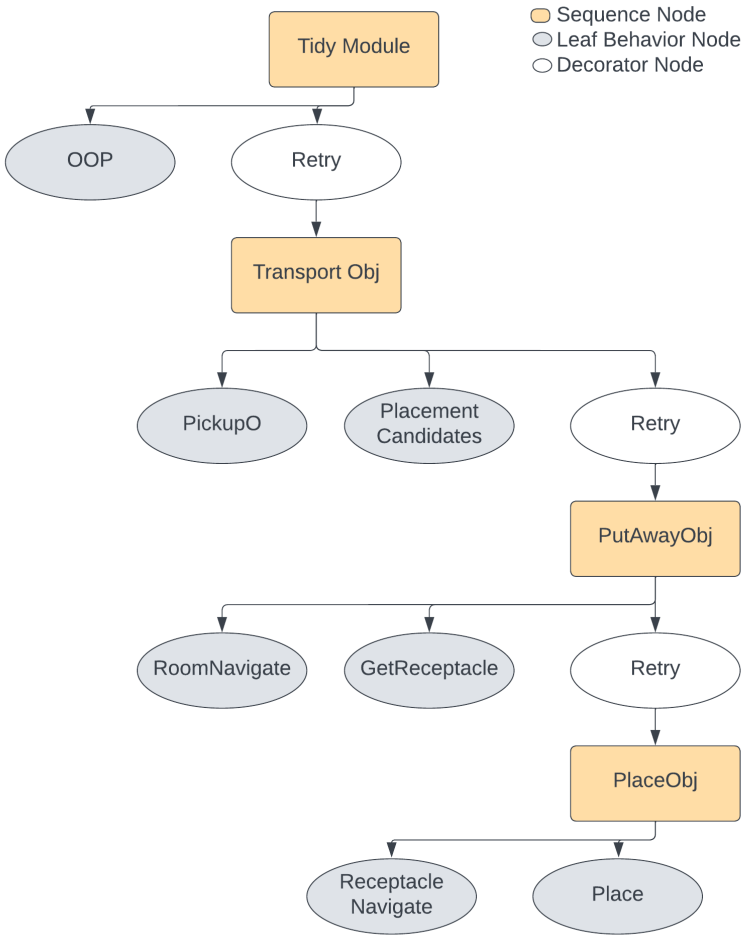


Figure 5.3. The complete behavior tree of the home-robot tidy module

can be called anywhere during the workflow. Consequently, Behavior Trees (BTs) are used to monitor and orchestrate the flow of the entire system. BTs are a modular control architecture developed for controlling autonomous agents that support reactive behavior. [26] A BT consists of control nodes and leaf nodes, where the leaf nodes are atomic operations that include actuation and sensing. The control nodes are behavior nodes that chain together multiple nodes. Each node (with its children) is a behavior that the robot can exhibit. A behavior can be composed of multiple behaviors. For instance, picking up a misplaced object is composed of two behaviors: identifying a misplaced object and picking up a target object. Figure 5.3 shows the BT of the home-robot tidy module. The system begins by calling the misplaced object identification (OOP) method

of the rearrangement module in Section 5.2.2. For every object, potential placement candidates (PlacementCandidates) are computed in Section 5.2.2. The Pickup Behavior in Section 5.2.3 is called on the misplaced object. RoomNavigator, followed by ReceptacleNavigator modules are executed, and given by the placement candidates. The PlaceBehavior is finally called to place the object. If the place action fails, then the robot tries other candidate receptacles until one succeeds, highlighting BT's advantages. This is implemented through multiple Decorator Nodes that can facilitate retry behaviors. The different messages from each behavior are passed around through blackboard mechanisms. The visual recognition module constantly runs in the background throughout the episode. The system continues to run until the robot either makes an unrecoverable mistake (such as dropping the object or a hardware failure) or all items are correctly placed.

5.4 Experiments

The proposed system is tested through various real-world experiments, involving (i) *Semantic mapping and visual recognition* for generating coarse semantic environment representations and detecting target objects and receptacle surfaces, (ii) *Object rearrangement* for identifying and repositioning misplaced objects, (iii) *Object manipulation* for ensuring stable ob-

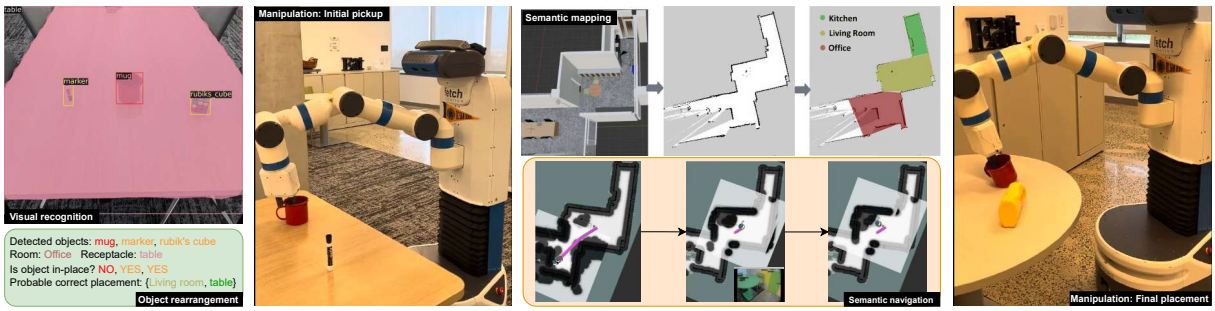


Figure 5.4. All proposed system components. The visual recognition module detects both target objects and receptacle surfaces. The object rearrangement module identifies misplaced objects and suggests their desired location. The manipulation module ensures the reliability of each pick and place action. The mapping module builds a 2D environment map and semantically paints it with room labels. Finally, the navigation module uses the semantic map to plan the robot's trajectory.

ject interactions, and (iv) *Semantic navigation* for robot's trajectory planning with the generated semantic map.

Figure 5.4 contains a pictorial representation of each of the modules at work for a tidy-up task. All the experiments are performed in the real world using a simple apartment environment, created from an actual communal office space within a university laboratory. The overall environment has an office space, living room, and a kitchen as shown in the semantic map in Figure 5.4. The following sections describe the different types of experiments.

