Leveraging Sparse and Dense Features for Reliable State Estimation in Urban Environments

Noha Radwan

Technische Fakultät
Albert-Ludwigs-Universität Freiburg

Dissertation zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften

Betreuer: Prof. Dr. Wolfram Burgard

Leveraging Sparse and Dense Features for Reliable State Estimation in Urban Environments

Noha Radwan

Dissertation zur Erlangung des akademischen Grades Doktor der Naturwissenschaften
Technische Fakultät, Albert-Ludwigs-Universität Freiburg

Dekanin	Prof. Dr. Hannah Bast
Erstgutachter	Prof. Dr. Wolfram Burgard Albert-Ludwigs-Universität Freiburg
Zweitgutachter	Prof. Dr. Roland Siegwart Eidgenössische Technische Hochschule Zürich
Tag der Disputation	27. Juni 2019

Zusammenfassung

Ein Ziel der Forschung im Bereich mobile Robotik ist die Entwicklung intelligenter Plattformen, die verschiedene Aufgaben im Alltag ihrer Nutzer erledigen können. Im letzten Jahrzehnt sind Roboter immer mehr Bestandteil unseres Lebens geworden, wo sie Aufgaben in vielen Umgebungen übernehmen, wie etwa in der Fertigung und Montage im industriellen Kontext, der Heimassistenz und Bildungsarbeit innerhalb von Gebäuden, oder dem Rasenmähen und der Paketauslieferung in Außenbereichen. Trotz der signifikanten Fortschritte in diesen Anwendungsgebieten bleibt der zuverlässige Einsatz von Robotern in urbanen Umfeldern eine Herausforderung.

Um das Ziel einer allgegenwärtigen Robotik zu erreichen, ist es für mobile Roboter essentiell ihren eigenen Zustand und den anderer Agenten in ihrer Nähe zu schätzen. Damit dieses Ziel erreicht werden kann müssen jedoch mehrere Hürden bewältigt werden. Die Wahl der Sensormodalität für die Informationsbeschaffung über die Umgebung spielt eine wesentliche Rolle in der Repräsentationsleistung des Lokalisierungsmoduls. Während Lidar-Sensoren geometrische Informationen bereitstellen können, so bieten Kameras eine kostengünstige Alternative mit hoher Farb- und Texturinformationsdichte, die unerlässlich für eine korrekte Bewertung des Umfelds sind. Die genaue Lagebestimmung eines Roboters ausschließlich mit Kameras ist jedoch, vor allem in urbanen Szenarien, eine schwierige Aufgabe. Der komplexe Aufbau städtischer Umgebungen mit vielen repetitiven Elementen und Glasfassaden der Gebäude machen eine zuverlässige Lokalisierung sehr schwierig. Des Weiteren erfordern die wechselnden Wetter- und Lichtbedingungen, sowie die gebäudebedingte häufig wechselnde Szenerie, eine konstante Wartung des Lokalisierungsmoduls um eine genaue Zustandsschätzung aufrechtzuerhalten. Urbane Umgebungen sind durch Fußgänger, Autos und Radfahrer üblicherweise von hoher dynamischer und stochastischer Natur. Diese Agenten bewegen sich üblicherweise entlang unterschiedlicher Trajektorien und folgen verschiedenen Verkehrsregeln. Daher wird auch deren Zustandsschätzung zu einer schwierigen Aufgabe. Die genannten Schwierigkeiten verhindern es ebenfalls für Experten genaue, generalisierende und handgefertigte Zustandsschätzer zu entwickeln, da es unmöglich ist alle potentiellen Szenarien zu antizipieren und Lösungen dafür zu programmieren. Einen vielversprechenden Lösungsansatz für dieses Problem bieten Roboter, die ihre informationsreiche Umgebung zu nutzen verstehen und mit semantischen, strukturellen und geometrischen Informationen ein zuverlässiges und genaues Modell zur Zustandsschätzung lernen.

In dieser Arbeit untersuchen wir die Problemstellung der zuverlässigen Zustandsschätzung

in urbanen Umgebungen durch neue Techniken, die diese Herausforderungen durch das Ausnutzen spärlich und häufig vorhandener Merkmale angehen. Wir präsentieren eine Lokalisierungsmethode nach dem Vorbild menschlicher Ortsbeschreibungen, die Textinformationen von visuellen Umgebungsdaten nutzt um die Position des Roboters mit Hilfe von öffentlich verfügbaren Karten zu schätzen. Dadurch erreicht unsere Methode einen globalen Anwendungsbereich mit niedrigen Bandbreiteanforderungen. Wir verwenden distanz- und sprachbasierte Metriken, um einen unbewegten Textausschnitt aus der Umgebung mit Orientierungspunkten in Karten zu assoziieren. Damit ist die von uns vorgestellte Technik die Erste, die Textinformationen zur zuverlässigen Positionsschätzung nutzt. Um auch ohne solche Information eine genaue Lagebestimmung zu gewährleisten, präsentieren wir zusätzlich eine visuelle Multitask-Lokalisierung, die Ähnlichkeiten zwischen der Lokalisierung, der Bestimmung der Eigenbewegung und der semantischen Szenensegmentierung nutzt, um damit eine Verbesserung jeder dieser Aufgaben zu erreichen. Der Einsatz von geometrischen und semantischen Zwangsbedingungen in unserem Netzwerk führt zu einer genaueren Lageschätzung, die geometrisch konsistent mit den Bewegungen des Roboters ist und gleichzeitig robust gegen Aliasing der Sensorik und schlechte Lichtverhältnisse bleibt. Als letzten Schritt führen wir eine multimodale, interaktionsbasierte Methode zur Verhaltensvorhersage ein, die für ein gegebenes Zeitfenster die Sicherheit einer Straßenüberquerung abschätzt. Dieser Beitrag erweitert bestehende Methoden zur Verhaltensvorhersage durch das Betrachten der Interaktionen und gegenseitigen Abhängigkeiten der Verkehrsteilnehmer, um simultan genaue Trajektorien für jeden Agenten vorherzusagen. Durch das Kombinieren dieser Pfade mit der Erkennung des Ampelsignals ist unser Model in der Lage die Sicherheit einer Straßenüberquerung unabhängig von der Art der Kreuzung abzuschätzen.

Wir demonstrieren mit umfangreichen Experimenten auf verschiedenen Benchmarks und auf realen Daten die Effektivität unserer Methoden im Schätzen des Zustands des Roboters und der wahrnehmbaren Agenten in seinem Umfeld. Des Weiteren zeigen wir empirisch die Generalisierung und Robustheit unserer Methoden im Kontext verschiedener Umgebungen und schwieriger sensorischer Bedingungen, wodurch wir den Pfad zu einem zuverlässigen, dauerhaften Einsatz autonom navigierender Roboter in unseren Städten einschlagen.

Abstract

An ultimate goal of mobile robotics research is the development of intelligent platforms that are capable of undertaking a variety of tasks in everyday life for their users. Over the previous decade, robots have become more integrated into our daily lives, performing tasks in numerous environments including industrial settings such as assembly and manufacturing, indoor scenes such as home assistance and educational tasks, and outdoor areas such as lawn mowing and parcel delivery. Despite the significant strides achieved in the various application areas, reliably deploying robots in urban environments remains an open challenge.

In order to realize the goal of ubiquitous robotics, the ability of mobile robots to reliably estimate their state as well as the state of the agents in their vicinity is crucial for their successful deployment. However, in order to achieve this goal, robots need to overcome several challenges. The choice of sensor modality employed for extracting information about the environment plays a major role in the representational capabilities of the localization module. While LiDAR sensors are able to provide geometric information of the environment, cameras provide a low cost alternative with rich color and texture information which is crucial for reasoning about the scene. However, accurately estimating the pose of the robot using only cameras is an arduous task especially in urban scenarios. The complex nature of urban environments due to the presence of multiple repetitive structures and glass facades of buildings render the task of reliable localization extremely challenging. Furthermore, the varying weather and illumination conditions in addition to the frequently changing nature of the scene due to constructions necessitates the constant maintenance of the localization module to enable accurate state estimation. Urban environments commonly are of a highly dynamic and stochastic nature caused by moving pedestrians, cars and cyclists. Each of these agents often traverses a different trajectory and obeys different traffic rules. This in turn makes the task of estimating the state of surrounding agents extremely challenging. The aforementioned challenges render highly accurate generalizable hand-crafted solutions to the state estimation problem unattainable, as it is infeasible for an expert to anticipate and pre-program solutions for all potential scenarios. A promising solution for this problem is robots that are able to leverage the abundant rich information in the environment such as semantic, structural and geometric information in order to learn models for reliable and accurate state estimation.

In this thesis, we address the problem of reliable state estimation in urban environments by introducing novel techniques that address these challenges through exploiting sparse

and dense features of the scene. Inspired by how humans describe their location in urban cities, we propose a visual localization method that leverages the textual information in the scene to estimate the location of the robot by utilizing publicly available maps. This enables our method to achieve global scale breadth and low bandwidth requirements. We employ distance and linguistic-based metrics to probabilistically associate stable text from the environment with landmarks in the map. Our proposed method is the first to utilize textual information from the scene to produce reliable position estimates. In order to enable accurate pose estimation in the absence of textual information, we propose a multitask visual localization method that leverages the inter-task similarities between localization, ego-motion estimation and semantic scene segmentation for the mutual benefit of each of these tasks. Incorporating geometric and semantic constraints into our network enables the prediction of accurate pose estimates that are geometrically consistent with the robot motion, while being tolerant to perceptual aliasing and adverse illumination conditions. Finally, we propose a multimodal interaction-aware behavior prediction method for predicting the safety of street intersections for crossing during a given time interval. Our contribution goes beyond existing behavior prediction approaches by leveraging the interaction interdependencies between the various traffic participants to simultaneously predict an accurate future trajectory for each participant. Furthermore, by utilizing the predicted motions along with recognizing the traffic light signal, our model can estimate the safety of a street intersection for crossing while being invariant to the type of intersection.

We present extensive experimental evaluations on several benchmarks as well as real-world datasets and show the effectiveness of our proposed methods in reliably estimating the state of the robot and all observable agents in its vicinity. Moreover, we provide thorough empirical evidence that demonstrates the generalization ability and robustness of our methods to different environments and challenging perceptual conditions, thus paving the way towards the reliable life-long deployment of robots that can autonomously navigate in our complex cities.

Acknowledgments

I would like to thank the many people without whose constant support, encouragement, and guidance this thesis would not have been possible.

First of all, I would like to thank my adviser Wolfram Burgard for giving me the opportunity to pursue my PhD at his lab, pushing me to explore new research ideas, and providing a nice and collaborative work environment. I highly appreciate his guidance and constant feedback throughout the years, as well as granting me the opportunity to attend multiple conferences around the world.

Throughout my time at the Autonomous Intelligent Systems (AIS) group, I had the privilege of learning from and working with many brilliant researchers. I would like to thank Cyrill Stachniss and Gian Diego Tipalidi for their constant encouragement, feedback, and for always taking the time to discuss and brainstorm. A huge thank you to Bastian Steder and Michael Ruhnke for guiding me through the EUROPA2 project, especially all the time they spent helping me get familiarized with the project code, and being patient with me at the early stages of using the Obelix robot. I thank Daniel Büscher and Christian Dornhege for their valuable feedback and precious insights during writing this thesis. I would like to thank Gabriel Oliveira for the fruitful discussions throughout the PhD. A huge thank you to Wera Winterhalter and Freya Fleckenstein for assisting with the final stages of the EUROPA2 project and the countless "outside/inside" experiments. I thank Abhinav Valada for his constant motivation, enthusiasm and valuable insights during our numerous collaborations.

Special thanks to all my officemates throughout the years for the fun conversations and putting up with my random rants: Nichola Abdo, Christoph Sprunk, Mladen Mazuran, Ayush Dewan, Manuel Watter, Jan Wülfing, Daniel Büscher and Jannick Zürn. I thank Wera Winterhalter, Freya Fleckenstein and Christian Dornhege for all the "closed door" discussions and the stress relief toys. A big thank you goes to Christoph Sprunk and Nichola Abdo for their sound advice throughout the years, always providing help and moral support. Special thanks to Federico Boniardi, Mladen Mazuran and Ayush Dewan for all the insightful mathematical discussions, to Gabriel Oliveira and Ayush Dewan for the much needed walks to Lidl and their fun discussions, and to Andreas Eitel and Sudhanshu Mittal for motivating me to stay healthy and exercise. I would like to thank Abhinav Valada for motivating and encouraging me throughout the years, and keeping me calm during the stressful deadlines. This thesis would not have been possible without you.

A big thank you to Susanne Bourjaillat, Evelyn Rusdea and Michael Keser for handling

all administrative and technical issues during the PhD period as well as always being very kind and supportive throughout this time.

I would like to thank all the students that I had the opportunity to co-supervise and work with at AIS. Specifically, I thank Claas Bollen for his efforts and hard work. A big thank you to Ekatarina Polkh for her dedication, hard work and always going the extra mile. I would like to thank all of my colleagues for making work more enjoyable and setting the ideal atmosphere for both productivity and fun: Tim Caselitz, Tayyab Naseer, Tim Welschehold, Johan Vertens, Alexander Schaefer, Lukas Luft, Johannes Meyer, Chau Do, Marina Kollmitz, Nicolai Dorka, Micheal Krawez, Niklas Wetzel and Oier Mees.

I am very grateful to my friends for being there for me regardless of where I was. A big thank you goes to Eteri Sokhoyan, Christoph Schöller, Achim Leydecker, Aaron Klein, Simon Bartels, Julia Berger, Diego Saul, Basma Hassabo, Rawan Khalil, Khaled Hassan, and Noha Abdelghani.

Above all I thank my parents, my grandparents, my sister and my brother for believing in me and loving me unconditionally.

Contents

Chapter 1

Introduction

From the vast depths of the oceans to the peaks of the highest mountains, humans have had the desire to explore since the beginning of time, whether for survival reasons such as finding better hunting grounds, or in search for better living conditions. Satisfying this desire to explore, however, meant finding solutions to the question of how to reach specific locations (navigation), which in itself entails sufficient knowledge of your position along the traversed path (localization). The earliest localization methods dating back to 3000 BC used the locations of stars for navigating across seas, oceans and deserts [38]. During the middle ages, tools such as the magnetic compass and the kamal were used to aid in celestial navigation by providing rudimentary measures for the positions of the stars [129]. Over time, the tools enabling celestial navigation evolved from using quadrants and astrolabes to cross staffs and sextants around the early 1700s [55].

The 1920s featured the installation of radio beacons [40]. Using radio sets, accurate localization was made possible by calculating the distance to the nearest known transmission station. Radio-based localization, however, came with the disadvantage of requiring relatively flat regions to enable accurate localization estimates. In order to circumvent this issue, the Global Positioning System (GPS) was launched in the early 1970s as a satellite-based radio navigation system [59]. It enabled users with a GPS receiver to gain accurate knowledge of their location with less interference from mountains and buildings compared to radio beacons. Currently, GPS is employed in multiple domains such as geofencing for tracking animals, persons or vehicles [110], cartography [27], disaster relief [161], aircraft tracking [5, 76], and navigation [198] whether using a smartphone or an automotive navigation system [100].

The 1900s also witnessed the introduction of the concept of a robot. The term “robot” was first introduced in a satire play, referring to beings that performed all unpleasant manual labor [109]. Over the remainder of the century, the idea of employing robots to carry out labor-intensive, mundane or dangerous tasks gained popularity, with the goal of increasing operation safety and saving time. The first industrial robot was introduced in the 1960s by Unimation [1]. The robot undertook the dangerous task of transporting die castings from an assembly line to a welding station, where it welded the parts onto auto bodies. As industrial robots evolved, they became more integrated into factories [2, 3]. However, their operation remained mostly confined to designated areas where they carried out tasks with a set of pre-programmed actions. Deploying robots in more complex environments to perform tasks that entail navigating from one location to another necessitates developing robust methods to reliably estimate the state of the robot and other agents in its vicinity amongst other challenges.

After several years of battling these challenges, teams from Carnegie Mellon University were the first to develop a robot that could autonomously navigate across cities. Using GPS and gyroscope information, their autonomous vehicle was able to navigate from Pittsburgh to San Diego for 98% of the $2{,}800$ mile journey. Despite this successful demonstration of utilizing GPS as a solution to the localization problem, employing GPS solely is insufficient for reliably deploying robots in urban environments. This comes as a consequence of the dense structure of urban cities and the presence of high rises which cause a degradation in the quality of the GPS signal. Following the advances realized through utilizing GPS, Simultaneous Localization and Mapping (SLAM) frameworks have enabled further progress in the field of robotics by facilitating state estimation using a map that is incrementally built while navigating the environment [183]. By incorporating the map building process coupled with loop closure detection, SLAM approaches have revolutionized state estimation methods by reducing trajectory drift that accumulates over time. Utilizing SLAM for state estimation has brought many advances including RHINO the interactive museum tour-guide robot [47], robotic vacuum cleaners [4], lawnmowers [7], self-driving cars [6, 10, 14] and, more recently, delivery robots [9, 11, 13]. Each of these innovations employed more advanced systems for navigation and state estimation, improving upon the state of the art and opening up new possibilities. Despite the generalizability of SLAM to the type of sensor used, LiDAR sensors are most commonly used due to the high accuracy of the depth estimates provided by these sensors which in turn facilitates accurate estimation of the pose of the robot [49, 123, 162]. Nonetheless, the cost of LiDAR sensors prohibits the widespread production and retail of mobile robots using them, which has consequently lead to an increased interest in camera-based state estimation methods. Unlike LiDAR-based state estimation, camera-based approaches present a low-cost and an affordable solution. However, achieving comparable localization accuracy to LiDAR-based methods using camera-based approaches remains an open challenge which is essential for the large scale deployment of robots. In this thesis, we introduce several frameworks to address the fundamental problem of *reliable state estimation for mobile robots by leveraging the inherent semantic and geometric structures from the sensor data* with the goal of *enabling efficient and reliable deployment in urban environments.*

Current predictions indicate that autonomous vehicles will make over 80% of last-mile deliveries by 2025 [12]. However, several challenges remain preventing the widespread deployment of robots in everyday life. In order to reliably localize in dense urban cities, the GPS signal alone is insufficient due to signal degradation and outages in the vicinity of skyscrapers. Currently, the majority of localization approaches employ highly precise maps that are custom-built by mapping service providers [17, 18, 20]. However, due to the constantly changing nature of the environment, the maps need to be frequently updated in order to enable accurate localization. Moreover, utilizing a pre-existing map for localization leads to confining the autonomous operation of the robots to the areas mapped beforehand. This has limited companies that deploy robots for pickup and delivery of packages, such as Hermes, to operate within a small region of the city which is guaranteed to have an up-to-date map [19]. Additionally, the presence of multiple dynamic objects such as pedestrians, cars and bikes makes the localization task even more challenging, since to enable reliable localization, it is required to distinguish between the stable features such as those belonging to buildings, and features belonging to dynamic or potentially movable objects such as cars. Semantic information in the form of street and shop signs is, however, abundant in dense urban cities and is used by humans to localize and navigate in their daily lives. Effectively leveraging this semantic information can aid in robustly localizing in text-rich urban cities.