5.4.1 Long-horizon object rearrangement

The first experiment considered is a long-horizon tidy-up task, where the robot has to identify multiple misplaced objects and move them to their respective target locations spanning multiple rooms. The rearrangement episode typically begins with detecting a misplaced object, o_1, in the environment. The entire tidy module is called to rearrange the object to the correct location. Upon reaching the destination, the robot further scans the environment for any other misplaced objects. If it finds another such object o_2, it repeats the entire process sequentially until o_2 has also been correctly placed.

Figure 5.5 illustrates the process where $o_1 = \texttt{mug}$ is transported from an office table to

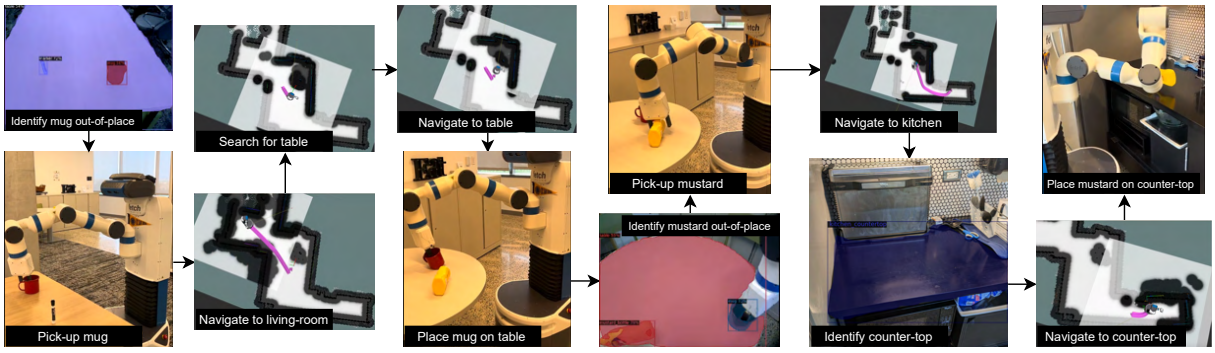


Figure 5.5. Long horizon rearrangement task. Initially, a `mug` is identified to be incorrectly placed on the office table. Then, the robot picks it up and navigates to the desired target location by first going to the living room and then moving towards the table receptacle. After placing the mug, a second object `mustard_bottle` is found misplaced on the living room table. Subsequently, the robot picks the bottle and transports it to the countertop in the kitchen.

the living-room table, and $o_2 =$ `mustard_bottle` is then moved from the living-room table to the kitchen counter-top.

5.4.2 User-preference based object tidy-up

The second experiment focuses on transferring an object o to different locations, catering to individual user preferences. This experiment acknowledges the subjective nature of object placement in homes. Section 5.2.2 describes a collaborative-filtering approach for generating a user matrix about how objects can be placed differently based on human preference. For this experiment, two users, U_1 and U_2, are sampled and their preferences regarding target room locations and receptacle surfaces are tabulated for eight different objects in Table 5.1.

Table 5.1. Preferred object placements for two sampled users

	Sampled user U_1		**Sampled user U_2**	
Objects	**Preferred rooms**	**Preferred receptacles**	**Preferred rooms**	**Preferred receptacles**
rubik's cube	office	[shelf, table]	living room	[drawer, table]
	kitchen	[counter, table]	office	[table, drawer]
	living room	[drawer, table]	kitchen	[drawer, table]
mustard bottle	kitchen	[drawer, counter]	kitchen	[shelf, counter]
	living room	[table, sofa]	living room	[table, drawer]
	office	[table, drawer]	office	[drawer, table]
marker	living room	[drawer, shelf]	office	[table, drawer]
	office	[table, drawer]	kitchen	[table, drawer]
	kitchen	[drawer, table]	living room	[table, shelf]
cracker box	kitchen	[drawer, table]	office	[shelf, drawer]
	living room	[drawer, table]	kitchen	[drawer, table]
	office	[drawer, shelf]	living room	[drawer, sofa]
bleach cleanser	living room	[drawer, table]	office	[shelf, table]
	office	[shelf, table]	kitchen	[drawer, table]
	kitchen	[shelf, drawer]	living room	[table, drawer]
gelatin box	office	[table, shelf]	living room	[table, drawer]
	kitchen	[drawer, counter]	office	[table, shelf]
	living room	[drawer, table]	kitchen	[drawer, counter]
potted meat can	kitchen	[counter, shelf]	office	[drawer, table]
	living room	[drawer, table]	kitchen	[counter, shelf]
	office	[drawer, table]	living room	[drawer, table]
mug	kitchen	[counter, sink]	living room	[table, shelf]
	living room	[shelf, sofa]	office	[drawer, table]
	office	[drawer, table]	kitchen	[sink, drawer]
soup can	living room	[table, drawer]	office	[drawer, shelf]
	kitchen	[drawer, counter]	kitchen	[drawer, shelf]
	office	[drawer, table]	living room	[sofa, drawer]

Real-world experiments are conducted using the `mug` object. As per Table 5.1, U_1 considers the preferred target room to be `kitchen`, with the top-2 receptacle surfaces being `counter` and `sink`. In contrast, U_2 desires the `mug` to be primarily placed in the `livingroom`, with the top-2 receptacles being `table` and `shelf`. Thus, multiple real-world episodes performed by sampling the preferences of U_1 and U_2 from the object rearrangement module, respectively.

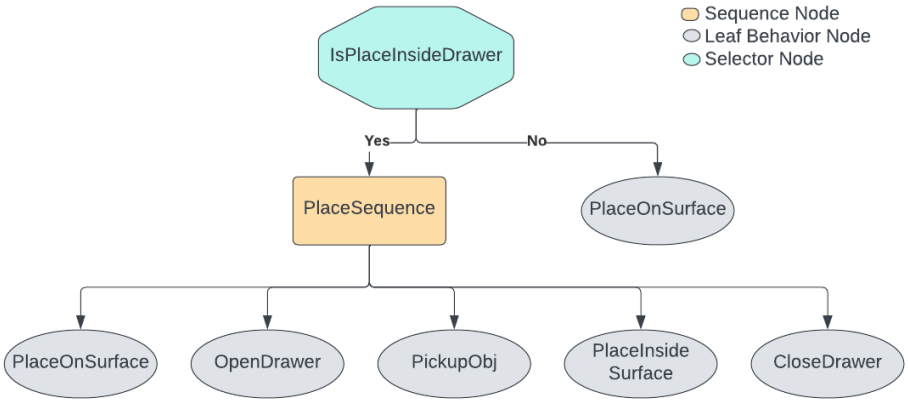


Figure 5.6. The behavior tree to place an object into the drawer.

5.4.3 Complex interactions

A rearrangement task may require the robot to interact with the environment before proceeding with object placement beyond just picking and placing. For instance, placing an object inside a *closed* receptacle. In this work, this concept is demonstrated through the task of placing a Rubik's Cube inside a drawer. Because the drawer is initially closed, the robot has to perform multiple sub-tasks based on the behavior tree shown in Figure 5.6. Furthermore, as depicted in Figure 5.7, the robot estimates a temporary location for the Rubik's Cube and predicts grasp poses to open the drawer. Following that, the robot places the cube into the temporary location and opens the drawer, so it can grasp and place the cube inside the drawer.

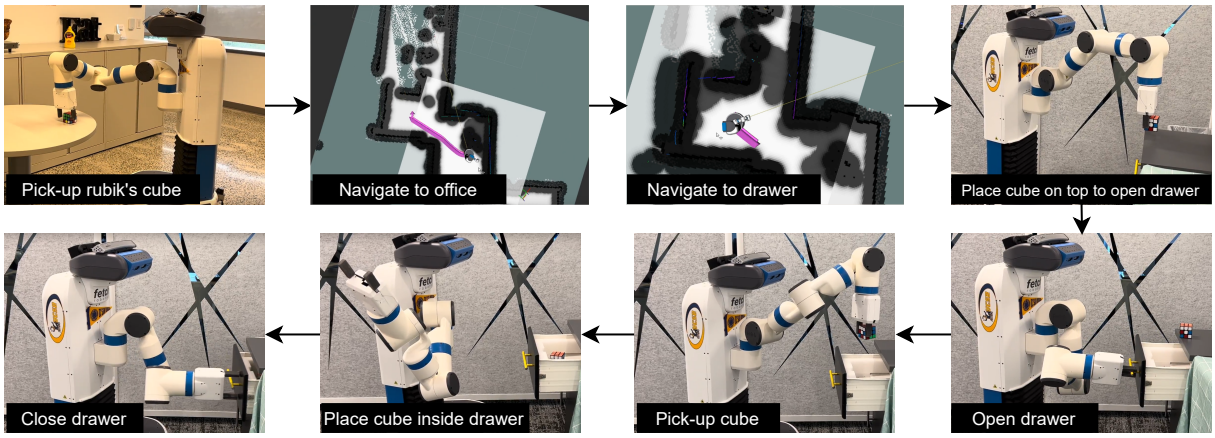


Figure 5.7. In a multifaceted task such as placing a Rubik's cube into a drawer, a robot must undertake a series of interrelated actions. Initially, the robot approaches the drawer. Recognizing that the drawer must be opened to place the cube inside, it then discerns the need to temporarily set down the Rubik's cube. Only after opening the drawer can it successfully place the cube within.

5.5 Conclusion

The world needs a home robot that can do more than vacuuming. This work has presented key components for navigating robustly in a home setting, detecting objects and receptacles, and determining if they are out of place. Skills for manipulating and handling objects in a daily setting for a task such as clean-up or reset of a home to a nominal setting are introduced to allow clean-up. Finally, using a combination of common-sense reasoning and recommender systems, a strategy to detect objects out of place and suggest improved locations to put them is discussed. All these techniques are integrated into a consistent and robust framework using behavior trees and implemented on the Fetch robot using a Robot Operating System (ROS)-based architecture. The final system has been demonstrated to work in a real-world scenario with modest complexity, for the clean-up of space by placement of objects in appropriate locations.

5.6 Acknowledgements

Chapter 5, in part, is a reprint of S. R. Iyer, **A. Pal**, J. Hu, A. Adeleye, A. Aggarwal and H. I. Christensen, "Household navigation and manipulation for everyday object rearrangement

tasks", in *IEEE International Conference on Robotic Computing (IRC), 2023*. The dissertation author was a co-author of this paper.

Chapter 6

Conclusion and Future Work

The field of Embodied AI, primarily driven by recent developments in the fields of robotics and computer vision, has experienced unprecedented growth over the last decade or so. A major factor contributing to this success has been the judicious amalgamation of classical control and motion planning techniques in robotics, along with the ability of deep-learning-based models to interpret and understand semantic context from the surroundings. The primary motivation for the many works [118, 117, 119, 101, 120, 62] done in this dissertation has been to introduce several such *hybrid* semantic models, which can be sequentially used for building smart and robust home robots.

6.1 Dissertation summary

Chapter 2 considers the problem of visual place categorization, where an autonomous robotic system is spawned in a previously unseen environment, for example, a novel household apartment, and learns to predict the semantic category of surrounding scenes based on its visual sensory input. The key idea is to incorporate deep fusion models that are derived by the joint understanding of holistic scene representations, and information about objects in the vicinity. A number of hybrid models are proposed, where a classical technique is used to build a 2D geometric map comprising obstacles and free space, and a learning model is used to predict the scene label from images, and sequentially augment the metric map with semantic context.

Experiments across static image datasets, dynamic videos suffering from motion blur, and a real robotic platform showcase the generalization ability of the presented approach.

Chapter 3 expands the idea of semantic scene understanding to the domain of outdoor driving scenarios. Predicting the driver's focus of attention is an active area of research in the autonomous driving community. However, most existing techniques rely on raw human gaze information by recording the driver's eye movements, which often ignore scene semantics. To alleviate this issue, a novel detection approach was presented, which retains the information about the intent of the driver, while also capturing driving-specific contextual information typically overlooked by raw gaze. Building on top of this, a complete saliency prediction framework is proposed that further augments the semantic gaze by taking into account vital aspects such as distance to objects (depth), ego vehicle speed, and pedestrian crossing intent. Exhaustive experiments conducted through four popular saliency prediction algorithms show the superior performance of the current work in a majority of cases.