Among the challenges faced for making robots ubiquitous is the choice of sensors used for state estimation. Despite the accuracy of LiDAR-based localization methods [39, 107, 128], glass structures such as windows which commonly occur in urban environments disturb the localization accuracy of the system [186]. Similarly, seasonal, weather and illumination conditions such as snow, presence of fallen leaves or cast shadows disturb the accuracy of camera-based localization methods [175]. Currently, the minimal set of sensors required to enable accurate and efficient localization remains to be defined [44]. While current approaches for camera-based localization that employ hand-engineered features with a 3D model of the environment are able to provide accurate localization estimates in certain conditions, such methods, however, do not generalize well to varying weather and seasonal appearance changes [130, 167]. Furthermore, local feature-based localization methods require expert knowledge for designing and choosing features that are representative of the deployment environment, which in turn reduces the robustness of these methods when applied in different environments than the ones for which the features were designed. On the other hand, although deep learning-based localization methods are able to generalize to varying environments, adverse weather and seasonal conditions, they require a large amount of training data and achieve lower localization accuracy in comparison to local feature-based methods as they cannot encode the geometric features of the environment into the network [95, 138]. The major challenge lies in the ability to combine the advantages of both localization approaches by encoding the geometric and semantic structural information of the environment into the localization network, thereby enabling accurate localization that is robust to weather and illumination conditions. Naively encoding the structural information through concatenation leads to suboptimal performance as over time the network accumulates irrelevant information. Similarly, defining a fixed feature set from the semantics of the environment and discarding information from the remaining structures restricts the operation to pre-defined environments that follow the hand-crafted definition. Learning the set of stable semantic features in a self-supervised manner will enable the network to utilize this information more efficiently, while facilitating deployment in new environments.

Ubiquitous robotic deployment will eventually lead to the presence of robots with various purposes navigating the streets and sharing the living space with humans. In order for robots to be safely deployed among humans, their ability to infer the state and future motion of humans and agents in their vicinity is crucial. In most situations humans have the ability to predict the future motion of those surrounding them without the need for explicit communication. Current research methods, in the area of computational motion prediction, rely on some mode of communication between the mobile robots for instance at street intersections to signal the right-of-way [50, 131]. This however, requires the standardization of the communication protocol among all manufacturers, which is infeasible. Furthermore, additional protocols need to be devised for facilitating communication between robots and humans. Alternative approaches for estimating the future trajectories of nearby pedestrians and vehicles manually define a set of behavioral rules [84], rank the significance of surrounding pedestrians and vehicles in order to approximate motion dependencies and interactions [197], or estimate the behavior of a single pedestrian or vehicle regardless of their surroundings [179]. Employing such methods prevents the deployment of robots in new environments without manually adjusting the set of rules, which is both taxing and infeasible at a large scale. Furthermore, they are unable to accurately estimate the future motion [115] or have slow interaction times [77]. This can lead to dire consequences as current records from the California Department of Motor Vehicles (DMV) for 2018 show that two thirds of the filed autonomous vehicle collisions are rear-ending accidents, with 78% of the accidents occurring with the vehicle in fully autonomous mode [16]. The statistics suggest that the manner in which the autonomous vehicles currently behave is different from what human drivers would expect. Another example can be seen in San Francisco where citizens petitioned against the operation of delivery robots on sidewalks due to the operating speeds of the robots and their slow interaction times which renders them as a source of threat for the nearby pedestrians [15]. Among the core challenges of safely navigating across intersections is the ability to estimate the motion of the nearby pedestrians and vehicles in real-time. As opposed to manually defining a set of behavioral rules that do not generalize well to new scenarios, leveraging the observed motion for all observable pedestrians and vehicles is expected to enable the robot to learn a prediction model that is more aware of its surroundings. Hence, enabling the prediction of safer and more accurate trajectories that resemble human behavior. Concurrently, leveraging information from multiple modalities such as LiDARs, radars, and cameras can aid in making decisions that are robust to failures from each single modality, which is crucial when navigating street intersections.

Considering the aforementioned challenges that face the deployment of robots in urban environments, we pose the following research questions that we address in this thesis:

- How can a robot accurately localize in constantly changing urban environments without the need to frequently update the map? How can we leverage the abundant textual semantic information from the environment in order to enable robust localization?

- How do we enable a robot to robustly localize in challenging perceptual conditions? Accordingly, how can we encode geometric and semantic structural features into a convolutional neural network architecture to enable robust localization with online run-time capabilities?

- How can we enable a robot to learn to estimate the motion of all observable agents in the scene without explicitly modeling the interdependencies in their motion? Moreover, can we leverage this information to enable robots to learn to safely navigate across street intersections?

In the scope of this thesis, we tackle the aforementioned questions and provide solutions, which outperform current state-of-the-art methods in terms of both run-time performance and benchmarking metrics.

1.1 Key Contributions

In this thesis, we present several contributions to the field of robotics research by developing solutions for state estimation problems in urban environments. Our contributions address the tasks of visual localization and motion prediction by providing solutions that enable reliable, efficient, and robust state estimation for mobile robots. Firstly, we tackle the problem of localization in urban environments by leveraging the abundant textual semantic information. We employ a probabilistic particle filter-based method with dedicated sensor models to estimate the position of the robot using publicly available maps, thus rendering our method robust to frequent changes in the environment. Next, in order to facilitate pose estimation in the absence of textual features, we propose a multitask convolutional neural network to simultaneously predict the semantics, ego-motion and global pose of the robot by exploiting the interdependencies among the tasks. Jointly learning all three tasks enables our method to aggregate the geometrical and structural features of the scene, which in turn improves the accuracy and reliability of our approach. Finally, we propose a multimodal framework for learning to predict the safety of street intersections for crossing by jointly predicting the motion of all observable traffic participants and recognizing the traffic light signal. Utilizing information from both modules renders our approach generalizable to different environments, with online run-time capabilities. We briefly highlight each of these contributions in the remainder of this section as well as list them at the end of the introduction of each respective chapter.

Visual Localization using Textual Information In Chapter 3, we address the problem of localizing in urban environments by leveraging abundant textual features from street signs and shop fronts within the environment. Our approach is the first to utilize textual information from scene images to enable the localization of mobile robots in urban environments. We first extract text from natural scene images and utilize a data association method to select a number of potential matching landmarks which are then used for probabilistically estimating the 2D location of the robot. We propose two variants of our approach: a single-shot global localization variant to estimate the position at a single timestep, and a pose tracking variant which utilizes the odometry information of the robot to maintain a tracked estimate over the various timesteps. As our method employs publicly available maps for extracting the landmarks, it can be robustly applied in new environments without the need for an initial mapping step. Leveraging the textual features further renders our method both robust to appearance changes due to weather and season, and easy to deploy in resource constrained systems. Through extensive experimental evaluations in three cities, we demonstrate the effectiveness and robustness of our proposed method.

Geometrically Consistent Semantic Visual Relocalization In order to enable the robust relocalization of mobile robots, independent of the presence of textual features in the environment, we propose to leverage geometric and semantically stable features from the scene in a multitask learning framework. We propose a convolutional neural network architecture to jointly estimate the 6-DoF pose, 6-DoF odometry and pixel-wise semantic segmentation of the scene in Chapter 4. Our approach combines the advantages of local feature-based methods and convolutional neural networks for localization. We propose a novel fusion scheme to enable the incorporation of semantic and geometric features into the pose regression network by temporally aggregating motion-specific features and semantic representations of the scene. In order to enable the network to fully leverage the geometric information, we propose a novel loss function that enforces the consistency of the predicted poses with the motion of the robot. We additionally propose a self-supervised warping method to enable the aggregation of temporal features into the semantic segmentation stream, thereby enabling the prediction of accurate segmentation masks. We introduce two challenging datasets to facilitate the evaluation of our proposed method in varying outdoor environments, as well as provide qualitative and quantitative analysis on a publicly available indoor benchmarking dataset. Experimental evaluations demonstrate that our approach is the first deep learning-based method to outperform state-of-the-art techniques while simultaneously predicting multiple tasks and achieving fast run-time, hence enabling reliable online deployment. Our extensive ablation studies further demonstrate the robustness of our method to perceptual aliasing, dynamic obstacles and motion blur, and provide insights on the representations learned and the generalization capabilities of the networks.

Interaction-Aware Motion Prediction With the goal of enabling the autonomous navigation of mobile robots in urban environments, in Chapter 5 we propose a multimodal convolutional neural network framework for predicting the safety of street intersections for crossing. Our architecture consists of two sub-networks: an interaction-aware motion prediction stream and a traffic light recognition stream. Our motion prediction sub-network is the first deep learning-based behavior prediction method to estimate the future trajectories of all observed dynamic agents by utilizing causal convolutions and accounting for the complex interactions among the various agents. Our traffic light recognition sub-network employs an attention-based method to provide an estimate for the state of the traffic light. We fuse the learned representations from both sub-networks to estimate the safety of the street intersection for crossing. Incorporating the uncertainties in the predictions of each sub-network enables our crossing predictor to learn a probabilistic function for deciding the safety of a street intersection for crossing, while being robust to mispredictions from either sub-network. Additionally, as we utilize information from both the motion prediction and the traffic light recognition sub-networks, our approach is robust to the type of intersection and does not require prior knowledge of the environment. In order to facilitate the evaluation of our approach, we introduce a first-of-a-kind dataset captured at various intersections. We provide extensive evaluations for each of the sub-networks as well as the entire framework on multiple publicly available benchmarks, which demonstrate the accuracy of the predictions of each component. Our proposed method generalizes well to various environments, achieves fast run-time and requires small storage space thereby making it efficiently deployable in an online manner while achieving state-of-the-art accuracy for motion prediction.

1.2 Publications

Major parts of the work presented in this thesis have undergone or are currently undergoing international peer review. In the following, we list the corresponding publications in chronological order.

- N. Radwan, G. D. Tipalidi, L. Spinello, and W. Burgard. Do you see the Bakery? Leveraging Geo-Referenced Texts for Global Localization in Public Maps. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2016.

- N. Radwan, W. Winterhalter, C. Dornhege, and W. Burgard. Why Did the Robot Cross the Road? - Learning from Multi-Modal Sensor Data for Autonomous Road Crossing. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

- A. Valada[*], N. Radwan[*], and W. Burgard. Deep Auxiliary Learning for Visual Localization and Odometry. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2018.

- A. Valada[*], N. Radwan[*], and W. Burgard. Incorporating Semantic and Geometric Priors in Deep Pose Regression. In *Proc. of the Workshop on Learning and Inference in Robotics: Integrating Structure, Priors and Models at Robotics: Science and Systems (RSS)*, 2018.

- N. Radwan[*], A. Valada[*], and W. Burgard. VLocNet++: Deep Multitask Learning for Semantic Visual Localization and Odometry. In *IEEE Robotics and Automation Letters (RA-L)*, 2018.

- N. Radwan, and W. Burgard. Effective Interaction-aware Trajectory Prediction using Temporal Convolutional Neural Networks. In *Proc. of the Workshop on Crowd Navigation: Current Challenges and New Paradigms for Safe Robot Navigation in Dense Crowds at IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

- N. Radwan, A. Valada, and W. Burgard. Multimodal Interaction-aware Motion Prediction for Autonomous Street Crossing. In *arXiv preprint arXiv:1808.06887, Int. Journal of Robotics Research (IJRR)* (Under Review), 2018.

Furthermore, the following publications of the author of this thesis present work related to localization and multitask robot learning. However, they are outside of the scope of this thesis and thus are not covered.

- G. Oliveira[*], N. Radwan[*], W. Burgard, and T. Brox. Topometric Localization with Deep Learning. In *Proc. of the Int. Symposium on Robotics Research (ISRR)*, 2017.

- W. Burgard, A. Valada, N. Radwan, T. Naseer, J. Zhang, J. Vertens, O. Mees, A. Eitel and G. Oliveira. Perspectives on Deep Multimodal Robot Learning. In *Proc. of the Int. Symposium on Robotics Research (ISRR)*, 2017.

1.3 Collaborations

This thesis covers work that was the outcome of collaborations with other researchers. Prof. Wolfram Burgard contributed through scientific discussions as the supervisor of this thesis. We outline below the collaborations beyond this supervision.

[*] Denotes equal contribution

- Chapter 3: The localization system presented in this chapter is an extension of the author's master thesis which was supervised by Luciano Spinello. The chapter builds on the same concept of localizing using a publicly available map and textual information from the scene, which the author of this thesis extensively expanded. The extensions include the introduction of two different approaches for localization, the introduction of two data association methods which capture the lexical relations between the map text and the extracted words, extensions of the datasets for which this method was evaluated as well as extensive experiments evaluating the performance of each of the proposed components. The work on the Single Shot Localization method was co-supervised by Gian Diego Tipaldi and Luciano Spinello.

- Chapter 4: Both the VLocNet and VLocNet++ architectures presented in this chapter are a result of collaboration with Abhinav Valada, who contributed the main architectural topologies as well as the weighted fusion layer. The formulation of the Geometric Consistency loss function and the self-supervised warping layer were developed by the author of this thesis. The experimental evaluation was further expanded by the author of this thesis through the addition of a novel challenging dataset containing multiple dynamic objects that we make publicly available. The related publications for this chapter are Valada *et al.* [193] and Radwan *et al.* [157].

- Chapter 5: This chapter is a result of the fruitful discussion with Abhinav Valada, Christian Dornhege and Wera Winterhalter. Christian Dornhege and Wera Winterhalter contributed during the early stages of this work in the formulation of the baseline Random Forest classifier. The implementation of the Naive Crossing Predictor baseline was realized by Wera Winterhalter. The subsequent work on the IA-TCNN, AtteNet and ACP architectures was carried out entirely by the author of this thesis.

Chapter 2

Background Theory

In this chapter, we briefly describe some of the basic concepts and theoretical foundations for the methods presented later in this thesis. We begin by introducing the robotic platform used for both conducting real-world experiments and capturing the datasets used in Chapter 4 and 5. Next, we describe the mapping of three dimensional world coordinates to the two-dimensional pixels in the camera model, followed by a brief mathematical overview of the particle filter algorithm for robot localization. We then briefly describe the constituting blocks of artificial neural networks, loss functions as well as some of the popular architectures. Finally, we conclude this chapter by detailing the auxiliary learning architecture for visual localization and ego-motion estimation which we build upon in Chapter 4.

2.1 Robotic Platform

We utilize the robotic platform shown in Figure 2.1 to carry out the real-world experiments conducted in this thesis. The robot was custom built for pedestrian assistance [108]. It is equipped with multiple laser scanners including SICK LMS 151, Velodyne HDL-32E and a Hokuyo UTM-30LX which are mainly used for mapping and localization. Furthermore, an XSens IMU and a Trimble GPS Pathfinder Pro are employed to provide information during the localization and mapping tasks, and a Bumblebee camera for visualization. In addition to the aforementioned sensors, we further equipped the robot with a ZED stereo camera for capturing stereo and depth images which were used in Chapter 4 for estimating the robot pose and understanding the observed scene. We mounted two Delphi ESR radar sensors to the left and right of the robot as shown in Figure 2.1. Each radar provides both wide angle coverage at mid-range and high resolution coverage at long-range. The radars are designed specifically for the automotive industry, allowing the detection and tracking of adjacent vehicles and pedestrians across the width of the equipped platform. The long-range coverage can identify vehicles up to $174\mathrm{m}$ with a field-of-view $\pm 10°$, while the mid-range coverage has a shorter range of only $60\mathrm{m}$ but a larger field-of-view $\pm 45°$. The radars were added with the goal of enabling the detection and tracking of

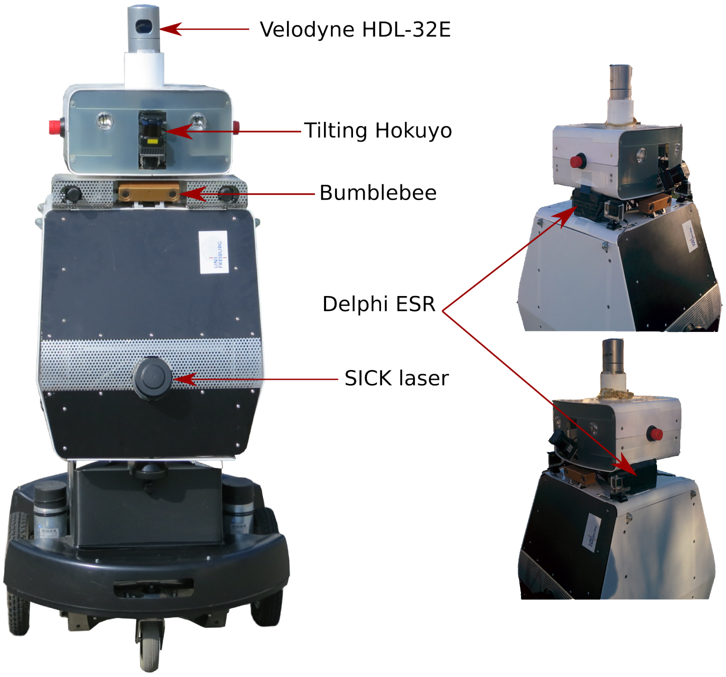


Figure 2.1: Our robotic platform (Obelix) that was used for capturing the different datasets and conducting real-world experiments.

objects at long ranges such as vehicles at a road intersection and will be used in Chapter 5.

2.2 Camera Model

In this section, we describe the pinhole camera model, in which a scene view is transformed by projecting 3D points M to the 2D image plane m using a perspective transformation [8] $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ such that:

$$\begin{aligned} \pi(M) &= K\,[R \mid t]\,M \\ \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \end{aligned} \tag{2.1}$$

where K is the intrinsic camera matrix with focal length f and optical center C. The joint rotation-translation matrix $[R \mid t]$ is used to describe the camera motion in the scene or the motion of objects in front of a static camera.

Figure 2.2 depicts a simple pinhole camera model. The line perpendicular to the image plane that passes through the optical center is referred to as the principal axis, and the intersection point of the image plane with the principal axis is called the principal point.

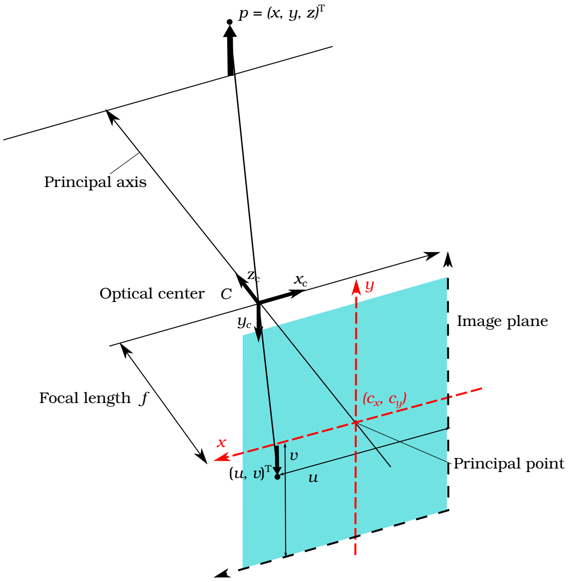


Figure 2.2: Illustration of the pinhole camera model describing the transformation relationship between a point in three dimensional world coordinates $p = (x, y, z)^T$ and its corresponding two dimensional projection $(u, v)^T$ on the image plane [8].

The distance between the principal point and the optical center defines the focal length of the camera. We employ the previously described equation in Chapter 4 to temporally transform images.

2.3 Particle Filter

Among the motivations of employing the particle filter algorithm for robot localization [70] is its ability to deal with arbitrary distributions outside the scope of the well known normal distributions. This ability is facilitated by the definition of the particle filter itself, wherein the posterior distribution of the robot $p(\mathbf{x})$ is represented by a set of weighted samples which are referred to as particles:

$$\chi = \left\{ \left\langle \mathbf{x}^j, w^j \right\rangle \right\}, \tag{2.2}$$

for $j \in \mathcal{N}$, where j is the number of particles. Each particle represents a specific belief over the pose of the robot and the importance weight over that belief, where the weights sum up to one.

The particle filter is a non-parametric recursive Bayes filter approach, and as such estimates the posterior distribution $p(\mathbf{x}_t)$ of the state of the robot at time t given the prior distribution $p(\mathbf{x}_{t-1})$. Ideally, one would sample the particles directly from the underlying posterior distribution, however, in robotic applications this is often not possible. In this thesis, we employ the importance sampling principle [73, 165], which allows us to estimate the posterior distribution by sampling particles from a different distribution and weighing them using the ratio between both distributions. More precisely, given a target distribution $p(\mathbf{x}_t)$ representing the posterior of the robot, we draw samples from the proposal distribution $q(\mathbf{x}_t)$ in the prediction phase. In the correction phase, each sample j is assigned an importance weight to account for the difference between the target and the proposal distribution such that

$$w_t^j = \frac{p(\mathbf{x}_t^j)}{q(\mathbf{x}_t^j)}. \tag{2.3}$$

Once the importance weights have been computed, the resampling step needs to be carried out. During resampling, the particle set is updated by drawing samples out of it, each with a probability equal to the importance weight. The goal of the resampling step is to ensure that the density of the final particle set is representative of the posterior/target distribution.

In the context of robot localization, the particle filter algorithm can be employed to estimate the posterior distribution over the full trajectory of the robot. Thus each particle does not only contain the current pose information, but rather the entire trajectory history. In the prediction step, the motion model is used as the proposal distribution for sampling the particles such that:

$$\mathbf{x}_t^j \sim p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, u_t), \tag{2.4}$$

where $\mathbf{x}_t$ is the current pose of the robot, and u_t is the current odometry/motion command. In the correction step, the importance weight of the particles is computed via the observation model such that:

$$w_t^j \propto p(\mathbf{z}_t \mid \mathbf{x}_t, m), \tag{2.5}$$

where $\mathbf{z}_t$ is the current set of observations, and m is the map. The resampling step remains the same as in the original algorithm. There exist multiple methods for resampling, in the scope of this thesis, we employ the stochastic universal sampling method [30]. In Chapter 3, we employ the particle filter algorithm for robot localization in urban environments utilizing textual features from the scene.

2.4 Feed-Forward Neural Networks

Feed-forward neural networks or multi-layered perceptrons are the first artificial neural networks proposed. They consist of an input layer, a number of hidden layers and an output layer with information flowing in a single direction without loops between the input and output layers. Figure 2.3 shows a three-layer feed-forward neural network with multiple inputs, two hidden layers and two outputs. Each neuron or perceptron learns a weighted sum of its inputs and activates if the output value is higher than a certain threshold using a non-linear activation function. Formally, a single perceptron can be

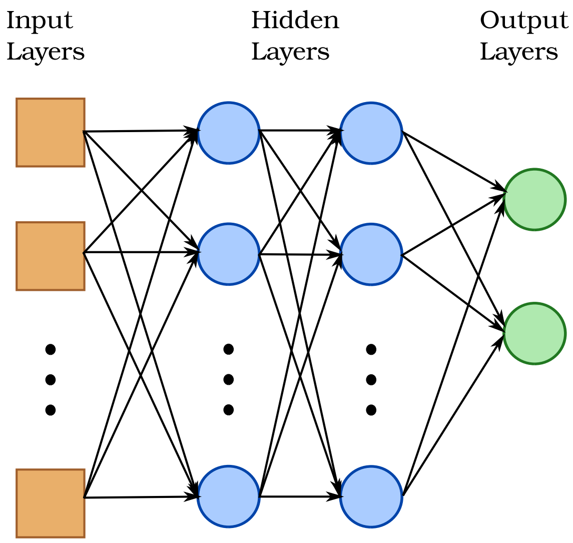


Figure 2.3: Depiction of a three-layer perceptron with multiple inputs, two hidden layers and two outputs.

modeled as such:

$$f(x) = \sigma\left(\sum_{i=1}^{N} w_i x_i + b\right), \tag{2.6}$$

where N is the number of inputs, w_i is the learned weight for each input x_i, b is the added bias and σ is the non-linear activation function enabling the perceptron to learn a non-linear mapping from the inputs to the output.

Within a single layer in a feed-forward neural network such as the one shown in Figure 2.3, there are no connections passing the information from one perceptron to the other. Furthermore, all perceptrons within the same layer use the same activation function, with the exception of the output layer which does not employ an activation function. Similar to the mathematical representation of the output of a single perceptron, the output of a layer l can be defined as:

$$f^l(x) = \sigma^l\left(\mathbf{W}^l \mathbf{x} + \mathbf{b}^l\right), \tag{2.7}$$

where $\mathbf{W}^l \in \mathbb{R}^{m \times n}$ is the l-th layer weight matrix with m equal to the number of neurons in the layer and n the number of input channels in x, $\mathbf{b}^l \in \mathbb{R}^m$ is the corresponding biases and σ^l denoting the activation function employed for the l-th layer. Common activation functions employed usually include tanh, sigmoid, Rectified Linear Units (ReLU) [137] and Exponential Linear Units (ELU) [56]. The intrinsic parameters of the network such as weights, biases and activation functions are often encapsulated in a single term θ with the predicted output of the network defined as $\hat{y} = f(x \mid \theta)$.

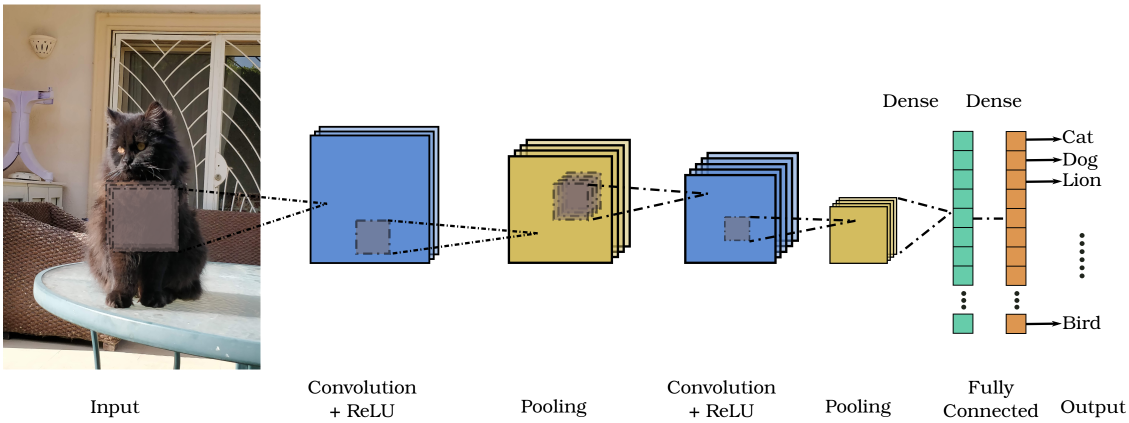


Figure 2.4: A depiction of a convolutional neural network architecture. The network consists of alternating convolution and pooling layers which learn the spatial structure of the input image, followed by a fully connected layer to integrate the global information into the output layer.

2.5 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a shift invariant variation of the feed-forward neural networks. Inspired by biological processes, CNNs are designed to require minimal pre-processing with connectivity patterns between the neurons resembling the organization of the visual cortex in animals. Individual neurons within CNNs only have access to parts of the input or what is known as the receptive field. Furthermore, the receptive fields of the different neurons overlap such that they fully cover the entire visual field. Moreover, unlike multi-layered perceptrons, the number of parameters in CNNs does not grow quadratically with the size of the network which enables the network to represent more complex functions with fewer parameters. A standard convolutional neural network as shown in Figure 2.4 is comprised of a series of convolutional and pooling layers which are optionally followed by recurrent layers or fully connected layers [104, 176]. In the following, we begin by discussing the basic constituting layers of a CNN, followed by some standard architectures and the cost functions that are used for training the networks.

2.5.1 Layers

In this section, we describe the fundamental layers which act as building blocks for creating the various architectures proposed in this thesis.

Convolution Layer: The convolutional layer is the core building block of the CNN architecture and consists of a set of learnable kernels (filters) with a small receptive field which extend through the entire depth of the input. During a forward pass, the dot product of the filters and the input entries is computed across the height and width of the input volume to produce a two-dimensional activation map representing the spatial response of the kernel at each location. Unlike fully connected layers, the learnable filters in a convolutional layer share parameters across the depth of the input volume, thereby conserving the number of learned parameters. The output tensor is then computed by stacking the activation maps across the depth dimension. The size of the output tensor is controlled by three hyperparameters; depth, stride and zero-padding. The depth parameter controls the number of filters used during the convolution operation; each of which learns to activate for different features in the input image. The stride defines the number of pixels with which the kernel matrix is shifted at a time; with a stride value of one referring to moving the kernel one pixel at a time. The zero-padding parameter controls the size of the output volume by adding zeros around the borders of the input tensor. Given an input tensor of size W, its output dimension Y can be computed as:

$$Y = \lfloor \frac{W - F + 2P}{S} \rfloor + 1, \tag{2.8}$$

where F denotes the size of the receptive field, S the stride and P denotes the zero-padding.

Dilated Convolution Layer: A drawback of the convolution operation is that the receptive field size grows only linearly with the number of layers. This can be very limiting when learning tasks where the information from several scales is crucial for instance in semantic segmentation or object detection. Dilated convolutions, also known as atrous convolutions were proposed to alleviate this problem by introducing a *dilation rate* parameter to convolutional layers [207]. The dilation rate specifies the spacing between the kernel values, thus increasing the receptive field while maintaining the same computational cost. Using dilated convolution, a 3×3 kernel with a dilation rate of 2 has the same receptive field as a 5×5 convolutional layer. Furthermore, by stacking multiple dilated convolutions with increasing dilation rates, the effective receptive field increases exponentially without an exponential increase in the number of parameters.

Pooling Layer: In order to reduce the spatial dimensions of the representations learned, pooling layers are employed within network architectures. They are most commonly inserted between successive convolution layers within a CNN in order to reduce the number of parameters of the network and hence control over-fitting, as well as provide translation invariance to the representations learned by the network. The pooling layer operates on individual depth slices of the input resizing them spatially using the specified pooling operation such as max, average, L2-norm, stochastic or global average. The spatial size of the output tensor can be computed using the filter size F and the stride S given an input tensor W as:

$$Y = \lfloor \frac{W - F}{S} \rfloor + 1. \tag{2.9}$$

Since pooling layers cause an aggressive reduction in the representation size resulting in the loss of the exact position of the features, generative models such as variational autoencoders and generative adversarial networks have discarded pooling layers and opted instead for convolutional layers with larger strides.

Fully Connected Layer: The structure of neurons within a fully connected layer resembles that of the neurons in the feed forward neural networks presented in Section 2.4. The term fully connected stems from the fact that each neuron within the layer is connected to every activation from the previous layer. Similar to the neurons in a multi-layer perceptron, the activations of the fully connected layer can be computed as a matrix multiplication with a bias offset. They are traditionally employed towards the end of a CNN in order to aggregate global features from various input activations.

Recurrent Layer: In order to enable the network to process sequential input such as in video processing or language translation tasks, recurrent layers can be introduced within the network architecture. Unlike convolutional and fully connected layers, recurrent layers perform sequential processing of the input over time. At each time step, the output of the recurrent layer is a combination of its internal state and the current input, with the sequential information preserved in the hidden state of the network. A drawback, however, of this sequential processing over time is that information has to travel through multiple neurons before reaching the current processing neuron, which leads to the problems of vanishing and exploding gradients during the network training. In order to overcome this issue, Hochreiter and Schmidhuber proposed the Long Short-Term Memory (LSTM) units which help preserve the error through back-propagation [85]. Unlike regular recurrent units, LSTMs incorporate multiple gates which filter the incoming signal and decide how much of the internal state of the network should be incorporated in the current timestep. More precisely, an LSTM contains the following gates; input, output and forget gates, each of which having a direct impact on the learned representations by the layer. Unlike a standard recurrent layer, the addition of the aforementioned gates facilitates the training of LSTMs while reducing the problem of vanishing and exploding gradients.

2.5.2 Architecture Topologies

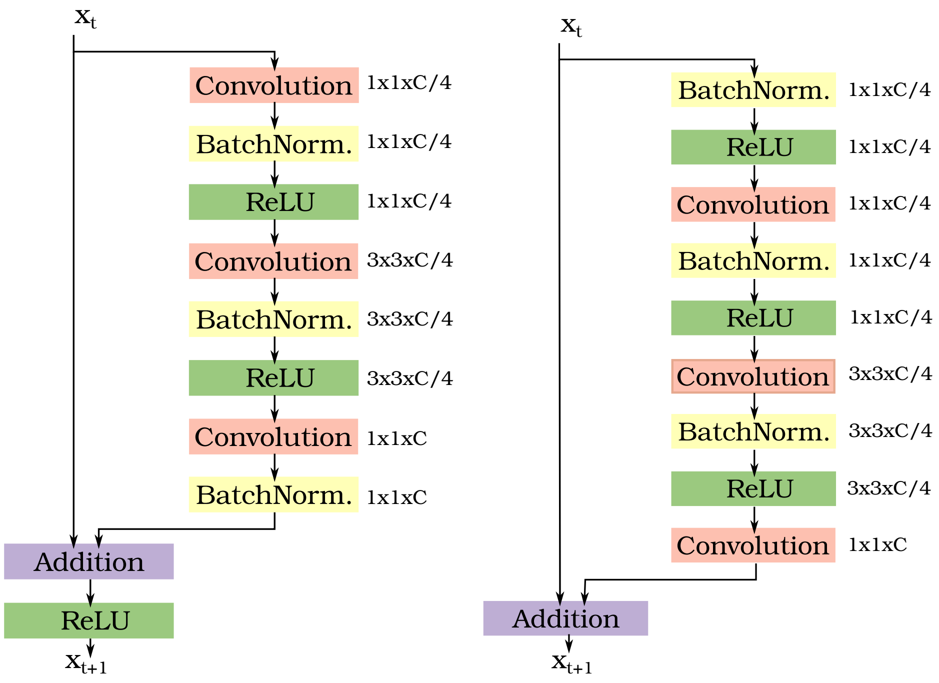
While multiple CNN architectures can be built by stacking the various layers described in the previous section, recent research has shown that increasing the depth of the network does not necessarily increase its representational capabilities [202]. On the other hand, increasing the depth renders the training of the network harder due to the vanishing gradient problem. In order to circumvent this issue, several state-of-the-art approaches have proposed alternative methods to stacking the blocks of the CNN by applying more complex connections [82, 87, 181]. In this section, we describe two architecture topologies which address the aforementioned issue that will be used later in this thesis as a base for our proposed networks.

Residual Networks: The Residual Network (ResNet) architecture won the first place in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [64] in 2015. As opposed to linearly stacking the layers of the CNN, ResNets propose a residual learning framework where the network architecture is comprised of several residual blocks. Each residual block consists of a series of layers linearly stacked and a parallel identity skip connection concatenating the input of the block to the output of the layers as shown in Figure 2.5(a). Thus given an input x_l, the output of the residual block can be represented as

$$x_{l+1} = F(x_l) + H(x_l),$$

where $H(x_l)$ represents the stacked non-linear layers within the residual block and $F(x_l)$ represents the shortcut connection. The addition of the skip connection thus facilitates the training of deeper architectures with faster convergence time as the gradient can directly flow through the skip connections as a shortcut. Figure 2.5(a) shows a bottleneck residual unit which consists of three convolutions in the order of 1×1, followed by 3×3 and 1×1 with alternating batch normalization and ReLU activations. The initial 1×1 convolution reduces the number of channels while the final 1×1 convolution restores the dimensions to match the input tensor. There exist two variants of the residual unit, an identity shortcut connection and a projection connection. Figure 2.5(a) illustrates an identity shortcut connection in which the input and output tensors have the same dimensions. In the projection shortcut connection on the other hand, a 1×1 convolution is applied to the input tensor in order to match its dimensions to that of the output. Note that in addition to the aforementioned bottleneck residual block structure, a two layer design was also proposed by the authors consisting of consecutive 3×3 convolutions.

Despite the addition of identity connections within a residual block in [81], the function $F(x_l)$ is never equivalent to the identity function due to the presence of the ReLU activation function after the addition of both connections. This in turn leads to loss of information about the original state of the signal. In order to remedy this, He *et al.* [82] proposed an alteration to the residual unit; namely the pre-activation residual unit shown in Figure 2.5(b). The contents of the residual unit remain the same, with a bit of reordering. Through moving the batch normalization layer to the beginning of the residual unit, the input of the layer is ensured to be renormalized after the addition operation from the previous layer, thereby improving the regularization of the network. Similarly, moving the ReLU activations to the beginning of the unit as opposed to after the addition operation ensures that the original information is preserved throughout the entire network. Utilizing the pre-activation residual units has shown an overall performance improvement as they reduce over-fitting while improving the network convergence. We employ both



(a) Original Residual Unit (b) Pre-activation Residual Unit

Figure 2.5: Schematic illustration of the original residual unit [81] and the pre-activation residual unit [82].

the original residual units and the pre-activation residual units as building blocks for our network architectures in Chapter 4 and 5.