Chapter 4 considers a special case of general robot navigation by grounding it to the task of object-goal navigation. This involves navigating through a previously unseen environment in search of instances of particular objects. The key idea described in this dissertation is to utilize object-object relationships between target objects, and larger, more salient parent objects which serve as receptacles for the target objects. The idea is implemented via two smart search strategies – (i) Using multi-object detectors as opposed to generic scene encoders for better localization of objects, and (ii) Incorporating reward shaping into the reinforcement learning framework, where a partial reward is introduced to the agent to encourage it to perform localized search around larger parent objects to look for the target. By utilizing the mentioned strategies, a huge improvement over the existing state-of-the-art approaches was observed, both in terms of improved evaluation metrics and also in terms of convergence speed. In a later work, an extension of this was proposed by making the partial rewards a continuous function of distance to the objects, thereby improving the performance further.

Chapter 5 takes aspects from all the previous work and integrates them towards a com-

plete home robot rearrangement planning task. Specifically, the task of tidying up indoor environments by transferring objects from their misplaced locations to their correct location is considered. There are primarily four major components to this – (i) Semantic mapping and visual recognition: These comprise perception modules, where a semantic map (similar to Chapter 2) of the environment is first built while the robot is touring the environment. Additionally, off-the-shelf visual recognition modules are used for detecting objects from ego-centric camera images recorded by the robot, (ii) Object rearrangement planner: This is the brain of the overall operation, which takes as input detections from the previous component, and figures out which objects are out-of-place, and determines their correct target location, (iii) Multi-stage navigation: Robot location in this work is considered in two stages, movement to the target room location, followed by detection of the correct target receptacle and subsequent motion planning of the robot towards it, and (iv) Object manipulation: This component is primarily used as a tool for object interaction, where off-the-shelf methods are used to pick-and-place target objects from their initial to final locations. The entire demonstration has been carried out on a real robotic platform in a tiny apartment environment.

6.2 Considerations for future work

As scholars and intellectuals from times immemorial have theorized, research is always a *work in progress*. That is, the ultimate purpose of any study should be to pave the way for further studies to be made along similar lines. Since the research in this dissertation was not an exception to this theory, it can be hypothesized that the work done here will pave the way for new research to be undertaken in the field of Embodied AI. To conclude this text, some of the existing challenges faced during the author's doctoral journey are highlighted, along with some recommendations for future work.

6.2.1 Main challenges faced over the years

- End-to-end methods never really generalize in robotics tasks: Ever since the dawn of the deep-learning era, end-to-end learning models have been quite popular as they can be trained entirely from data, thereby requiring very little human effort. While a large number of algorithms have been developed that work extremely well on specific datasets, such models rarely generalize to novel data samples outside the training distribution. As the field of robotics is full of such corner cases, the paradigm of end-to-end learning algorithms has tasted little success here.

- Static vs dynamic settings: Many tasks such as object detection, semantic segmentation, etc. have achieved a point of saturation in the field of computer vision. This is because the powerful learning algorithms these days are quite adept at exploiting the bias in static datasets. This leads to near-perfect performance in terms of evaluation metrics. However, when transitioning to more dynamic settings such as a moving robotic camera, a number of unexpected circumstances might occur such as poor resolution, bad illumination, motion blur, etc, all of which are quite difficult to model on a static dataset. As a result of this domain shift, even those perfect algorithms suffer in performance on real data.

- Real-time operation is a rarity: Due to the increasing computation capability of modern computers at relatively low costs, the trend in research has gone towards building large-scale models that are trained across multiple tasks, eventually improving the accuracy across all of them. Such foundational models often scale up to billions of parameters and require enormous computational resources. An unfortunate by-product of this is the inability to run these large models on robotic systems which are difficult to be fitted with large computers. This greatly impacts real-time performance for general robotics tasks.

- Cost of home robots limit the scope of realistic use-cases: Despite the enormous success of Embodied AI approaches, the topic has largely been limited to academic researchers and

very few industrialists. A major reason for this is the vast infrastructural cost of building a service robot.

6.2.2 Recommendations for future work

- Hybrid methods incorporating modular learning is the way to go: One of the main takeaways from this dissertation is that hybrid models that utilize the best of both classical planning and learned semantics, often perform the best when it comes to robotics tasks. By design, most hybrid approaches are modular, meaning that a complex task is tackled into several sub-components – as shown in Chapter 5, where an object rearrangement task was divided into semantic mapping and scene understanding, contextual planning, robotic navigation, and manipulation. By adopting this divide-and-conquer approach, the individual modules can build on existing approaches as opposed to manual design from scratch, while also being more interpretable than end-to-end learning methods.

- Domain adaptation should be key to any algorithm: To ensure the sustainability of newly developed algorithms in the wild, they must be taught as many real-world corner cases as possible. Therefore, for a proposed approach to be considered a success, whenever possible, it should be tested across a combination of sim-to-real tasks. Foundational models take a large step in this direction, as they are typically exposed to many different datasets during training, with the hope of truly learning representations of world models.

- Efficient ways to use smaller models are necessary: To ensure real-time performance, the size of learning-based models needs to be reduced, without sacrificing performance. A possible solution is to utilize knowledge distillation techniques to transfer knowledge from one or several large models into smaller models designed for specific downstream tasks. Recently, a number of pretraining and Parameter Efficient Fine-Tuning (PEFT) approaches have also gained popularity, particularly for large language models. Such approaches can greatly enhance the capability of modern algorithms to work on robotics platforms with low computational resources, or even edge devices such as cell phones.

- Simulation could be beneficial, but only with realistic assumptions: In robotics research, simulation is a necessary evil. Many tasks such as large-scale navigation, multi-object manipulation, and complex human-robot interactions are much easier to scale in simulation. However, as the number of synthetic benchmarks in Embodied AI continues to increase, an everlasting question remains – how well do models trained on simulation transfer to real-world settings? Towards answering this question, there has been a strong push amongst researchers to develop infrastructure that facilitates sim-to-real transfer on hardware. Consequently, a number of environments exist nowadays where an algorithm can first be designed and evaluated at scale in simulation, before transferring it to a real-robotic platform.