Densely Connected Convolutional Networks: Similar to ResNets, Densely Connected Convolutional Networks or DenseNets [87] propose a connectivity pattern to alleviate the vanishing gradient problem occurring during training deeper architectures while ensuring maximum information and gradient flow throughout the network. In the DenseNet architecture shown in Figure 2.6, each layer is connected to every other layer in a feed forward fashion such that the input for each layer is the concatenated feature maps of all preceding layers, while its output is used as input for all subsequent layers. This has the advantage of reducing the number of parameters employed as it reduces the amount of redundant feature maps learned by encouraging feature reuse. DenseNets are comprised of alternating dense blocks and transition blocks. Within a dense block, the dimensions of the feature maps remain constant to enable their concatenation, however their volume changes. Transition blocks on the other hand, perform downsampling between the dense blocks through 1×1 convolution and 2×2 pooling layers. The network architecture has one hyperparameter; growth rate which controls the number of feature maps added by each layer, thus regulating the amount of information contributed by each layer to the

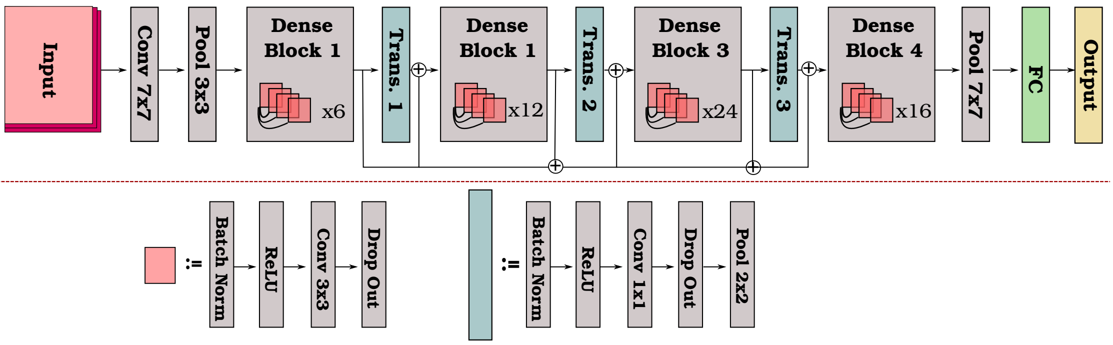


Figure 2.6: A schematic illustration of the DenseNet-121 architecture [87].

global state. DenseNets have achieved state-of-the-art performance on object recognition benchmarks [87], while consuming smaller number of parameters and less computation in comparison to other state-of-the-art architectures.

2.5.3 Cost Functions

Cost functions are an essential factor for training supervised neural networks that are used to measure the inconsistency between the predicted value and the ground-truth labels. In the context of this thesis, we primarily use two cost functions. In Chapter 4 and 5, we utilize the softmax function with cross-entropy loss in order to train our networks for the tasks of classification and pixel-wise semantic segmentation. Given N training examples, with ground-truth labels $y_i \in \{1, \cdots, C | i = 1, \cdots, N\}$ where C represents the number of classes and $f(x_i)$ denotes the activations of the network for input x_i. The softmax function with cross-entropy loss can be defined as:

$$\mathcal{L}_{CE} = \frac{-1}{N} \sum_{i=1}^{N} y_i \log \frac{exp\left(\mathbf{W}_{y_i} f(x_i) + \mathbf{b}_{y_i}\right)}{\sum_{j=1}^{C} exp\left(\mathbf{W}_j f(x_i) + \mathbf{b}_j\right)}, \tag{2.10}$$

where $\mathbf{W}$ and $\mathbf{b}$ are the weight and bias of the last fully connected layer in the network. The second cost function employed in Chapter 4 and 5 is the Euclidean loss for regression which minimizes the sum of squared differences as:

$$\mathcal{L}_{Euc} = \sum_{i=1}^{N} \left(y_i - f(x_i)\right)^2, \tag{2.11}$$

where N is the number of training examples, y_i is the ground-truth label and $f(x_i)$ is the predicted output of the network.

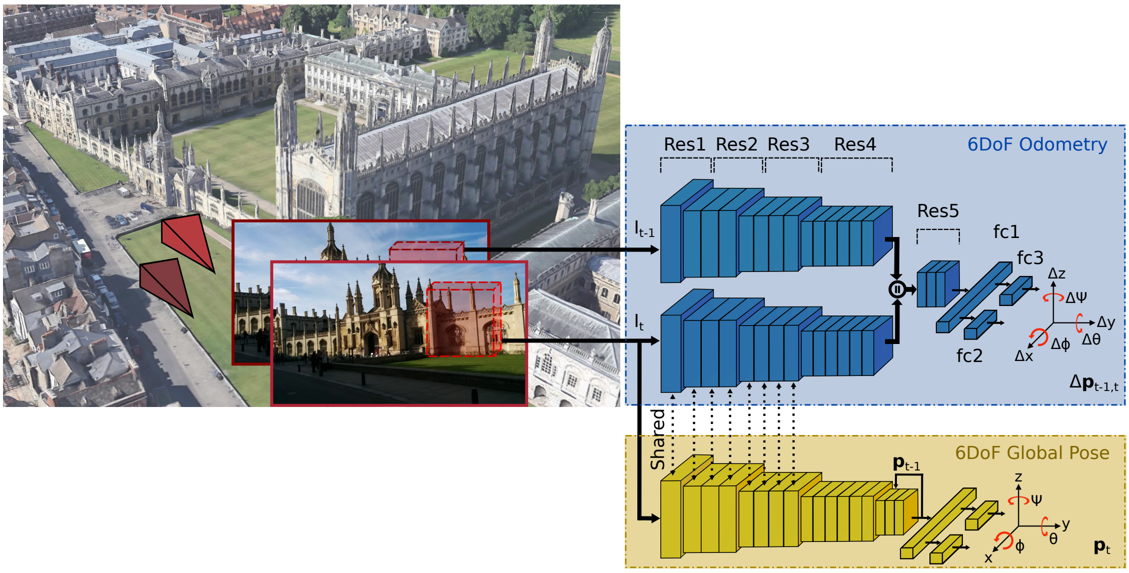


Figure 2.7: A depiction of the VLocNet architecture [193]. Given two consecutive monocular images, the network predicts the 6-DoF global pose and 6-DoF odometry simultaneously. Sharing the parameters between the global pose and odometry sub-networks enables the network to aggregate temporal features.

2.6 Auxiliary Learning Architecture for Pose Regression

In this section, we describe the VLocNet architecture [193] for regressing global poses and simultaneously learning to regress the relative motion between two camera frames using consecutive monocular RGB images. We build upon this architecture in Chapter 4 in order to learn a geometrically and semantically-aware pose regression model. The primary goal of the architecture is to precisely estimate the global pose while simultaneously learning to estimate the ego-motion of the robot. The features learned for relative motion estimation are leveraged by the global pose regression network to learn a more distinct representation of the scene. More specifically, the architecture consists of a three-stream neural network: a global pose regression stream and a Siamese-type double-stream for odometry estimation. An overview of the VLocNet architecture is shown in Figure 2.7.

Given a pair of consecutive monocular images (I_{t-1}, I_t), the network predicts both the global pose $\mathbf{p}_t$ and the relative pose $\Delta\mathbf{p}_{t-1,t}$ between the input frames. The input to the visual odometry streams are the images (I_{t-1}, I_t), while the input to the global pose stream is I_t. In the remainder of this section, we present the constituting parts of the VLocNet architecture along with how the joint optimization is carried out.

2.6.0.1 Visual Localization

Given an input image I_t and a previous predicted pose $\hat{\mathbf{p}}_{t-1}$, the network predicts the 6-DoF pose $\hat{\mathbf{p}}_t$. In order to estimate the global pose, the topology is built upon the ResNet-50 architecture [81] (Section 2.5.2) with the following modifications. The network architecture follows that of the ResNet-50 up to the last average pooling layer. Overall, the network consists of five residual blocks with multiple bottleneck residual units. Each residual unit consists of three convolutional layers with filter sizes: $1{\times}1$, $3{\times}3$, and $1{\times}1$. Similar to the standard residual units, each convolution layer is followed by a batch normalization with scaling and a non-linear activation. Unlike the standard ResNet-50 architecture, Exponential Linear Units (ELUs) are used for the non-linear activation, as opposed to the standard Rectified Linear Units (ReLUs), due to their ability to reduce the bias shift in neurons thus enabling the network to be more tolerant to noisy data [56]. Following the fifth residual block, a global average pooling layer is added in place of the average pooling layer, followed by three inner product layers: *fc1*, *fc2* and *fc3*. The first inner product layer has a dimension of $1{,}024$, and the following have dimensions 3 and 4, for regressing the translational $\mathbf{x}$ and rotational $\mathbf{q}$ components of the pose, respectively.

In order to enable the network to learn a pose estimate that is temporally consistent, the previous pose is fed through an inner product layer *fc4* of dimension D, then reshaped to fit the dimensions of the output of the last residual unit before the downsampling stage. Both tensors are then concatenated and fed to the subsequent residual unit. The single-task architecture is then referred to as VLocNet$_{\text{STL}}$.

2.6.0.2 Ego-Motion Estimation

The proposed architecture for relative pose estimation takes a pair of consecutive monocular images (I_{t-1}, I_t) as input and yields an estimate of ego-motion $\Delta\mathbf{p}_{t-1,t}$. The authors employ a two stream Siamese-type network based on the ResNet-50 architecture [81]. Each individual stream consists of four residual blocks, after which the feature maps are concatenated at the end of the *Res4f* block and convolved through the final residual block. Similar to the global pose regression stream, a global average pooling layer is added after the fifth residual block and subsequently three inner product layers, the first of which has dimension $1{,}024$, followed by 3 and 4.

As shown in Figure 2.7, the dual odometry streams have an architecture similar to the global pose regression network in order to facilitate feature sharing across both subnetworks. In the following section, we detail the sharing procedure between the global pose regression stream and the odometry stream taking the current image I_t, which has the goal of enabling the inductive transfer of information between both networks.

2.6.0.3 Deep Auxiliary Learning

The authors propose to jointly learn both global pose regression and visual odometry estimation due to the inherent similarities shared across both tasks in the feature space. Sharing features across multiple network streams can be considered as a form of regularization, in which both networks collaborate in updating the individual weights during back-propagation with the goal of minimizing the error term. Through this collaborative and competitive action, the network becomes less prone to over-fitting. In the proposed multitask learning architecture, the authors share weights between the global pose regression stream and the odometry stream taking image I_t from the current timestep as input, contrary to the majority of visual odometry estimation approaches in which the weights are shared across the two streams of the Siamese network. The proposed hard parameter sharing across multiple networks enables the ego-motion estimation network to effectively generalize to challenging situations such as motion blur or perceptual aliasing. Simultaneously, it enables the visual localization network to adjust its weights in a manner that places more attention towards areas of the image from which the relative motion can be easily estimated, thereby improving the predicted pose accuracy in challenging scenarios such as when estimating the pose in a textureless or structurally symmetric environment. Environmental aliasing in such scenes can substantially affect the accuracy of predicted poses in comparison to environments with abundant structural variations. However, using the proposed parameter sharing to jointly train both networks, the global pose stream can leverage the relative motion features from the visual odometry network to produce more accurate localization estimates. The multitask architecture is referred to as VLocNet$_{\text{MTL}}$. In Chapter 4, we employ the VLocNet architecture as a base for our multitask learning framework and investigate the effect of utilizing a novel loss function accompanied with a self-supervised context-aggregation mechanism on the accuracy of the learned pose predictions.

Chapter 3

Robust Visual Localization using Textual Information

Accurate robot localization plays a crucial role in the success of the overall mobile robotic system. While robotic platforms operating in urban environments most commonly utilize the GPS signal as a reliable source of localization information, the signal quality is often poor due to the presence of high rises causing GPS outages. Textual information in the form of street and shop signs is highly abundant in urban environments and constantly used by humans for a majority of their daily tasks, ranging from finding locations to specifying their position. Consequently this information is constantly updated and highly accurate rendering it suitable as a source of stable features. Nonetheless, it has yet to be exploited in the robotics field. In this chapter, we present a localization method that leverages the textual information in the scene to estimate the 2D pose in the environment. We utilize an off-the-shelf text spotting technique to extract text labels from the surrounding scene and employ a custom data association approach employing lexical and string similarity methods to identify landmarks using geo-referenced texts in public maps. Finally, we obtain the localization estimate by applying a probabilistic localization method with specific sensor models to integrate multiple observations. We evaluate our approach extensively on real-world data gathered from three different cities and demonstrate substantial improvement using our proposed method over GPS.

3.1 Introduction

Localization is one of the fundamental problems in the area of mobile robotics. The accurate knowledge of the robot position enables a variety of tasks including navigation, transportation, as well as search and rescue. Additionally, the exact information about the position of a user gives the opportunity to offer so-called location-based services with plenty of uses in social networking, health, guidance, entertainment and many others. While several approaches exist for addressing the robot localization and mapping problem in various environments using vision [72, 136, 178], the most popular and commercially used approach for solving the localization problem in outdoor scenarios is GPS. Although GPS can theoretically reach an accuracy of a few meters, it cannot always be achieved in practice due to GPS outages, e.g. inside or near buildings.

In the classical approach, localization is performed after a previous visit of the environment during which a map has been built. Despite the accuracy produced from localizing using self-built maps, employing this approach would require several subsequent visits of the environment in order to continuously update the map reflecting any changes in the environment. Recently, the continuous rise of large scale, and publicly available mapping services, such as Google Maps and OpenStreetMap, has lead to an increased interest in approaches utilizing the information provided by those maps for both robot localization and navigation tasks [23, 69, 125, 208], with the majority focus on geo-tagged street-level imagery. Unlike self-built maps, publicly available maps are constantly updated by the service provider thus enabling life-long localization without the need of continuous map updates on the client side. Furthermore, utilizing publicly available maps enables localization in previously unseen environments as it alleviates the need of the robot to perform the initial mapping step.

The use of vision-based approaches for localization has resulted in a variety of choices for the types of features that can be used. Features directly computed from images of the scene are most commonly selected, with growing effort in making the approach as robust as possible to the challenges faced localizing in outdoor environments. Such challenges include, but are not limited to, changes in lighting conditions e.g. day-time versus night-time, varying weather conditions and seasonal changes e.g. sunny summer day versus snowy winter afternoon, or structural scene changes e.g. as a result of construction sites in urban environments. The most common approaches to deal with lighting artifacts involve either developing special feature detectors to enable robustness [151, 167], or collecting large amounts of data in different conditions, and attempting to localize at run-time using the collected data [61, 80]. To handle structural variances in the environment, new data must be collected to reflect the changes in the map structure. In this work, we present an approach that is robust to environmental and structural changes. Moreover, as our approach uses pre-existent information in maps that is stable features across various environments and seasonal changes, it provides life-long localization capabilities and can

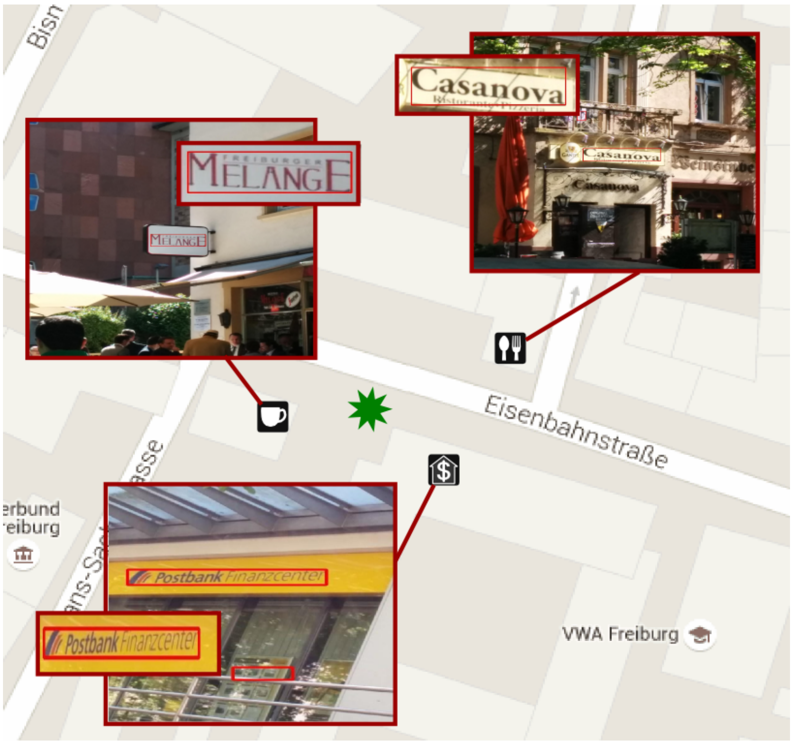


Figure 3.1: Overview of our text-based localization approach: Standing at a certain position, we capture images of the neighboring shops. Using text spotting, we extract the text information to match the observation with geo-referenced texts from the map and generate a pose estimate.

be used in different environments without necessary adaptations.

Observing human behavior allows us to gain a new perspective on which features to select for localization. When identifying their location or describing a path to follow, people tend to use names to guide the explanation, e.g, a name of a shop or a restaurant. Our approach moves away from vision-based feature matching to use mid-level representations for estimating the current geo-location of an image. We propose a method that exploits the abundance of textual information in the environment in order to localize either a robot or a user holding a mobile device. Our proposed approach is easy to deploy, intuitive to use and takes advantage of the available resources. Furthermore, using textual information gives us the ability to communicate information easily with the user via speech-based systems.

We propose a solution to the localization problem using a standard RGB camera, and publicly available map information *without* the use of any street-level imagery by exploiting the rich textual meta-data content of maps, such as the street-level annotations of shops and businesses, moving away from vision-based feature matching. Specifically, we concentrate on extracting text "in the wild" from images that are cross-referenced from the available annotated map. Accordingly, we present a new localization form with global-scale breadth, low bandwidth requirements (no images are transferred over the network), and life-long capabilities (publicly available maps are continually updated).

Figure 3.1 shows an overview of our approach which is split into three main stages. First, we extract text from the captured scene images. The extracted texts are then used to identify landmarks in the vicinity of the camera. Finally, we employ a particle filter [70] with a dedicated sensor model to obtain accurate location estimates. We propose two variants of our method, a Single-Shot Localization method and a Pose Tracking Localization method. Furthermore, as the accuracy of our localization method is dependent on the quality of the extracted text from the scene, we propose multiple landmark selection strategies. We perform extensive experiments evaluating the localization accuracy of our method in three cities in Freiburg, Zurich and London and demonstrate an improvement of 40% in translation using our proposed method over GPS-based localization.

In summary, the primary contributions of this chapter are as follows:

- A novel single shot global localization method using publicly available maps and textual features from the scene.
- A pose tracking localization approach with adapted sensor models to integrate the multiple observations.
- A probabilistic landmark selection strategy that utilizes both the lexical relations and word similarity for data association.
- Finally, we perform extensive experimental evaluation on data from three cities, comparing the localization accuracy of both methods as well as investigating the effect of the landmark selection strategy on the pose error.

3.2 Text-Based Localization

In the following, we formalize the problem of localization in urban environments using textual features in the scene. Our approach relies on RGB camera images and an IMU sensor. Given a map of the environment and at least two images containing text, our method extracts textual features from the images and employs a probabilistic data association strategy to match them to landmarks in the environment. In order to obtain a robust estimate of the pose, we adapt Monte Carlo methods accounting for the employed text extraction approach. We propose two variants for our pose estimation approach, a Single-Shot global Localization (SSL) method and a Pose Tracking Localization (PTL) variant. In the remainder of this section, we first describe the map and state representation, followed by the proposed pose estimation methods. Finally, we outline the text spotting and data association strategies that we utilize.

3.2.1 Map and State Representation

We represent the environment by a set of landmarks, each of which corresponds to a text that could belong to a shop, restaurant, street name, etc. The only assumption that we make is that the text is static, i.e., it is not scrolling over a display. Text signs which are not present in the map, e.g. “Stop”, are not considered a part of our environment model, and hence are not counted as landmarks for pose estimation. We assume that for each landmark l in the map, we have the following set of features:

- the name, which is the text that appears on the sign,
- the geo-location coordinates (l_x, l_y) of the sign,
- the orientation angle (l_o) of the sign (where 0 degrees is north), and
- the size (l_s), from which we compute the maximum distance of observing the sign.

In principle, any publicly available map can be used for the described representation, as the extra features required can be easily inferred from the map structure itself. Landmark orientation can be computed from the street orientation, as text is placed either parallel or orthogonal to the road. The map information provides knowledge about the orientation of the streets with respect to north, which can be directly generalized to all landmarks within that street. A consequence of localizing in an urban environment is that it is unlikely to be able to observe a sign of a shop that is two streets or more away from our location due to occlusions. Accordingly, we estimate the size of the landmark by the width of the nearest street. Note that the size of the landmark is only used by our SSL method. In the PTL variant, the size of the sign is not required due to the availability of motion information. In addition to the RGB camera, we rely on IMU information to obtain the angle with which a landmark is observed.

3.2.2 Pose Estimation

We propose two techniques for estimating the 2D pose of the robot: a SSL method and a PTL method. Both of our proposed approaches build on top of the particle filter method for robot localization [70] with a number of modifications. An overview of the particle filter approach is presented in Section 2.3. In the SSL variant of our method, the goal is to estimate the location of the robot in a single timestep using only information available at this timestep. For the PTL method, on the other hand, utilizing the odometry information from the IMU, we estimate the most likely position at each timestep by sharing the information among the various poses. In the following, we elaborate on each of the variants. The text spotting and data association approaches are discussed fully in Section 3.2.3. For the time being, we assume that given an observation image z_i, the text spotting method in Section 3.2.3 extracts a set of words D^i. The extracted words are then used by the data association approach to output a set of potential matching landmarks L^i. More precisely, the data association function $A_{z_i} : D^i \mapsto L^i$ for observation z_i takes as input the set of extracted words D^i and returns a potentially empty set of matching landmarks L^i from the map.

3.2.2.1 Single Shot Localization (SSL)

Given a set of observations $Z := \{z_1, \dots z_n \mid n \in \mathbb{N}\}$ and the map m, our goal is to find the position of the robot $x \in \mathbb{R}^2$ by estimating $p(x \mid z_{1:n}, m)$. First, we assume that the individual measurements are independent given the position and the map, which by applying Bayes' theorem leads to

$$p(x \mid z_{1:n}, m) = \frac{p(z_{1:n} \mid x, m)}{p(z_{1:n} \mid m)} p(x \mid m). \tag{3.1}$$

Since we assume the individual measurements are independent given the position of the robot and the map, the above equation simplifies to

$$p(x \mid z_{1:n}, m) = \frac{\prod_{i=1}^{n} p(z_i \mid x, m)}{p(z_{1:n} \mid m)} p(x \mid m). \tag{3.2}$$

Given that the position of the robot is independent of the map and the individual measurements are independent given the position and the map, we apply Bayes' theorem to get

$$\begin{aligned} p(x \mid z_{1:n}, m) &= \frac{\prod_{i=1}^{n} p(x \mid z_i, m) p(z_i \mid m)}{p(z_{1:n} \mid m)} p(x \mid m) \\ &= \eta \prod_{i=1}^{n} p(x \mid z_i, m), \end{aligned} \tag{3.3}$$

for some constant η. To calculate $p(x \mid z_i, m)$ for observation z_i, we integrate over all possible landmark associations L^i that are obtained from the data association function $A_{z_i} : D^i \mapsto L^i$ described in Section 3.2.3:

$$\begin{aligned} p(x \mid z_i, m) &= \sum_{l \in L^i} p(x, l \mid z_i, m) \\ &= \sum_{l \in L^i} p(x \mid l, z_i, m) \cdot p(l \mid z_i, m). \end{aligned} \tag{3.4}$$

Since the belief computed by Equation (3.4) is multimodal with the number of modes growing combinatorially with the possible data associations, we approximate it with a weighted sample set. To sample from it we resort to the importance sampling principle and choose as proposal distribution

$$\begin{aligned}\pi(x) &= p(x \mid z_q, m) \\ &= \frac{p(z_q \mid x, m) \cdot p(x \mid m)}{p(z_q \mid m)},\end{aligned} \tag{3.5}$$

where we chose the measurement z_q uniformly at random from Z. According to the importance sampling principle, we compute for each sample its importance weight

$$\begin{aligned} w(x) &= \frac{p(x \mid z_{1:n}, m)}{p(x \mid z_q, m)} \\ &= p(z_{1:n} \mid m)^{-1} p(z_q \mid m) \prod_{i \neq q} p(z_i \mid x, m) \\ &\propto \prod_{i \neq q} p(z_i \mid x, m). \end{aligned} \tag{3.6}$$

We model the individual likelihood for each landmark $z_i \in Z$ with a mixture distribution over the latent data association variables. We assume the set of associations returned from the text spotting phase to be equally likely. This results in:

$$\begin{aligned} p(z_i|x, m) &= \sum_{l \in L^i} \frac{1}{|L^i|} p(z_i|l, x, m) \\ &= \frac{1}{|L^i|} \sum_{l \in L^i} \mathcal{U}(\delta_i; 0, l_s) \mathcal{N}(\beta_i; l_o, \sigma^2), \end{aligned} \tag{3.7}$$

where $|L^i|$ denotes the size of the landmark set returned by the data association function A_{z_i}. We denote the distance measurement by δ_i which can be computed as per Equation (3.8) and utilize the landmark size information l_s from the map for the expected value. Similarly for the angular measurement, we utilize the orientation information from the IMU for obtaining l_o for the expected value and compute the measurement β_i as shown in Equation (3.9).

$$\delta_i = \| \begin{pmatrix} l_x \\ l_y \end{pmatrix} - x \| \tag{3.8}$$

$$\beta_i = \mathsf{atan2}\left(l_y - x_y, l_x - x_x\right), \tag{3.9}$$

where (x_x, x_y) are the 2-D coordinates defining the robot position x. We employ a Gaussian distribution to model the angular measurement and model the distance measurement using a Uniform distribution such that:

$$\mathcal{U}(\delta_i; 0, l_s) \;=\; \begin{cases} \frac{1}{l_s}, & \text{if } 0 \leq \delta_i \leq l_s \\ 0, & \text{else.} \end{cases} \tag{3.10}$$

In order to allow our method to be tolerant to outliers and false measurements from the data association phase, we apply a robust method for computing the particle weights, inspired by the trimmed estimator approach [164]. A trimmed estimator excludes extreme values while computing the desired statistics. Extreme values can be either the lowest/highest n percentile or the n-th maximum/minimum points. We discard the lowest 20 percentile of likelihood values to compute the weights.

3.2.2.2 Pose Tracking Localization (PTL)

While the approach presented in the previous section provides a localization estimate within a single timestep, the accuracy of the results is highly dependent on the presence of at least two text-containing signs in the scene. In order to overcome this limitation, we propose a second variant of our method where we share information across timesteps and utilize the odometry information from the IMU in order to gain probabilistic localization estimates for each timestep.

We utilize a similar formulation as in the previous section, given a map of the environment m and observations $Z_1, \dots, Z_t$, our goal is to estimate the probability $p(x_{1:t} \mid Z_{1:t}, m)$ of being at locations $x_{1:t}$ at timesteps $i \in 1, \dots, t$. For the first timestep, we employ the SSL approach (Section 3.2.2.1) to obtain an initial estimate of the position and spread all particles at this position with equal weights. At each subsequent timestep, we utilize the odometry information u_t in order to update the position of particle $j \in \{1, \dots, J\}$; where J is the number of particles as follows:

$$x_t \sim p(x_t \mid x_{t-1}, u_t), \tag{3.11}$$

We represent the target distribution as $p(x_{1:t} \mid Z_{1:t}, u_{1:t})$ and sample from the proposal $p(x_{1:t} \mid Z_{1:t-1}, u_{1:t})$. Following the Markov assumption, we reformulate the proposal as

$$p(x_{1:t} \mid Z_{1:t-1}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t) \cdot p(x_{1:t-1} \mid Z_{1:t-1}, u_{1:t-1}). \tag{3.12}$$

Using this formulation, the importance weight for each particle is then computed as

$$\begin{aligned} w\left(x^j_{1:t}\right) &= \frac{p(x^j_{1:t} \mid Z_{1:t}, u_{1:t})}{p(x^j_{1:t} \mid Z_{1:t-1}, u_{1:t})} \\ &= \frac{p(x^j_{1:t} \mid Z_{1:t}, u_{1:t})}{p(x^j_t \mid x_{t-1}, u_t) \cdot p(x^j_{1:t-1} \mid Z_{1:t-1}, u_{1:t-1})} \\ &\propto p(Z_t \mid Z_{1:t-1}, x^j_{1:t}, u_{1:t}). \end{aligned} \tag{3.13}$$

Similar to the SSL method, we assume all data associations to be equally likely and employ a mixture of Gaussians distribution over the latent variables in order to model the individual likelihood resulting in:

$$
\begin{aligned}
p(z_i^t \mid x_{1:t}^j, Z_{1:t-1}) &= \sum_{l \in L^i} \frac{1}{|L^i|} p(z_i^t \mid l, x_{1:t}^j, Z_{1:t-1}) \\
&= \frac{1}{|L^i|} \sum_{l \in L^i} \mathcal{N}(\beta_i; l_o, \sigma^2), \qquad (3.14)
\end{aligned}
$$

where l_o represent the expected angular measurement and β_i the actual measurement computed as in Equation (3.9). Note that unlike the SSL approach, this method does not utilize the distance measurement in the likelihood computation. However, similar to the SSL variant of our method, we use a trimmed estimator to eliminate false measurements resulting from the data association phase.

3.2.3 Text Spotting and Data Association

In this section, we elaborate the text spotting and data association methods employed as well as any post-processing steps. While our approach is independent of the text spotting method employed, we provide some heuristics which aim at reducing the noise in the extracted text and enable correct data association regardless of any discrepancies occurring between the map information and the scene.

3.2.3.1 Text Spotting

Text spotting refers to the combined action of text detection and text recognition from an input image. In this thesis, we do not implement a text spotting method, as we consider it outside the scope of our investigation. Instead, we treat the text spotting procedure as a black-box method with an input image and an output consisting of a set of words each with a confidence score. In order to recognize texts in natural scene images, we employ the method from Neumann and Matas [141], which falls into the category of approaches that use region groupings. In their work, the authors train a sequential classifier for character detections to select extremal regions from the component tree of the image. They further use a number of heuristic functions to effectively prune the selected regions. This allows for a fast exhaustive search of the state space of character sequences before grouping the regions into high level text blocks. We adopted this method in our work due to its robustness, and relied on an open-source implementation by the authors. In principle, any combination of text detection and recognition approaches can be used without affecting our pipeline.

The text extraction method provides a list of the different detected words, each with an associated confidence score. We perform two-phase post-processing spell correction on the extracted text. As a first step to reduce noise in the returned text, we threshold the returned words based on the confidence score, thus reducing noise due to text detection errors. Furthermore, we discard words with multiple repeated characters and single letter words, e.g., “gaummmm”. The goal of the second stage is to fix any substitution errors (e.g., the letter “l” and the number “1”). For this purpose, we create a custom dictionary containing only words occurring in the map and use the GNU Aspell* spell checker to find the closest matching dictionary word. In order to avoid incorrect matches, the spell correction is only carried out if the minimum number of edits needed to convert the extracted word to a dictionary word is less than half of the length of the extracted word. As an example, given “volksl” as a detected word with the ground-truth text “volksbank”, and the closest matching dictionary word “oska”. Since the edit distance score (4) is greater than half the length of the word (3), the word does not get corrected and remains as is for the data association phase. By employing this constraint, our spell correction method can be regarded as lazy, wherein a correction is only performed if the number of required changes does not exceed a certain word length-specific threshold. This in turn adds flexibility at the landmark association phase by avoiding incorrect matches due to overconfidence.

3.2.3.2 Data Association

After the post-processing stage, we use the extracted text to assign a set of landmarks for each image by measuring the similarity score between the corrected text and each landmark in the map. We use a function based on the similarity score to choose the top N landmarks and select the landmarks with probability higher than that of random guessing. We introduce three different measures for calculating the similarity score, namely: i) Levenshtein distance, ii) WordNet Lin similarity, and iii) Combined weighted similarity.

The Levenshtein distance [116] measures the minimum number of operations to transform one string of text to another. The allowed operations are insertion, deletion, substitution and transposition. Mathematically, the Levenshtein distance is defined as follows

$$lev_{a,b}(i,j) = \begin{cases} max(i,j), & \text{if } min(i,j) = 0 \\ min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i-1,j-1), \\ lev_{a,b}(i,j-1)+1 \end{cases} & \text{otherwise} \end{cases} \tag{3.15}$$

where a and b are the two strings we are trying to compute the similarity of, and i, j are indices for a and b respectively, with an initial value equal to the length of each string.

WordNet [134] is a collection of English nouns, verbs, adjectives and adverbs arranged into a lexical database. Words are grouped together into sets, dubbed synsets, expressing

[*] K. Atkinson. GNU Aspell, 2003. http://aspell.net

a unique concept. Synsets are connected with each other using semantic and lexical relations. There exists a number of different similarity measures quantifying the semantic and lexical relatedness of two words based on the WordNet database [46]. In this chapter, we employ the Lin similarity measure [118] which calculates the similarity between two words based on both thesaurus and probabilistic information. It is formally defined as follows:

$$sim_{wn}(c_1, c_2) \;=\; \frac{2 \cdot IC(lso(c_1, c_2))}{IC(c_1) + IC(c_2)}, \tag{3.16}$$

where IC is the Information Content of the word, and lso is the Least Common Subsumer (most specific ancestor node) of the two words. The Lin similarity thus takes into account both the information shared between the two words and the difference, hence providing a more accurate ranking of the similarity than the path-based similarity measures in WordNet [112, 201].

In order to measure the similarity between the corrected text and the map landmarks, we devise an approach that takes advantage of the uniqueness of named entities. A named entity is a real world object that can be denoted by a proper name; for instance the name of a person, the name of an organization, the name of a country, etc. Our approach is divided into several stages, first the corrected text is split into a set of distinct words S_t. This splitting is also performed for the text of each landmark l^i forming S_{l^i}. We remove the named entities from each set and place them into two separate sets: $S_{t_{NE}}$ and $S_{l^i_{NE}}$. For each named entity in $S_{t_{NE}}$, we check if it exists in $S_{l^i_{NE}}$, if it does, then we compute the Lin similarity between the words in S_t and S_{l^i} and assign that as the score to the landmark. Otherwise, the landmark is assigned a score of 0. Since named entities are unique identifiers for places, we use this idea to enable matching text to a landmark if the name on the map differs from the street sign, e.g. “Café Lichtblick” and “Restaurant Lichtblick”. At the same time, we want to avoid situations were we match two places with different names, e.g. “Restaurant Lichtblick” and “Restaurant Wolfshole”.

Our combined weighted similarity uses the scores from both the Levenshtein distance and the WordNet similarity methods described above. The final score is computed as a weighted sum of both the similarity measures. In order to be able to combine both scores, we first normalize their values by setting the scale to be between 0 and 1. However, as a low Levenshtein distance means a high similarity score, we invert the computed distance in order to compute the Levenshtein similarity measure as follows:

$$sim_{lev}(t_1, t_2) = 1.0 - lev(t_1, t_2). \tag{3.17}$$

Accordingly, we define the overall similarity score as:

$$sim(t_1, t_2) = \alpha \cdot sim_{lev}(t_1, t_2) + (1 - \alpha) \cdot sim_{wn}(t_1, t_2), \tag{3.18}$$

where α acts as a hyperparameter controlling the value of the final score. A higher value of α places more importance on the Levenshtein score over the WordNet score and vice versa.

Finally, as an extra step to reduce false positive matches, we only consider landmarks as potential matches if they have non-zero scores from both the Levenshtein distance and the WordNet similarity measures.

Each of the aforementioned similarity measures assigns a score to the landmarks within the map. The landmarks are then ordered descendingly and the top N with a matching score higher than that of random guessing are assigned to the observation. In the event that the extracted text does not match any of the landmarks in the map, then this observation is discarded. In Section 3.4.2, we evaluate the performance of the proposed data association strategies and their effect on the localization accuracy.

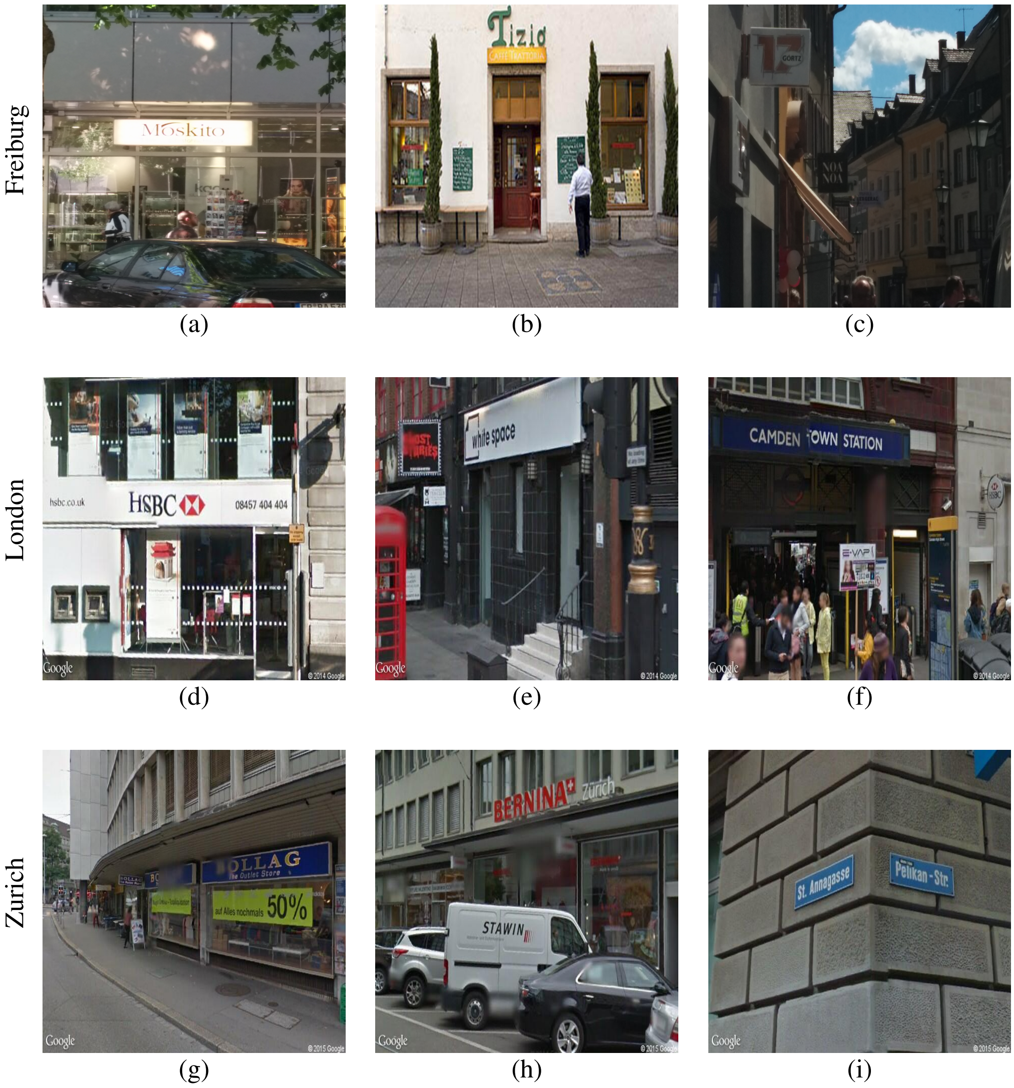
3.3 Data Collection and Labeling

We collected data from three different cities: Freiburg, London and Zurich and use OpenStreetMap as the source of information for creating our custom map. For each city, we collected data from different regions covering suburbs, industrial and commercial zones. We utilized Google Street View to collect the London and Zurich datasets, and obtained the odometry data manually using the ground-truth pose information. In order to mimic drift accumulation in real world data, we augmented the odometry with random noise sampled from a Gaussian distribution.