6.3 Acknowledgements

Chapter 6, in part, is a reprint of the following papers:

- **A. Pal**, C. Nieto-Granda and H. I. Christensen, "DEDUCE: Diverse scEne Detection methods in Unseen Challenging Environments", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019*. The dissertation author was the primary author of this paper.

- **A. Pal**, S. Mondal and H. I. Christensen, "Looking at the right stuff - Guided semantic-gaze for autonomous driving", in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020*. The dissertation author was the primary author of this paper.

- **A. Pal**[1], Y. Qiu[1] and H. I. Christensen, "Learning hierarchical relationships for object-goal navigation", in *Conference on Robot Learning (CoRL), 2020*. The dissertation author was the joint-primary author of this paper.

[1]Equal Contribution

- S. Madhavan, **A. Pal** and H. I. Christensen, "Role of reward shaping in object-goal navigation", in *Embodied AI Workshop, Conference on Computer Vision and Pattern Recognition (CVPR), 2022*. The dissertation author was a co-author of this paper.

- S. R. Iyer, **A. Pal**, J. Hu, A. Adeleye, A. Aggarwal and H. I. Christensen, "Household navigation and manipulation for everyday object rearrangement tasks", in *IEEE International Conference on Robotic Computing (IRC), 2023*. The dissertation author was a co-author of this paper.

Bibliography

[1] Nichola Abdo, Cyrill Stachniss, Luciano Spinello, and Wolfram Burgard. Robot, organize my shelves! tidying up objects by predicting user preferences. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[2] Waleed Abdulla. Mask R-CNN for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017.

[3] Radhakrishna Achanta, Francisco Estrada, Patricia Wils, and Sabine Süsstrunk. Salient region detection and segmentation. In *International conference on computer vision systems*, pages 66–75. Springer, 2008.

[4] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Süsstrunk. Frequency-tuned salient region detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 1597–1604, 2009.

[5] Michael Ahn et al. Do as i can, not as i say: Grounding language in robotic affordances, 2022.

[6] Stefano Alletto, Andrea Palazzi, Francesco Solera, Simone Calderara, and Rita Cucchiara. DR(eye)VE: A dataset for attention-based tasks with applications to autonomous and assisted driving. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2016.

[7] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

[8] Bram Bakker, Jürgen Schmidhuber, et al. Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization. In *Proceedings of IAS*, pages 438–445, 2004.

[9] Dhruv Batra, Angel X Chang, Sonia Chernova, Andrew J Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, et al. Rearrangement: A challenge for embodied ai. *arXiv preprint arXiv:2011.01975*, 2020.

[10] Vincent-Pierre Berges, Andrew Szot, Devendra Singh Chaplot, Aaron Gokaslan, Roozbeh Mottaghi, Dhruv Batra, and Eric Undersander. Galactic: Scaling end-to-end reinforcement learning for rearrangement at 100k steps-per-second. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[11] Johann Borenstein, Yoram Koren, et al. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE T-RO*, 7(3):278–288, 1991.

[12] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection: A survey. *Computational Visual Media*, pages 1–34, 2014.

[13] Richard Bormann, Florian Jordan, Wenzhe Li, Joshua Hampp, and Martin Hägele. Room segmentation: Survey, implementation, and analysis. In *ICRA*, 2016.

[14] Anthony Brohan et al. Rt-1: Robotics transformer for real-world control at scale, 2023.

[15] Anthony Brohan et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023.

[16] Neil Bruce and John Tsotsos. Saliency based on information maximization. In *Advances in neural information processing systems*, pages 155–162, 2006.

[17] Wolfram Burgard, Armin B. Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. The interactive museum tour-guide robot. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI '98, page 11–18, USA, 1998. American Association for Artificial Intelligence.

[18] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *IEEE transactions on pattern analysis and machine intelligence*, 41(3):740–757, 2018.

[19] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018.

[20] Sebastian Castro. Behavior Trees for Home Service Robotics Tasks. https://www.youtube.com/watch?v=xbvMnpwXNPk, 2022.

[21] Haonan Chang, Kai Gao, Kowndinya Boyalakuntla, Alex Lee, Baichuan Huang, Harish Udhaya Kumar, Jinjin Yu, and Abdeslam Boularias. Lgmcts: Language-guided monte-carlo tree search for executable semantic object rearrangement, 2023.

[22] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020.

[23] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *IEEE ICCV*, pages 2722–2730, 2015.

[24] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Proceedings of the IEEE international conference on computer vision*, pages 686–695, 2017.

[25] Junsuk Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2219–2228, 2019.

[26] Michele Colledanchise and Petter Ögren. *Behavior trees in robotics and AI: An introduction*. CRC Press, 2018.

[27] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. A Deep Multi-Level Network for Saliency Prediction. In *International Conference on Pattern Recognition (ICPR)*, 2016.

[28] Guowei Cui, Wei Shuai, and Xiaoping Chen. Semantic task planning for service robots in open worlds. *Future Internet*, 13(2), 2021.

[29] AJ DAVISON. Real-time simultaneous localization and mapping with a single camera. In *IEEE ICCV*, pages 1403–1410, 2003.

[30] Matt Deitke, Dhruv Batra, Yonatan Bisk, Tommaso Campari, Angel X Chang, Devendra Singh Chaplot, Changan Chen, Claudia Pérez D'Arpino, Kiana Ehsani, Ali Farhadi, et al. Retrospectives on the embodied ai workshop. *arXiv preprint arXiv:2210.06849*, 2022.

[31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[32] Yan Ding, Xiaohan Zhang, Chris Paxton, and Shiqi Zhang. Task and motion planning with large language models for object rearrangement. *arXiv preprint arXiv:2303.06247*, 2023.

[33] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *PAMI*, 34, 2012.

[34] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model, 2023.

[35] R. Druon, Y. Yoshiyasu, A. Kanezaki, and A. Watt. Visual object search by learning spatial context. *IEEE RAL*, 5(2):1279–1286, April 2020.