Due to the unavailability of Google Street View in Freiburg however, we used a Google Tango tablet for manually collecting images and odometry. Furthermore, we gathered the GPS coordinates for each pose in order to compare the performance of our localization method with that of GPS from a mobile device. For both the London and Zurich datasets on the other hand, we were unable to obtain the raw GPS measurements from the Google Street View interface and as such do not perform GPS comparisons.

In each city, the data is grouped into a number of paths. Each path consists of a sequence of poses with a maximum distance of 100m between two consecutive poses. For each pose, the observations were captured by standing in a particular location and rotating in place. The Freiburg dataset contains a total of 60 poses and the corresponding map contains 180 landmarks. We manually added a few annotations to the Freiburg map in order to capture some unidentified shops and alterations due to construction sites which were not captured in OpenStreetMaps. Figure 3.2(a-c) shows example images from the Freiburg dataset. The captured images display a wide range of lighting conditions which increase the difficulty of the text spotting and data association tasks.

The London dataset is comprised of 300 poses with an associated map containing approximately $1{,}000$ landmarks from different shops, restaurants and signs. Figure 3.2(d-f) displays example images from the London dataset. Image stitching artifacts and parallax errors further increase the difficulty of the text spotting and data association tasks on this dataset. Similar to the London dataset, the Zurich dataset is comprised of 300 poses



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3.2: Example images from the data collected from Freiburg, London and Zurich. The images show shop fronts, subway stations and street signs. The images were captured in different areas of the cities and each contains at least one word from a shop front or street sign.

with an associated map containing 900 landmarks. The poses were captured from central, industrial and rural regions. In Figure 3.2(g-i), we show sample images from the dataset. Benchmarking the localization performance on this dataset is extremely challenging due to the sparsity of text containing images which lead to an average of two observations per pose, versus three observations per pose for the remaining datasets. Furthermore, the dataset images cover a larger variety of text fonts in multiple different sizes and languages which increase the difficulty of the text spotting task.

3.4 Experimental Evaluation

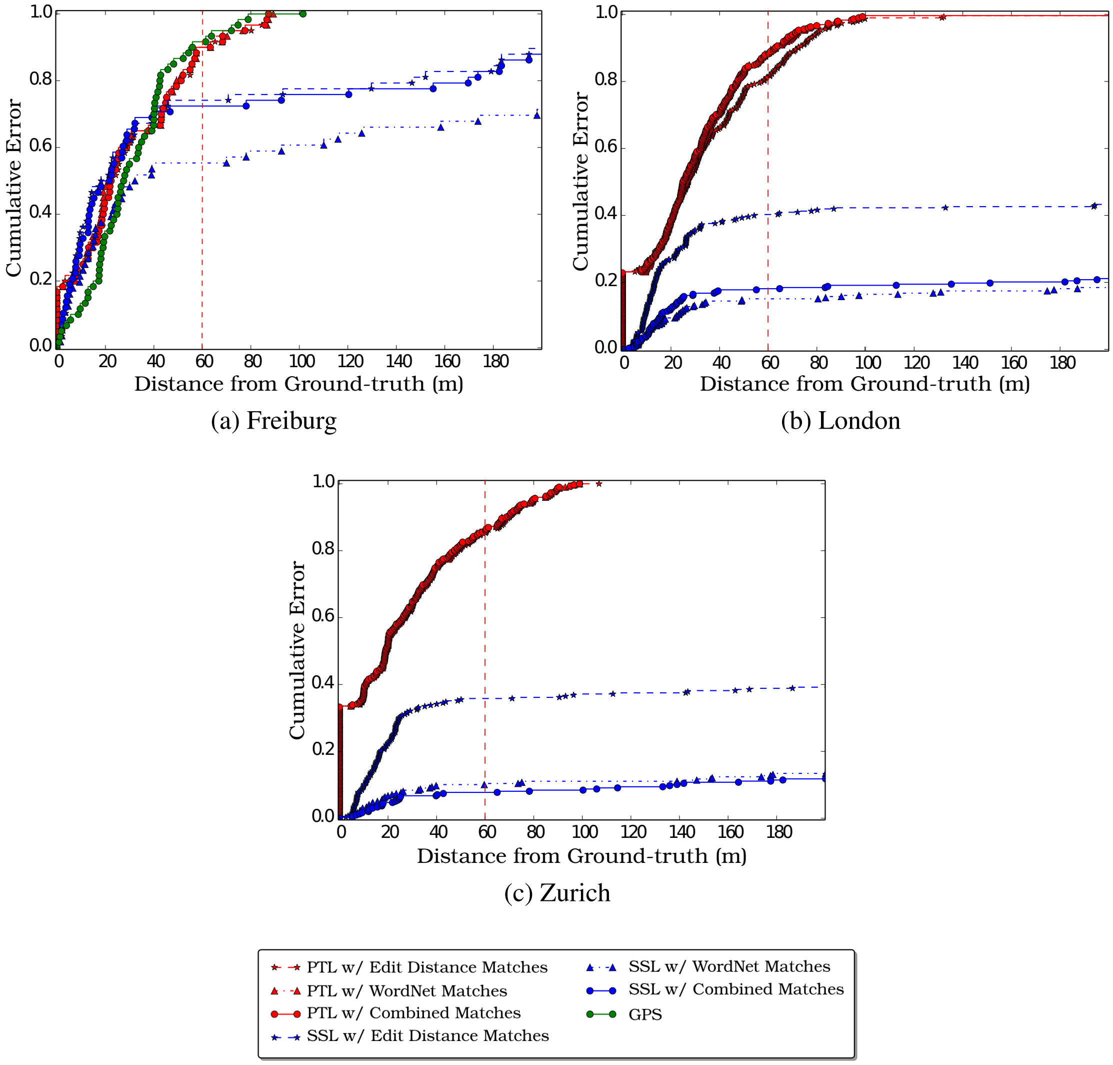
In this section, we perform experimental evaluations benchmarking the performance of our proposed text-based localization approaches on the aforementioned dataset. We compare the performance of both our SSL and PTL methods with GPS in Section 3.4.1. In order to investigate the effect of the data association strategy on the localization performance, in Section 3.4.2, we evaluate the performance of each localization approach by employing the various data association strategies. In Section 3.4.3, we present a qualitative analysis of our proposed approach along with an ablation study on the effect of the text spotting method on the accuracy of the localization estimates.

In order to quantify the performance of the SSL method, at each timestep we estimate the current position of the robot independent of the previous locations. Initially the particles are distributed around one of the observed landmarks, selected at random. The weight assigned to each particle is then computed by calculating the likelihood of observing the remaining landmarks. The reported position is then the weighted average over all the particles.

In the remainder of this chapter, we define localization failures as cases where the text spotting method fails to extract any text for 50% or more of the captured images/observations. In such cases, our algorithm outputs a message signaling the lack of sufficient data to accurately estimate a location. For the WordNet Lin similarity data association method, we use the English version of WordNet. Moreover, for the Combined weighted similarity, we use cross-validation on each dataset to determine the optimal value to employ which provides the best balance between the comprising similarity scores.

3.4.1 Evaluation of the Localization Performance

We evaluate the performance of the proposed localization methods on each of the collected datasets. Figure 3.3 shows the cumulative error plot on the Freiburg, London and Zurich datasets. For each dataset, we compare the performance of both our Single Shot Localization (SSL) and Pose Tracking Localization (PTL). For both of the proposed localization methods, we also plot the performance employing the data association strategies proposed



(a) Freiburg

(b) London

(c) Zurich

Figure 3.3: Cumulative error plots on all paths in the Freiburg, London and Zurich datasets.The x-axis shows the distance from ground-truth position in meters, and the y-axis shows the percentage of points with distance less than or equal to the x-value. For both our Pose Tracking Localization (PTL) and Single Shot Localization (SSL) approaches, localization failures occur with error values of 60.0m or higher as highlighted by the red dashed line in each plot.

in Section 3.2.3.2. Furthermore, on the Freiburg dataset, we additionally compare the performance of our proposed methods with GPS. However, for both the London and Zurich datasets, we were unable to extract the raw GPS measurements from Google Street View maps and thus do not provide GPS comparison on either dataset. On all three datasets, we highlight the localization failures with a red dashed line in Figure 3.3.

In order to evaluate the performance of the localization approaches on the Freiburg dataset, we split the data into 9 distinct paths. On this dataset, the value of α for which we achieve the lowest localization error is 0.8. Using this α value, more weight is put on the results from the Levenshtein similarity approach than the WordNet similarity measure. This is a consequence of using the English language WordNet for evaluating our method, as a large number of the detected words are undefined within the English version for the Freiburg dataset. Figure 3.3(a) shows the cumulative error plot on the Freiburg dataset. Both the SSL and PTL methods achieve a mean localization error of 10.9m versus 27.0m achieved by GPS excluding the localization failures. The results further show that both our STL and PTL methods achieve a error between 0.0 and 40.0m for over 69.0% of the localization poses, whereas the GPS achieves this error value for only 60.0% of the given poses. Comparing the performance of the SSL method with the PTL method, we observe that the former is more susceptible to localization failures (3.3%) due to the lack of sufficient information. While employing the PTL method, no localization failures are encountered as the odometry information substitutes lost/missing information due to text spotting failures. We discuss the effect of the various data association strategies on the localization performance in Section 3.4.2.

We split the London data into a total of 67 paths and set the value of α to 0.5. Unlike the values selected for Freiburg and Zurich datasets, we did not perform any cross-validation when selecting this value. Since this is the only dataset with text entirely in English, we instead chose to select a value giving equal weights to each approach in order to evaluate the performance of using the Combined weighted similarity measure. Figure 3.3(b) shows the cumulative localization error plot of both the SSL and PTL methods, which shows a stark difference in comparison to the Freiburg dataset. For 70.0% of the poses, the PTL method achieves a localization accuracy between 0.0 and 40.0m. On the other hand, the SSL approach is able to guarantee an error between 0.0 and 40.0m for only 39.5% of the poses. Moreover, the results show that the PTL method achieves an accuracy near 1.0m for more than 20.0% of the poses without any localization failures. This further validates the benefit of sharing information across multiple timestamps on the accuracy of the predicted localization poses.

We divided the 300 poses of the Zurich dataset into 98 distinct paths. For the Combined weighted similarity measure, we set α to 0.9 which was obtained through cross-validation. Using this value, the Combined weighted similarity measure assigns more weight to the score assigned by the Levenshtein distance measure than WordNet. We hypothesize this occurs to account for the discrepancy between the language of the street signs and the dictionary of WordNet.

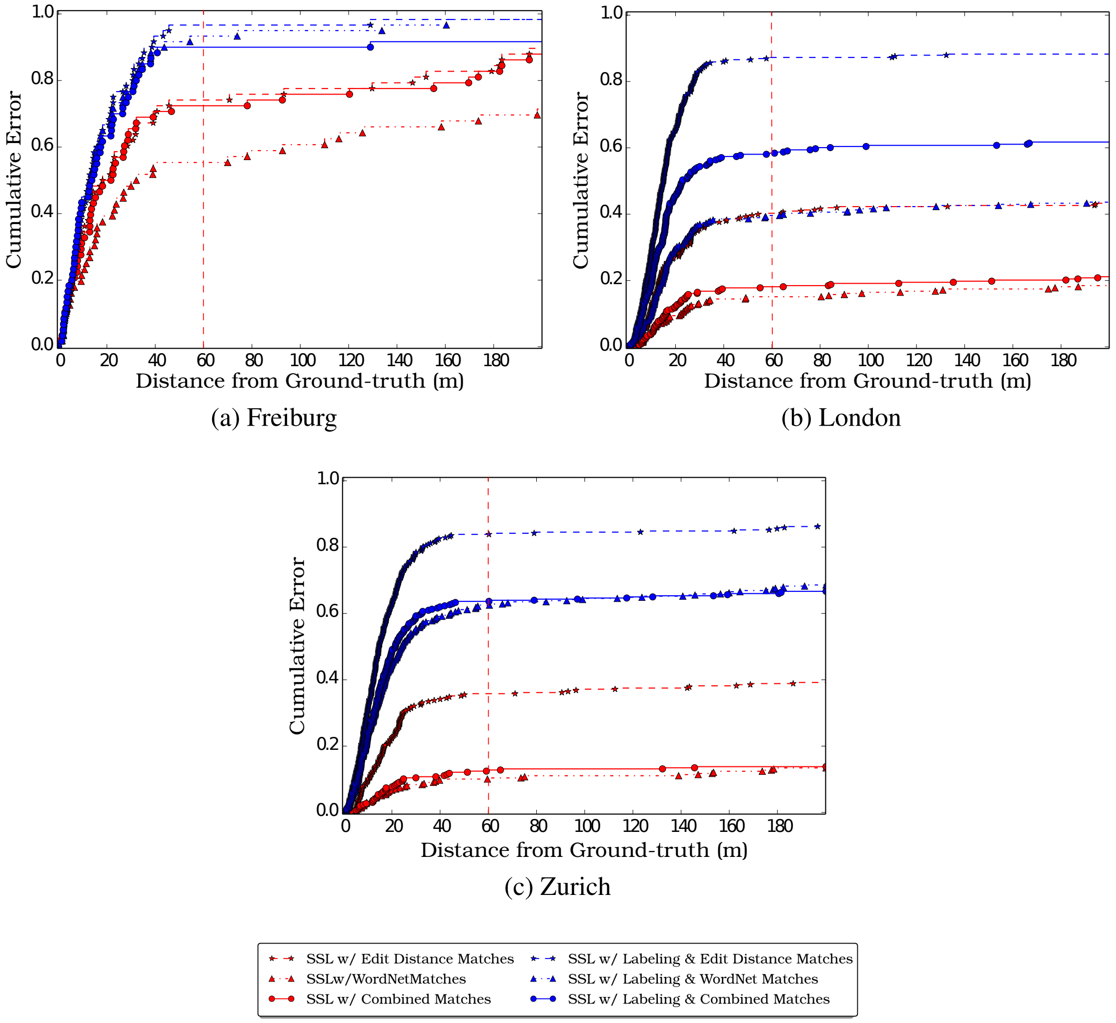
We plot the cumulative error of all paths of the Zurich dataset in Figure 3.3(c). Investigating the results shows that the PTL method achieves a localization accuracy of 1.0m for 35.0% of the poses with an overall average accuracy of 25.3m excluding localization failures. On the contrary, the SSL variant of our method is able to achieve an accuracy between 0.0 and 40.0m for only 35.0% of the poses. Furthermore, the number of localization failures due to text spotting errors increases by 50.0% in comparison to the PTL method. This further validates our hypothesis that by sharing information across the path, we enable our localization method to be more robust to text spotting and data association failures.

3.4.2 Text Spotting and Data Association Evaluation

In order to investigate the effect of the data association strategy on the localization accuracy, we plot for both the SSL and PTL approaches the cumulative localization error using each of the similarity measures proposed in Section 3.2.3.2. For each variant, we plot the performance using both the extracted text from the text spotting method of [141] and ground-truth text labels. Using ground-truth text labels enables us to evaluate the performance of the similarity measures independent of the text spotting method employed. Figure 3.4 and Figure 3.5 show the cumulative error plots for the SSL and PTL methods respectively on all three datasets. Similar to Figure 3.3, we highlight on each figure the localization failures using a red dashed line.

Investigating the results for the SSL method shown in Figure 3.4 shows that using the ground-truth text labels significantly improves the localization accuracy on all datasets. More precisely, the percentage of poses for which a localization error between 0.0 and $40.0\mathrm{m}$ is guaranteed increases by at least 20.0% by using the ground-truth text labels. This observation highlights the importance of the text spotting method used on the overall performance and since the SSL method attempts to estimate the pose from a single timestep without any prior information, its performance is highly dependent on the quality of the extracted text from the scene. Furthermore, investigating the performance of the proposed similarity measures shows that the best performance is using the Levenshtein similarity matching and the worst by utilizing the WordNet Lin similarity measure for all datasets. We attribute the suboptimal performance of the WordNet Lin similarly measure to be a direct consequence of using the English language WordNet while having words occurring in both German and English within our map for both the Freiburg and Zurich datasets.

Figure 3.5 shows the average cumulative error plots for the PTL method on all datasets. The results show that the performance of our PTL variant is neither affected by the text spotting method employed nor the data association strategy employed. This validates our hypothesis that sharing information across the timesteps in addition to using the odometry information helps in boosting the performance. One interesting observation to make in Figure 3.5(b) for the London dataset is that the localization accuracy is higher when employing the WordNet Lin similarity measure over the Levenshtein distance measure for data association. We hypothesize this occurs due to missed detections by the text spotting method. Since the Levenshtein distance does not account for the lexical relations between words, missed detections can result in incorrect matches and in turn a lower localization accuracy. As an example, consider the following text as the output of the text

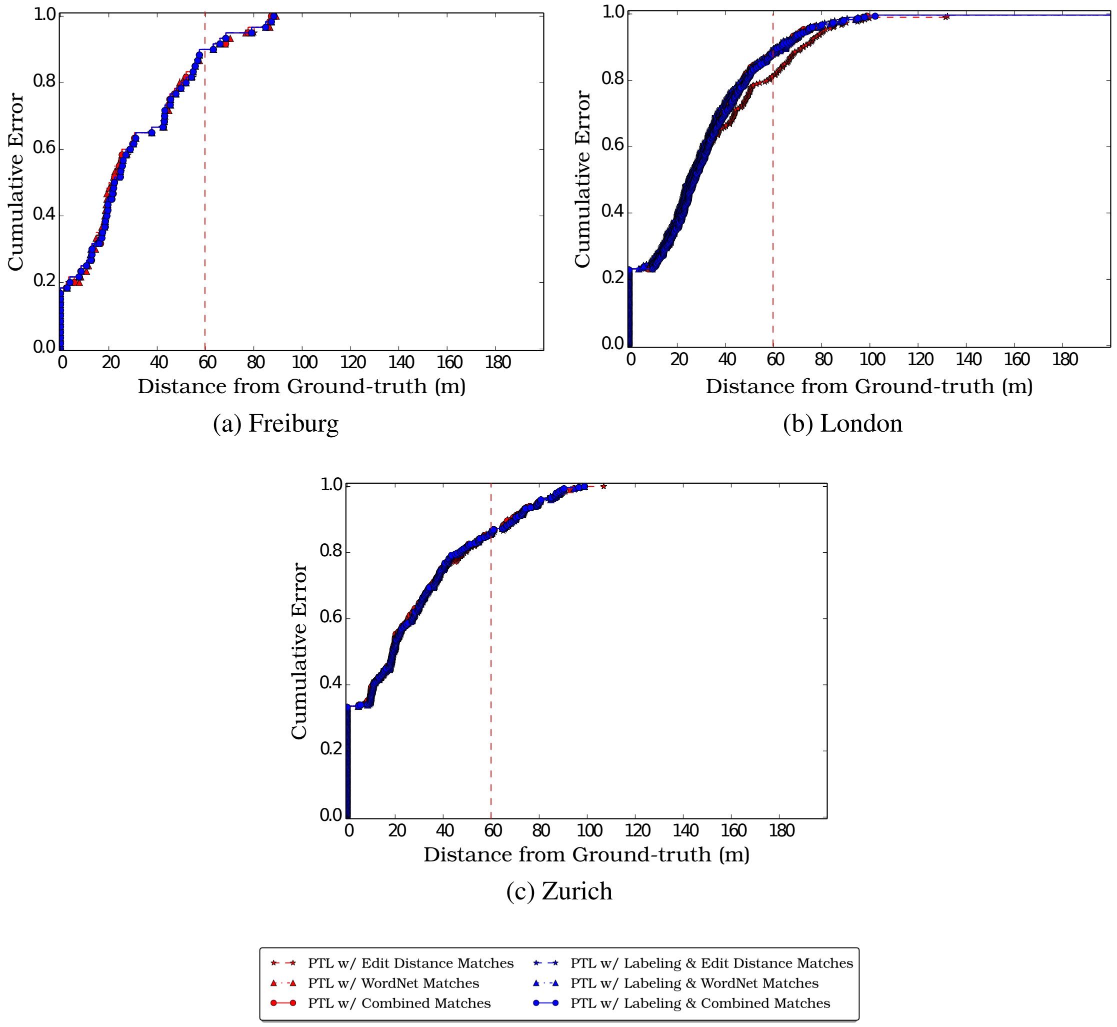


(a) Freiburg

(b) London

(c) Zurich

Figure 3.4: Cumulative error plot comparing between the performance of the different data association approaches for the Single-Shot Localization (SSL) approach on all datasets. The x-axis shows the distance from ground-truth position in meters, while the y-axis shows the percentage of sample-poses with a distance less than or equal to x.



(a) Freiburg

(b) London

(c) Zurich

Figure 3.5: Cumulative error plot on the Freiburg, London and Zurich datasets. The plot compares the performance of the different data association approaches and the best achievable performance for the Pose Tracking Localization (PTL) approach. The x-axis shows the distance from ground-truth position in meters, and the y-axis shows the percentage of sample-poses with distance less than or equal to the x-value.

spotting method: “insider”. The landmark with the highest probability match using the Levenshtein distance similarity measure is “nuntee”, while the correct match chosen by the WordNet Lin similarity is “insider dealings”. However, since “nuntee” has a smaller Levenshtein distance than “insider dealings”, it gets selected as a top matching landmark.

3.4.3 Ablation Study

In the following, we provide additional results evaluating the performance of the various parts of our localization framework. In order to gain an intuition into example scenarios where our localization method is unable to estimate the pose, in Figure 3.6 we illustrate three example scenarios from the London and Zurich datasets. Each column shows observation images from a single pose with red rectangles highlighting the text detection output. In Pose-1 (leftmost column), due to parallax error and text detection failures only the third observation was successfully associated to a landmark, leading to a localization error by our SSL method due to insufficient information. Similarly, for the remaining two poses, motion blur in the images and incorrect text detections have lead to a data association accuracy of 0% leading to unsuccessful localization.

Figure 3.7 illustrates successful localization runs from each of the benchmarking datasets using our SSL method. Pose-1 depicts an example from the Freiburg dataset. In each of the observation images, the text spotting method is able to accurately detect the text leading to successful data association and consequently an accurate localization estimate. For Pose-2 and Pose-3, despite the presence of noise in the text spotting output, during data association we are able to match a minimum of two out of the three observations leading to successful localization runs.

In order to qualitatively evaluate the accuracy of our proposed method for localization, in Figure 3.8, Figure 3.9 and Figure 3.10 we present example poses from the Freiburg, London and Zurich datasets respectively. For each pose, we show a segment of the map using Google Maps for visualization, along with the camera images capturing the observations. Moreover, for each observation we show the selected landmark assigned during the data association phase. The green star shows the ground-truth position, while the red shows the estimated location.

In Figure 3.8, despite correctly recognizing only two of the three landmarks, our SSL method is still able to accurately estimate the location with an error of 8.0m. Moreover, notice in Figure 3.9, despite the inability to detect the full text in all the observations, our method is still able to achieve a data association success of 100%. Selecting for each landmark the best N matches, enables our method to be robust to text spotting failures. Furthermore, Figure 3.10 depicts an example with only two observations where our localization method is able to accurately estimate the pose with an accuracy of 10.0m. Despite the presence of false positive detections by the text spotting method, utilizing the proposed post-processing spell correction technique enables us to discard most of

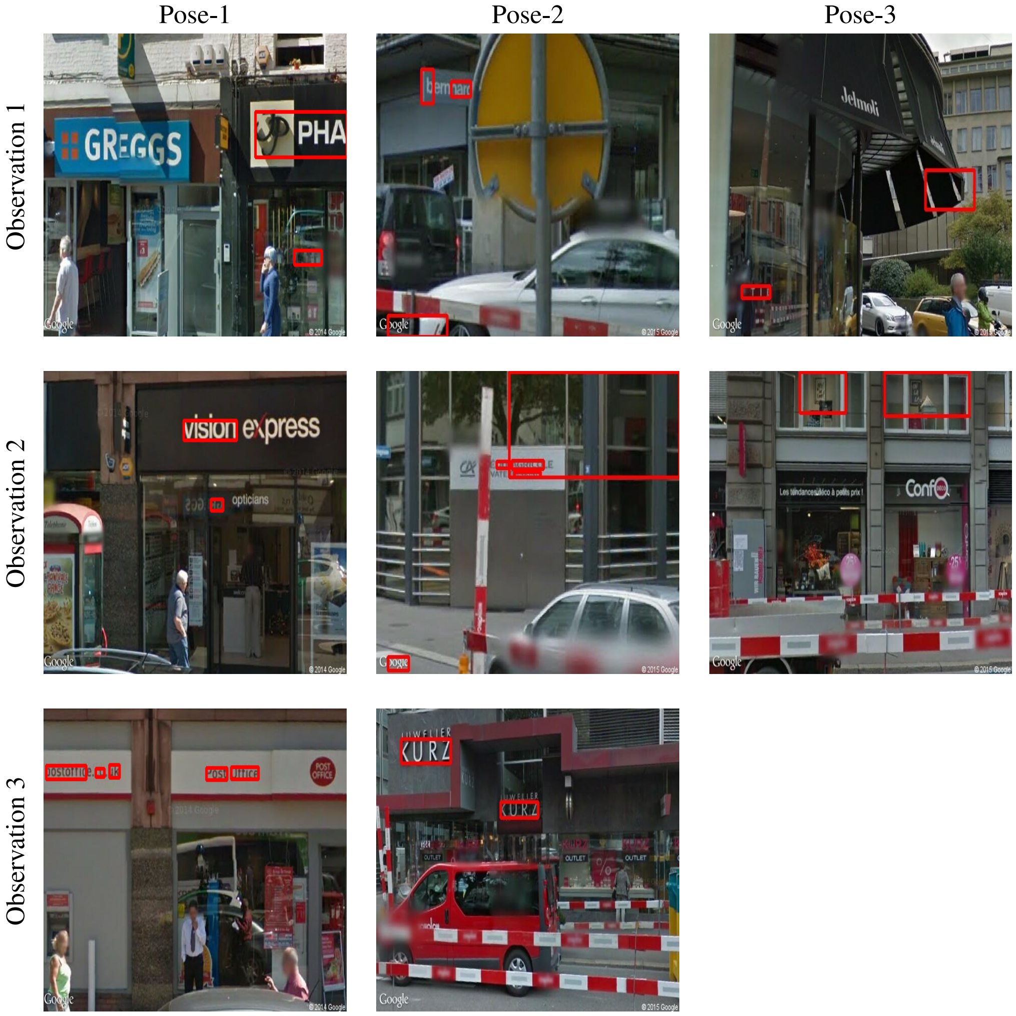


Figure 3.6: Example localization poses from the London and Zurich datasets. Each column depicts a single pose with the rows illustrating the captured observation images. Red rectangles highlight the text detections from the text spotting phase of our pipeline. For each of these poses, our Single-Shot Localization (SSL) method has been unable to localize due to text detection or data association failures.

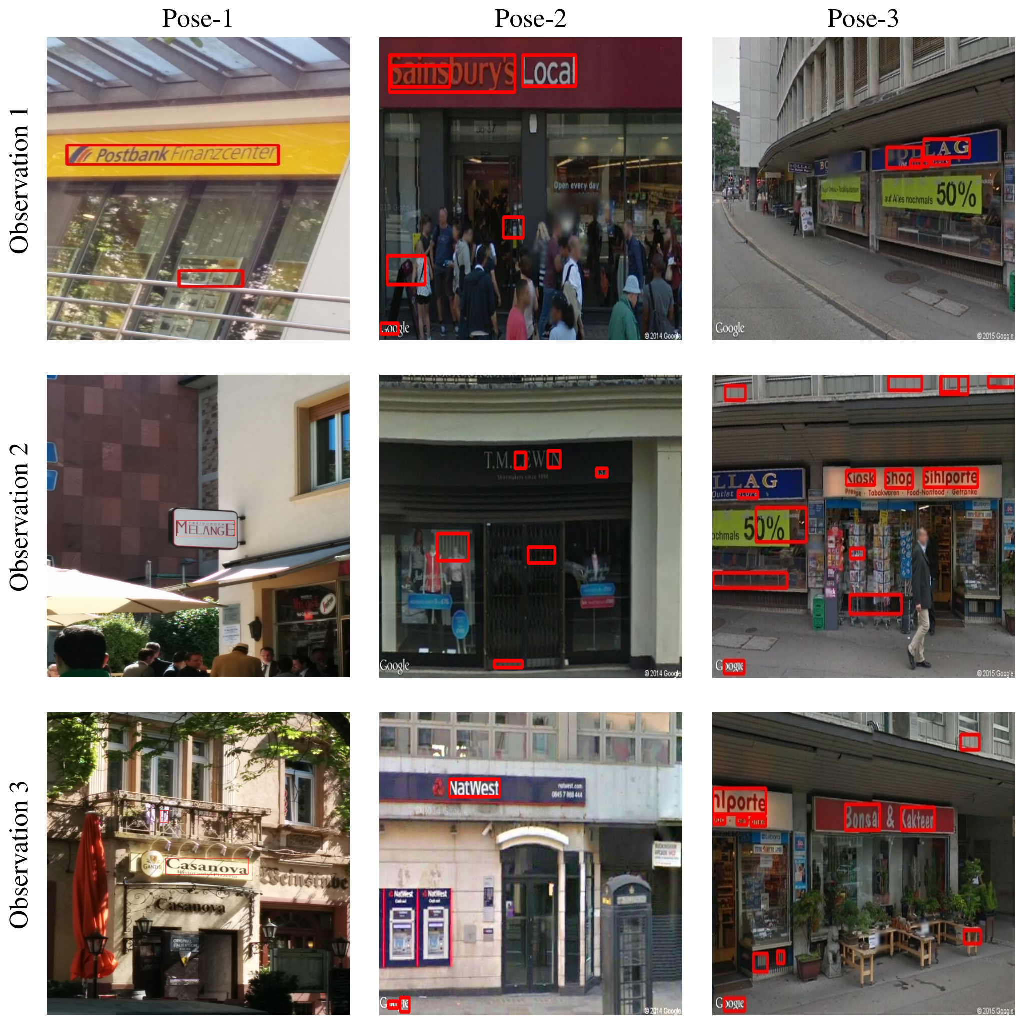


Figure 3.7: Example localization poses from the Freiburg, London and Zurich datasets. Each column depicts a single pose with the rows illustrating the captured observation images. Red rectangles highlight the text detections from the text spotting phase of our pipeline. For each of these poses, our Single-Shot Localization (SSL) method was able to successfully estimate the position despite noise in the text spotting method.

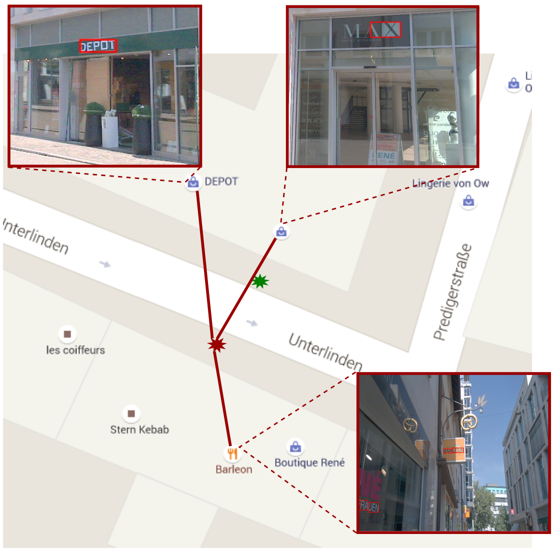


Figure 3.8: Example pose from the Freiburg dataset. The green star represents the ground-truth position, the blue star shows the estimated pose from our approach. Lines connect the pose with the observed landmarks. Red rectangles in the images show the output of the text-spotting phase.

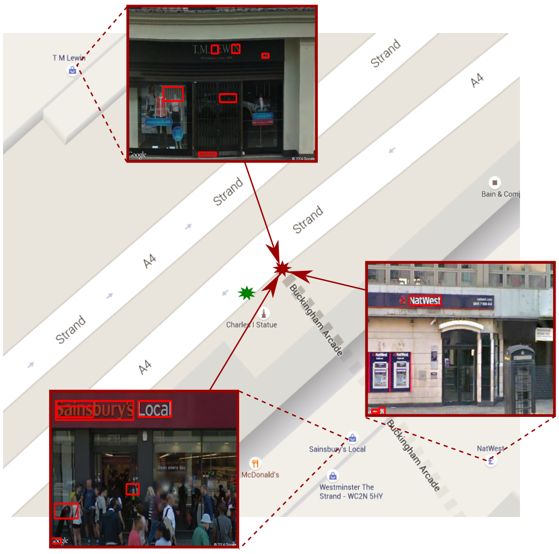


Figure 3.9: Example pose from the London dataset. The green star represents the ground-truth position, the blue star shows the estimated pose from our approach. Lines connect the pose with the observed landmarks. Red rectangles in the images show the output of the text-spotting phase.

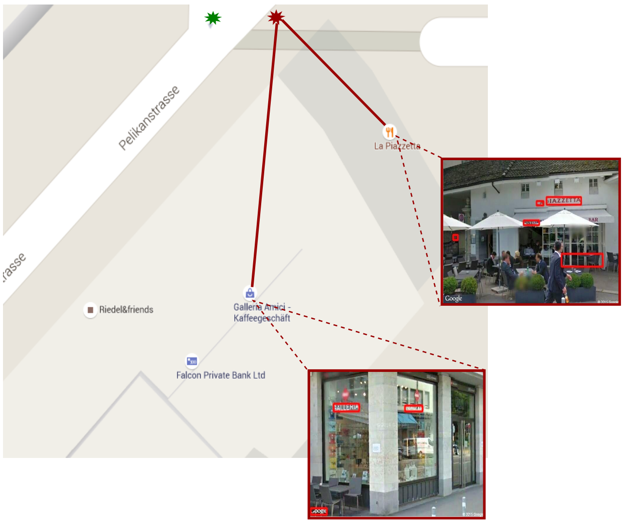


Figure 3.10: Example pose from the Zurich dataset. The green star represents the ground-truth position, the blue star shows the estimated pose from our approach. Lines connect the pose with the observed landmarks. Red rectangles in the images show the output of the text-spotting phase.

the false positives, and in turn reduces the amount of noise in the data association phase. A video illustrating the PTL performance on a sample path in Freiburg can be found at: `http://goo.gl/YNKx3v`.

3.5 Related Work

The use of visual information in localization and navigation tasks is becoming increasingly common due to low sensor cost, and rich information content. Visual Simultaneous Localization and Mapping (VSLAM) is an active research area, where visual sensors are used as the only information source [71]. In [63], the authors use an approximated Bayes network and a topological map of the environment for large-scale place recognition. They present a probabilistic approach for place recognition to overcome the aliasing problem. Furthermore, they use visual vocabulary along with Chow Liu tree for feature learning. Konolige and Agrawal [102] present a solution to the visual SLAM problem, in which they formulate the problem as a non-linear least squares optimization problem. They build a skeleton graph for both map estimation and data association, ensuring small footprint, long range tracking and both global and local registration.

Lothe *et al.* [122] use bundle adjustment combined with camera information and a coarse 3D model of the environment to perform monocular SLAM. Their approach takes advantage of road homography to reduce error accumulation, as it provides enough geometrical constraints to allow proper fitting of the captured cloud data with the environment model. Naseer *et al.* [139] present a solution to solving the visual SLAM problem across seasons. They use a deep convolutional neural network for feature extraction. Extracted features are used for building a similarity matrix between the different images from the database. Finally, using the similarity matrix, they use a flow network to perform sequence recognition. This information accompanied with odometry information is fed into a graph-based SLAM approach to find the most likely trajectory.

Techniques for localization using vision can be categorized as optimization-based techniques or retrieval-based techniques. Sattler *et al.* [167] introduce an approach for image-based localization in a 3D environment. Both the query image and the 3D model are represented by SIFT features from a visual vocabulary tree. Matching between an image point and a 3D point is accepted if the nearest neighbors of the points pass a certain correspondence ratio. Finally, they use RANSAC with 6-point DLT to estimate the pose. Another example for localization formulated as an optimization problem is the work of Qu *et al.* [151] in which they use local bundle adjustment and geo-referenced traffic signs. Starting from a known location, using local bundle adjustment, an estimate of the camera pose in relation to the start position is computed. Once a traffic sign is encountered, they run a traffic sign detection algorithm, where the detected signs are matched to the landmark database generating ground control points for the bundle adjustment process.

Torii *et al.* [184] aim to estimate the geo-coordinates of a query image from a large image dataset with known geo-coordinates. For a query image, they use a regression-based approach to search for nearby images using linear combinations of features. Once neighboring images have been selected, they apply an interpolation approach to estimate the final pose, where the nearest neighbor information is stored in a graph with nodes as images and edges as visual similarity between the nodes. An example of the retrieval-based techniques for localization is the work of Schindler *et al.* [170]. In their work, they acquire a large image dataset by driving in the city and collecting data with a camera. They build a vocabulary tree using visual features from the images using a hierarchical clustering approach. Given a query image, they use a voting scheme to find the closest matching database image that maximizes the information gain.