[36] Heming Du, Xin Yu, and Liang Zheng. Learning object relation graph and tentative policy for visual navigation. In *European Conference on Computer Vision*, pages 19–34. Springer, 2020.

[37] Heming Du, Xin Yu, and Liang Zheng. {VTN}et: Visual transformer network for object goal navigation. In *International Conference on Learning Representations*, 2021.

[38] Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

[39] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[40] S. Ekvall, D. Kragic, and P. Jensfelt. Object detection and mapping for service robot tasks. *Robotica*, 25(2):175–187, 2007.

[41] P. Espinace, T. Kollar, A. Soto, and N. Roy. Indoor scene recognition through object detection. In *ICRA*, 2010.

[42] Andreas Ess, Bastian Leibe, and Luc Van Gool. Depth and appearance for mobile scene analysis. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[43] Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen. Shifting more attention to video salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8554–8564, 2019.

[44] Ehsan Fazl-Ersi and John K Tsotsos. Histogram of oriented uniform patterns for robust place recognition and categorization. *IJRR*, 2012.

[45] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[46] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016.

[47] Alessandro Giusti, Jérôme Guzzi, Dan C Cireşan, Fang-Lin He, Juan P Rodríguez, Flavio Fontana, Matthias Faessler, Christian Forster, Jürgen Schmidhuber, Gianni Di Caro, et al. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE RAL*, 1(2):661–667, 2015.

[48] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017.

[49] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. *arXiv preprint arXiv:1806.01260*, 2018.

[50] Sandeep Goel and Manfred Huber. Subgoal discovery for hierarchical reinforcement learning using learned policies. In *AAAI Conference on Artificial Intelligence*, 2003.

[51] Birgit Graf, Matthias Hans, and Rolf D Schraft. Care-o-bot ii—development of a next generation robotic home assistant. *Autonomous robots*, 16(2), 2004.

[52] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021.

[53] Pratik Gujjar and Richard Vaughan. Classifying pedestrian actions in advance using predicted video of urban driving scenes. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'19)*, Montreal, Canada, May 2019.

[54] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *IEEE CVPR*, pages 2616–2625, 2017.

[55] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE T-SSC*, 4(2):100–107, 1968.

[56] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[57] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE CVPR*, June 2016.

[58] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar slam. In *2016 IEEE international conference on robotics and automation (ICRA)*, 2016.

[59] Derek Hoiem, Alexei A Efros, and Martial Hebert. Geometric context from a single image. In *IEEE ICCV*, volume 1, pages 654–661. IEEE, 2005.

[60] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, 2022.

[61] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 20(11):1254–1259, 1998.

[62] Shrutheesh R. Iyer, Anwesan Pal, Jiaming Hu, Akanimoh Adeleye, Aditya Aggarwal, and Henrik I. Christensen. Household navigation and manipulation for everyday object rearrangement tasks, 2023.

[63] Allison Janoch, Sergey Karayev, Yangqing Jia, Jonathan T Barron, Mario Fritz, Kate Saenko, and Trevor Darrell. A category-level 3d object dataset: Putting the kinect to work. In *Consumer depth cameras for computer vision*, pages 141–165. Springer, 2013.

[64] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. SALICON: Saliency in context. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[65] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts, 2023.

[66] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *IEEE CVPR*, pages 2901–2910, 2017.

[67] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *IEEE CVPR*, pages 3668–3678, 2015.

[68] Yash Kant, Arun Ramachandran, Sriram Yenamandra, Igor Gilitschenski, Dhruv Batra, Andrew Szot, and Harsh Agrawal. Housekeep: Tidying virtual households using commonsense reasoning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, 2022.

[69] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *IJRR*, 30(7):846–894, 2011.

[70] Dongsung Kim and Ramakant Nevatia. Symbolic navigation with a generic map. *Autonomous Robots*, 6(1):69–88, 1999.

[71] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[72] Ralf Kittmann, Tim Fröhlich, Johannes Schäfer, Ulrich Reiser, Florian Weißhardt, and Andreas Haug. Let me introduce myself: I am care-o-bot 4, a gentleman robot. *Mensch und computer 2015–proceedings*, 2015.

[73] Christof Koch and Shimon Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence*, pages 115–141. Springer, 1987.

[74] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

[75] Noriyuki Kojima and Jia Deng. To learn or not to learn: Analyzing the role of learning for navigation in virtual environments. *arXiv preprint arXiv:1907.11770*, 2019.

[76] T. Kollar and N. Roy. Utilizing object-object and object-scene context when planning to find things. In *ICRA*, pages 4116–4121, 2009.

[77] Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: an interactive 3d environment for visual AI. *CoRR*, abs/1712.05474, 2017.

[78] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An interactive 3D environment for visual AI. *arXiv preprint arXiv:1712.05474*, 2017.

[79] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008.

[80] Ioannis Kostavelis and Antonios Gasteratos. Semantic mapping for mobile robotics tasks. *Robot. Auton. Syst.*, 66(C), April 2015.

[81] Ilya Kostrikov. Pytorch implementations of asynchronous advantage actor critic. https://github.com/ikostrikov/pytorch-a3c, 2018.

[82] Iuliia Kotseruba, Amir Rasouli, and John K Tsotsos. Joint attention in autonomous driving (JAAD). *arXiv preprint arXiv:1609.04741*, 2016.

[83] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017.

[84] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[85] Adam Daniel Laud. *Theory and application of reward shaping in reinforcement learning*. University of Illinois at Urbana-Champaign, 2004.

[86] Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, et al. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. *arXiv preprint arXiv:2108.03272*, 2021.

[87] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 280–287, 2014.

[88] Yiyi Liao, Sarath Kodagoda, Yue Wang, Lei Shi, and Yong Liu. Understand scene categories by objects: A semantic regularized scene classifier using convolutional neural networks. In *ICRA*. IEEE, 2016.

[89] Yiyi Liao, Sarath Kodagoda, Yue Wang, Lei Shi, and Yong Liu. Place classification with a graph regularized deep neural network. *IEEE Transactions on Cognitive and Developmental Systems*, 9(4), 2017.

[90] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[91] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017.

[92] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[93] Chris Linegar, Winston Churchill, and Paul Newman. Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera. In *ICRA*, pages 787–794. IEEE, 2016.

[94] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2024–2039, 2015.

[95] Nian Liu, Junwei Han, and Ming-Hsuan Yang. PiCANet: Learning pixel-wise contextual attention for saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3089–3098, 2018.

[96] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1871–1880, 2019.

[97] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE Transactions on Pattern analysis and machine intelligence*, 33(2):353–367, 2010.

[98] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.

[99] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3623–3632, 2019.

[100] Ren C. Luo and Michael Chiou. Hierarchical semantic mapping using convolutional neural networks for intelligent service robotics. *IEEE Access*, 6:61287–61294, 2018.

[101] Srirangan Madhavan, Anwesan Pal, and Henrik I. Christensen. Role of reward shaping in object-goal navigation, 2022.

[102] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5667–5675, 2018.

[103] Oleksandr Maksymets, Vincent Cartillier, Aaron Gokaslan, Erik Wijmans, Wojciech Galuba, Stefan Lee, and Dhruv Batra. Thda: Treasure hunt data augmentation for semantic navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15374–15383, October 2021.

[104] Massimiliano Mancini, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Robust place categorization with deep domain generalization. *IEEE RA-L*, 2018.

[105] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. OK-VQA: A visual question answering benchmark requiring external knowledge. In *IEEE CVPR*, pages 3195–3204, 2019.

[106] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2929–2936. IEEE, 2009.

[107] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

[108] So Yeon Min, Devendra Singh Chaplot, Pradeep Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov. Film: Following instructions in language with modular methods. *arXiv preprint arXiv:2110.07342*, 2021.

[109] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection. In *European Conference on Computer Vision*, 2022.

[110] Dmytro Mishkin, Alexey Dosovitskiy, and Vladlen Koltun. Benchmarking classic and learned navigation in complex 3D environments. *arXiv preprint arXiv:1901.10915*, 2019.

[111] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, pages 1928–1937, 2016.

[112] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE CVPR*, pages 891–898, 2014.

[113] Athma Narayanan, Isht Dwivedi, and Behzad Dariush. Dynamic traffic scene classification with space-time coherence. *arXiv preprint arXiv:1905.12708*, 2019.

[114] José Neira and Juan D Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on robotics and automation*, 17(6):890–897, 2001.

[115] C Nieto-Granda, J G Rogers III, A J B Trevor, and H I Christensen. Semantic Map Partitioning in Indoor Environments Using Regional Analysis. In *IROS*, Taiwan, October 2010. IEEE.

[116] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42:145–175, 2001.

[117] Anwesan Pal, Sayan Mondal, and Henrik I. Christensen. "Looking at the Right Stuff" - Guided Semantic-Gaze for Autonomous Driving. In *IEEE CVPR*, June 2020.

[118] Anwesan Pal, Carlos Nieto-Granda, and Henrik I Christensen. Deduce: Diverse scene detection methods in unseen challenging environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[119] Anwesan Pal, Yiding Qiu, and Henrik Christensen. Learning hierarchical relationships for object-goal navigation. In Jens Kober, Fabio Ramos, and Claire Tomlin, editors, *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pages 517–528. PMLR, 16–18 Nov 2021.

[120] Anwesan Pal, Sahil Wadhwa, Ayush Jaiswal, Xu Zhang, Yue Wu, Rakesh Chada, Pradeep Natarajan, and Henrik I. Christensen. Fashionntm: Multi-turn fashion image retrieval via cascaded memory, 2023.

[121] Andrea Palazzi, Davide Abati, Francesco Solera, Rita Cucchiara, et al. Predicting the driver's focus of attention: the DR(eye)VE project. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1720–1733, 2018.

[122] Andrea Palazzi, Francesco Solera, Simone Calderara, Stefano Alletto, and Rita Cucchiara. Learning where to attend like a human driver. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 920–925. IEEE, 2017.

[123] Derrick Parkhurst, Klinton Law, and Ernst Niebur. Modeling the role of salience in the allocation of overt visual attention. *Vision research*, 42(1):107–123, 2002.

[124] Abhishek Patil, Srikanth Malla, Haiming Gang, and Yi-Ting Chen. The H3D dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes. In *International Conference on Robotics and Automation*, 2019.

[125] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[126] Rolf Pfeifer and Fumiya Iida. Embodied artificial intelligence: Trends and challenges. *Lecture notes in computer science*, pages 1–26, 2004.

[127] A. Pronobis, O. M. Mozos, and B. Caputo. Svm-based discriminative accumulation scheme for place recognition. In *ICRA*, 2008.

[128] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[129] Ariadna Quattoni and Antonio B. Torralba. Recognizing indoor scenes. In *CVPR*, pages 413–420, 2009.

[130] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie. Objects in context. In *IEEE ICCV*, pages 1–8. IEEE, 2007.

[131] Vasili Ramanishka, Yi-Ting Chen, Teruhisa Misu, and Kate Saenko. Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[132] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[133] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, 2017.

[134] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[135] Ulrich Reiser, Christian Connette, Jan Fischer, Jens Kubacki, Alexander Bubeck, Florian Weisshardt, Theo Jacobs, Christopher Parlitz, Martin Hägele, and Alexander Verl. Care-o-bot® 3-creating a product vision for service robot applications by integrating design and technology. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[136] Parvaneh Saeedi, Peter D Lawrence, and David G Lowe. Vision-based 3-D trajectory tracking for unknown environments. *IEEE T-RO*, 22(1):119–136, 2006.

[137] Niko Sänderhauf, Oliver Brock, Walter Scheirer, Raia Hadsell, Dieter Fox, Jürgen Leitner, Ben Upcroft, Pieter Abbeel, Wolfram Burgard, Michael Milford, and Peter Corke. The limits and potentials of deep learning for robotics. *IJRR*, 2018.

[138] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[139] Alexander Sax, Jeffrey O. Zhang, Bradley Emi, Amir Zamir, Silvio Savarese, Leonidas Guibas, and Jitendra Malik. Learning to navigate using mid-level visual priors. In *Proceedings of the Conference on Robot Learning*, pages 791–812. PMLR, 30 Oct–01 Nov 2020.

[140] C Schaeffer and T May. Care-o-bot-a system for assisting elderly or disabled persons in home environments. *Assistive technology on the threshold of the new millenium*, 3, 1999.

[141] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[142] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

[143] Abhinav Shrivastava and Abhinav Gupta. Contextual priming and feedback for Faster R-CNN. In *ECCV*, pages 330–348. Springer, 2016.

[144] C. Siagian and L. Itti. Rapid biologically-inspired scene classification using features shared with visual attention. *T. PAMI*, 29(2):300–312, 2007.

[145] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[146] Robert Sim and James J Little. Autonomous vision-based exploration and mapping using hybrid maps and rao-blackwellised particle filters. In *IROS*, pages 2082–2089. IEEE, 2006.

[147] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, pages 567–576, 2015.

[148] Siddharth Srivastava, Shlomo Zilberstein, Abhishek Gupta, Pieter Abbeel, and Stuart Russell. Tractability of planning with loops. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[149] Hao Sun, Zehui Meng, Pey Yuen Tao, and Marcelo H Ang. Scene recognition and object detection in a unified convolutional neural network on a mobile manipulator. In *ICRA*, pages 1–5. IEEE, 2018.

[150] Niko Sünderhauf, Feras Dayoub, Sean McMahon, Ben Talbot, Ruth Schulz, Peter Corke, Gordon Wyeth, Ben Upcroft, and Michael Milford. Place categorization and semantic mapping on a mobile robot. In *ICRA*, 2016.

[151] Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[152] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in Neural Information Processing Systems*, 2021.

[153] Da Tang, Xiujun Li, Jianfeng Gao, Chong Wang, Lihong Li, and Tony" Jebara. Subgoal discovery for hierarchical dialogue policy learning. In *EMNLP*, pages 2298–2309, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[154] Ashish Tawari, Praneeta Mallela, and Sujitha Martin. Learning to attend to salient targets in driving videos using fully convolutional RNN. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3225–3232. IEEE, 2018.

[155] M. Tomono. 3-D Object Map Building Using Dense Object Models with SIFT-based Recognition Features. In *IROS*, pages 1885–1890, Oct 2006.

[156] Zhehang Tong, Dianxi Shi, and Shaowu Yang. Sceneslam: A slam framework combined with scene detection. In *ROBIO*, 2017.

[157] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. *ICCV*, 1:273, 2003.

[158] Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136, 1980.

[159] Alexander JB Trevor, John G Rogers, and Henrik I Christensen. Omnimapper: A modular multimodal mapping framework. In *ICRA*. IEEE, 2014.

[160] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017.

[161] Wenguan Wang, Jianbing Shen, and Fatih Porikli. Saliency-aware geodesic video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3395–3402, 2015.

[162] Wenguan Wang, Jianbing Shen, Hanqiu Sun, and Ling Shao. Video co-saliency guided co-segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(8):1727–1736, 2017.

[163] Wenguan Wang, Hongmei Song, Shuyang Zhao, Jianbing Shen, Sanyuan Zhao, Steven CH Hoi, and Haibin Ling. Learning unsupervised video object segmentation through visual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3064–3074, 2019.

[164] Wenguan Wang, Shuyang Zhao, Jianbing Shen, Steven CH Hoi, and Ali Borji. Salient object detection with pyramid attention and salient edges. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1448–1457, 2019.

[165] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement, 2021.

[166] Eric Wiewiora. *Reward Shaping*, pages 863–865. Springer US, Boston, MA, 2010.

[167] David E Wilkins. *Practical planning: extending the classical AI planning paradigm*. Elsevier, 2014.

[168] Melonee Wise Wise, Michael Ferguson, Derek King, Eric Diehr, and David Dymesich. Fetch and freight: Standard platforms for service robot applications. In *Workshop on Autonomous Mobile Service Robots, held at the 2016 International Joint Conference on Artificial Intelligence, NYC*, 2016.

[169] Jeremy M Wolfe, Kyle R Cave, and Susan L Franzel. Guided search: an alternative to the feature integration model for visual search. *Journal of Experimental Psychology: Human perception and performance*, 15(3):419, 1989.

[170] David Wooden. A guide to vision-based map building. *IEEE RAM*, 13(2):94–98, 2006.

[171] Mitchell Wortsman, Kiana Ehsani, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Learning to Learn How to Learn: Self-Adaptive Visual Navigation Using Meta-Learning. In *IEEE CVPR*, June 2019.

[172] Bohan Wu, Roberto Martin-Martin, and Li Fei-Fei. M-ember: Tackling long-horizon mobile manipulation via factorized domain transfer. *arXiv preprint arXiv:2305.13567*, 2023.

[173] J. Wu, H. I. Christensen, and J. M. Rehg. Visual place categorization: Problem, dataset, and algorithm. In *IROS*, 2009.

[174] Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. Tidybot: Personalized robot assistance with large language models. *arXiv preprint arXiv:2305.05658*, 2023.

[175] Ye Xia, Danqing Zhang, Jinkyu Kim, Ken Nakayama, Karl Zipser, and David Whitney. Predicting driver attention in critical situations. In *Asian Conference on Computer Vision*, pages 658–674. Springer, 2018.

[176] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *ICCV*, 2013.

[177] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[178] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057, 2015.

[179] Hao Yang and Jianxin Wu. Object templates for visual place categorization. In *ACCV*, pages 470–483, 2012.

[180] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. *arXiv preprint arXiv:1810.06543*, 2018.

[181] Xin Ye, Zhe Lin, Haoxiang Li, Shibin Zheng, and Yezhou Yang. Active object perceiver: Recognition-guided policy learning for object searching on mobile robots. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6857–6863. IEEE, 2018.

[182] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. BDD100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018.

[183] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[184] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[185] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *IEEE CVPR*, pages 5532–5540, 2017.

[186] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[187] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *European Conference on Computer Vision*, 2022.

[188] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *ICRA*, pages 3357–3364. IEEE, 2017.