The use of publicly available information for localization is an active research area, where a number of techniques exist to utilize the abundance of information, in the form of images, for learning purposes. The work of Hayes and Efros [80] is an example of such approaches, where the authors build a database from geo-tagged Flickr images. They extract features from the images and using a clustering approach, quantify the similarity between a query image and the database images using a probability density function. Similarly, Crandall *et al.* [61] exploit Flickr's database for large-scale image localization. In their work, they build a database of geo-tagged and textual-tagged images of popular touristy locations. Using a clustering approach, they are able to obtain an estimate of the location of a query image, and build a map of popular photo destinations. Similarly, Google's Street View Maps provides a rich information source for localization. Zamir and

Shah [208] build a database from Google's Street View images, and using SIFT for feature extraction on a query image, they compute "interest points". Nearest neighbors from the database images are found using a voting scheme, and the final location is returned based on the localization confidence of the image.

Geo-tagged Street View panoramas are used by Agarwal *et al.* [23] for global localization, using a stream of captured images, they formulate the problem as a non-linear least squares estimation. First they find a transformation of the tracked feature points in the captured image series, then they compute a rigid body transformation between the points and the Street View panoramas. Features are computed using SIFT and clustered into a codebook, and image similarity is measured using cosine similarity. Majdik *et al.* [125] explore the problem of global localization for a micro aerial vehicle by utilizing Google Street View data. They generate virtual views from the Street View images exploiting the geometry of the system. Using a histogram-voting scheme, they select the highest voted image. Another approach using publicly available map information from OpenStreetMap is the work of Brubaker *et al.* [45], where they extract the map information into a simplified graph, and using visual odometry measurements along with a mixture of Gaussians model estimate the position and orientation of the vehicle.

Text features are being recently exploited in the computer vision and robotics fields. Schroth *et al.* [172] perform image retrieval using text-features from a query image. They apply a text detection method to detect text areas, which are later used as features for approximate string matching on an image database to retrieve image related to the query image. Similarly, Tsai *et al.* [187] use text-based content search for image retrieval, using word-HOG descriptors as features. They perform feature matching and return the best corresponding images from the database such that both the query image and the retrieved images share textual content.

In the robotics field, Schreiber *et al.* [171] present an approach to detect road side signs and markings using an Optical Character Recognition (OCR) system. The extracted information can be used for localization and planning tasks, and was left as future work. Posner *et al.* [150] extract text from natural scene images to retrieve images semantically relevant to a query. They use a generative probabilistic model to relate the extracted text to terms during run-time. Unlike the previously mentioned methods, in this chapter we present an approach for metric localization using textual information. To the best of our knowledge, this is the first localization method capable of providing localization estimates relying solely on the textual information in natural scene images. Through utilizing the textual information in the scene in addition to the publicly available maps, our proposed localization approach has low bandwidth requirements, global-scale breadth and life-long capabilities.

3.6 Conclusion

In this chapter, we presented a novel approach to the global localization problem that exploits the abundance of textual information in urban environments. Our method first extracts texts from the natural scene images, associates it to a map consisting of landmarks and corresponding text labels and then estimates the pose of the camera using the observation angle.

We proposed two variants of our localization approach: a single-shot variant (SSL) which estimates the position using only information from the current timestep and a pose tracking variant (PTL) which shares information across the different poses in order to obtain a localization estimate that is tolerant to the amount of textual information available. We use an off-the-shelf text detection and recognition framework to extract textual information from the captured camera images. In order to remove noise from the extracted text, we use a probabilistic spell correction approach. Furthermore, we proposed three methods for measuring the similarity between the extracted text and landmarks by utilizing both distance metrics and linguistic features of the text. Utilizing our proposed data association method, we select for each observation a set of potential matching landmarks and utilize a particle filter-based approach with a dedicated sensor model to estimate the pose.

We evaluated our proposed approach on data captured from three different cities in Europe: Freiburg, London and Zurich by comparing the performance with GPS. Furthermore, we evaluated the performance of the proposed landmark selection methods and the effect of the text spotting method on the localization pose accuracy. The results demonstrate that our proposed single-shot global localization (SSL) method achieves a localization accuracy of up to $1\mathrm{m}$, which corresponds to a 40% improvement over GPS poses obtained with a mobile device. Moreover, by sharing information across the various poses, our approach is more tolerant to text detection failures and guarantees a maximum localization error of $23.0\mathrm{m}$ for 60% of the poses. Unlike feature-based visual localization approaches, our proposed method is robust to scene and environmental changes. Our proposed method only requires a stream of camera images and any publicly available map of the area, making our approach both efficient for systems with constrained resources and easy to deploy. The obtained results demonstrate the efficacy of using text as a source of information for localization in urban environments.

Chapter 4

Multitask Learning for Geometry and Semantics-Aware Pose Regression

Semantic scene understanding and localization are indispensable components of the robot's autonomy stack and natural precursors for any action execution or planning task. Despite the shared interdependencies between semantic scene understanding and localization, they have been for the most part tackled as disjoint problems. In this chapter, we propose a multitask learning architecture for learning semantics, visual localization and odometry estimation. We utilize the VLocNet++ architecture which employs multitask learning to jointly predict the semantic structure of the scene as well as regress the 6-DoF global pose and ego-motion. We propose a novel loss function that employs auxiliary learning to leverage the relative pose information during training, thereby embedding geometric knowledge of the world into the pose regression network. In order to aggregate motion-specific temporal information and incorporate semantic features into the localization network stream, we use a novel adaptive weighted fusion layer based on region activations. Furthermore, we propose a self-supervised warping technique that uses the relative motion to warp intermediate network representations in the segmentation stream for learning consistent semantics. Finally, we introduce a first-of-a-kind urban outdoor localization dataset with pixel-level semantic labels and multiple loops. Extensive experiments on the challenging Microsoft 7-Scenes benchmark and our DeepLoc and DeepLocCross datasets demonstrate that our network surpasses the state-of-the-art, outperforming local feature-based methods while simultaneously performing multiple tasks and exhibiting substantial robustness in challenging scenarios.

4.1 Introduction

In Chapter 3, we proposed a localization approach targeted towards outdoor urban environments which utilizes the textual features in the scene. In order to identify the location, our method extracted textual features corresponding to shop fronts and street signs from captured RGB images of the scene, and leveraged the information in a probabilistic manner with the use of publicly available maps to produce a location estimate. While the accuracy of the pose estimate from the method proposed in Chapter 3 outperforms GPS in urban areas, it is constrained to regions containing textual features. However, as robots often navigate in different areas of the environment, they can encounter locations with sparse or no textual features. In such scenarios, the lack of textual information would restrict the application of our text-based localization method. In order to address this limitation, in this chapter we propose a visual localization method that incorporates geometric and semantic information of the scene into the pose regression pipeline.

Visual localization is a crucial component for various robotics and computer vision systems such as Simultaneous Localization and Mapping (SLAM) [183], Augmented Reality (AR) [126], and autonomous navigation [75]. In order to satisfy the goal of reliable robotic deployment in various environments, localization systems need to be robust to changes in the scene resulting from illumination and seasonal variation, dynamic objects such as people or vehicles, and structural variation such as demolition and construction of buildings.

Visual localization techniques can be classified as either topological [140] or metric-based [23] methods. Topological localization methods usually divide the environment into a discrete set of locations and provide coarse estimates of the position within those discretized cells by employing image retrieval techniques [25, 62, 180]. On the one hand, employing topological localization approaches enables robust localization in large environments. On the other hand, the localization accuracy is bounded by the size of the discrete set. Metric-based localization approaches produce a 6-DoF estimate of the robot pose within the environment. For the most part, feature-based approaches that employ SfM information achieve state-of-the-art performance on several benchmarking datasets [169, 195]. However, the run-time and complexity of finding feature correspondences using such methods grows with the size of the environment. Moreover, failures often occur with drastic changes in the viewpoint, motion blur or occlusions due to the requirement of having a minimum number of matches to produce pose estimates.

The recent success of convolutional neural networks in numerous tasks has lead to a surge in the number of approaches employing deep learning for estimating the robot pose [94, 138, 200]. Although convolutional neural networks are more robust to appearance variations than local feature-based methods, their performance remains an order of magnitude worse in comparison. Since a majority of the deep learning-based approaches attempt to directly regress the 6-DoF pose using a single image of the scene, they are unable to accurately model the 3D structure of the scene which subsequently leads to the inferior performance. To address this shortcoming, we propose a novel loss function that enables embedding the geometric knowledge of the scene by leveraging auxiliary learning to jointly estimate the ego-motion of the robot. We then utilize the relative motion information in our Geometric Consistency loss function to constrain the search space during training.

In order to ensure the learning of the inter-task correlation between the visual odometry estimation and global pose regression tasks, we employ the VLocNet architecture (Section 2.6) for simultaneously predicting the ego-motion and the global pose of the robot. We further employ our novel Geometric Consistency loss function which enables the learning of consistent global poses by incorporating the relative motion information from earlier timesteps. Utilizing the VLocNet architecture with the proposed loss function enables the network to learn a more accurate representation of the scene, while allowing it to be robust to appearance changes in the scene.

Inspired by our work in Chapter 3, and how humans often describe their location with respect to surrounding landmarks, we further explore incorporating semantic knowledge of the scene into our pose regression network. To this end, we propose the VLocNet++ architecture [157] to jointly learn semantic segmentation, global localization and visual odometry estimation by reformulating the problem from the multitask learning perspective. Simultaneously learning tasks across a wide variety of domains is, however, challenging due to the difference in units and scales of the various loss functions. Nonetheless, utilizing a joint formulation promotes inter-task learning which in turn improves the generalization capabilities of the network. Furthermore, as labeled real-world data is hard to obtain in the robotics domain, simultaneously learning multiple tasks mitigates the problem of requiring vast amounts of task-specific training data. Additionally, deploying a single joint model is more efficient in enabling online performance capabilities which is crucial for robots with limited resources operating in the real world.

We believe that jointly learning semantics of the scene enables the pose regression network to learn a more stable structural representation of the environment by drawing the attention of the network towards more informative regions within the scene. Similarly, incorporating location-specific information from the pose regression network can help improve the accuracy of the predicted segmentation masks. Current approaches for semantics-aware visual localization rely on predefined structures in the scene and either extract features from the structures, emphasize the stable features [99] or combine them with local features [177]. However, the absence of the predefined structures due to scenarios such as occlusions, results in a substantial degradation in the performance of these methods. As a solution to this problem, we employ a novel layer for aggregating information from multiple sources. Our proposed adaptive weighted fusion layer [157] is able to fuse relevant features from the semantic segmentation stream into the localization stream based on both the semantic category and region activations.

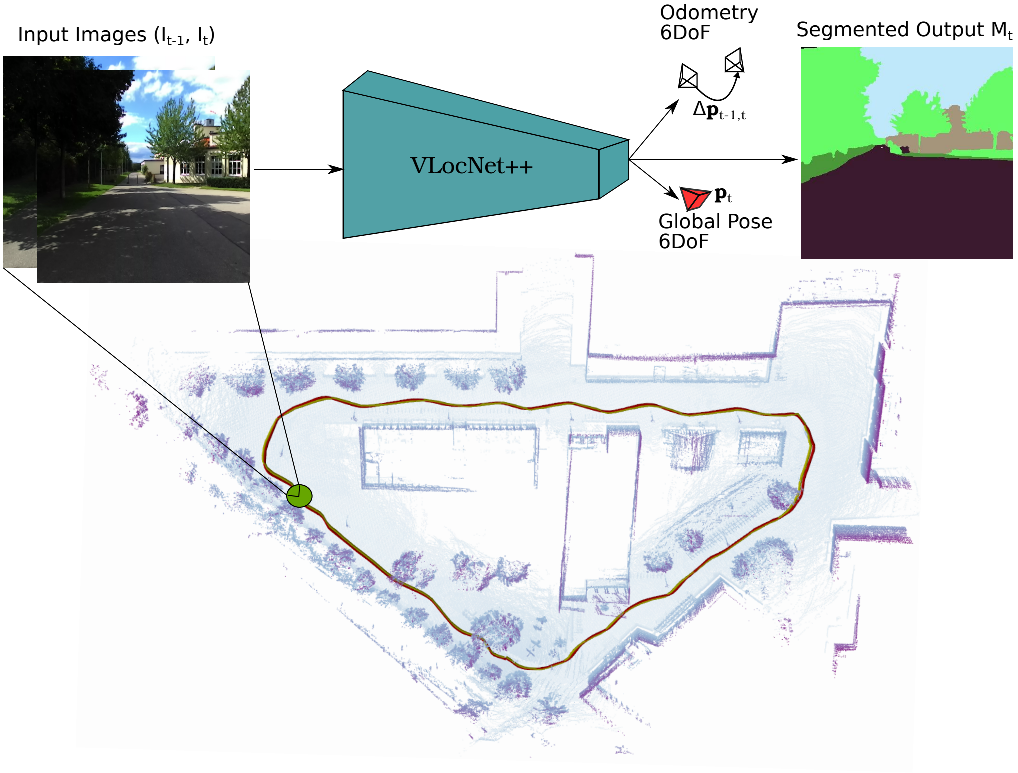


Figure 4.1: Given a pair of consecutive monocular images (I_{t-1}, I_t), our proposed VLocNet++ architecture [157] predict the 6-DoF global pose $\mathbf{p}_t$, the ego-motion $\Delta\mathbf{p}_{t-1,t}$ and semantics of the scene M_t. The results are shown for the second testing sequence of the DeepLoc dataset. VLocNet++ achieves accurate pose estimates by leveraging semantic and geometric knowledge from the environment, as well as aggregating information from the previous timestep.

A critical prerequisite for the task of semantic visual localization is the ability to predict consistent semantics. Early cognitive studies demonstrated that learning self-motion is crucial for humans to acquire basic perceptual skills [152]. Taking inspiration from this work, we propose a novel semantic context aggregation technique that leverages the predicted ego-motion of the robot to improve the temporal consistency of the semantic segmentation predictions. We propose the novel adaptive weighted fusion layer that employs differential warping to intermediate representations of the network, transforming the features from the previous timestep to the current timestep using pixel-wise depth predictions as an external input [127] and the relative poses from the odometry stream of our network. This in turn improves the performance of our semantic predictions as well as leads to faster convergence times as the network learns to aggregate more scene-level information. Figure 4.1 depicts the output of our proposed VLocNet++ architecture given two input images.

In order to facilitate training the proposed multitask models, and due to the lack of publicly available datasets with labels for relocalization, ego-motion and semantic segmentation, we introduce two new datasets: DeepLoc and DeepLocCross. The datasets were captured using our robotic platform presented in Section 2.1 in varying outdoor environments. The DeepLoc dataset is comprised of multiple trajectories traversed on a university campus and contains RGB-D images with pixel-level semantic and 6-DoF pose labels. The dataset contains repetitive, reflective and multiple weakly textured regions which in turn make it extremely challenging for the tasks at hand. The DeepLocCross dataset, on the other hand, was captured in a highly dynamic road environment covering multiple road intersections and pedestrian crossings. For this dataset, we provide the RGB-D images with 6-DoF pose labels as well as the trajectories of all dynamic objects in the scene and the intervals at which the pedestrian crossings are safe for crossing. The highly dynamic nature of the dataset and the presence of occlusions render this dataset challenging for relocalization and pose estimation tasks, as well as behavior and motion prediction tasks. In addition to the aforementioned datasets, we further benchmark our methods on the Microsoft 7-Scenes dataset [175]. Extensive experimental evaluations demonstrate that employing our proposed loss function and self-supervised context aggregation technique enables the network to learn accurate representations of the environment, thereby setting the new state of the art while simultaneously predicting multiple tasks. In summary, the primary contributions of this chapter are as follows:

- We introduce the novel Geometric Consistency loss function (Section 4.2.1) that enables the network to encode geometric and temporal motion information by exploiting the relative motion information.

- We propose the novel self-supervised warping layer (Section 4.2.2.3) that enables temporal semantic context aggregation, thereby improving the accuracy of the semantic predictions and reducing the training time.

- We introduce the DeepLocCross dataset (Section 4.3.3) captured in an outdoor dynamic environment for localization, ego-motion estimation, motion and behavior prediction tasks. The dataset consists of multiple loop traversals in a highly dynamic environment with multiple weakly textured and low-light regions, thereby rendering it extremely challenging for a number of computer vision and robotics tasks.

- We present extensive evaluation of the proposed contributions on multiple indoor and outdoor datasets. The results demonstrate the accuracy and robustness of employing our proposed techniques to the tasks of visual localization and semantic segmentation.

Furthermore, we include the following contributions which are an outcome of joint work with Abhinav Valada [157, 193]:

- We introduce a novel multitask learning VLocNet++ architecture (Section 4.2.2) for simultaneously predicting the global pose, ego-motion and semantics from consecutive monocular images.
- We propose a novel adaptive weighted fusion layer (Section 4.2.2.4) for element-wise fusion of feature maps based on region activations to exploit inter- and intra-task dependencies.
- We introduce the DeepLoc dataset (Section 4.3.2) consisting of multiple loops with pixel-level semantic labels and localization ground-truth. It contains repetitive, translucent and reflective surfaces, thereby making it extremely challenging for benchmarking a variety of tasks.
- We present comprehensive quantitative comparisons on the performance of our task-specific networks as well as our multitask networks with deep learning-based approaches and state-of-the-art local feature-based techniques on publicly available benchmarking datasets.

4.2 Technical Approach

In the following section, we formalize the problem of estimating geometrically consistent poses. We begin by introducing our proposed Geometric Consistency (GC) loss function that enables the network to predict poses consistent with the motion model. Subsequently, we present the multitask VLocNet++ architecture [157] for simultaneously predicting the semantic segmentation, visual localization and ego-motion given a pair of consecutive monocular images. Note that while the presented architecture focuses on multitask learning, each of the constituting task-specific models can be deployed independently during inference.

4.2.1 Geometric Consistency Loss

While the majority of existing deep learning-based approaches for global pose regression [94, 138, 200] directly minimize the Euclidean loss function between the ground-truth and predicted poses, their performance is suboptimal in comparison to sparse feature-based localization approaches. This comes as a direct consequence of attempting to learn the full 3D structure of the scene using only a single monocular image at a time. In this chapter, as opposed to naively minimizing the Euclidean loss function, we propose the novel GC loss function in order to learn accurate global pose estimates, which in addition to minimizing the Euclidean loss, adds another loss term to constrain the current pose prediction by minimizing the relative motion error between the ground-truth and the estimated motion from the odometry stream. Utilizing the predictions of the network from the current and the previous timesteps, the relative motion loss term $\mathcal{L}_{Rel}\left(f\left(\theta \mid I_t\right)\right)$, defined in Equation (4.1), can be computed as a weighted sum of the translational and rotational errors between the ground-truth relative motion and the relative motion computed from the aforementioned predictions; where θ is defined to be the internal parameters of the network, and $f(\theta \mid I_t)$ denotes the predicted output of the network for image I_t. Learning both translational and rotational pose components within the same loss function is inherently challenging due to the difference in scale and units between both quantities. Although initial works [95, 199] have employed an external hyperparameter β to counteract this problem, this nonetheless adds the extra prerequisite of manually tuning the value of β for each new scene to achieve reasonable results. We replace this β term with two learnable weighting variables $\hat{s}_x$ and $\hat{s}_q$ for the translational and rotational components of the loss, respectively. As the variables are learnable, their values get updated during the optimization process and consequently do not require manual tuning. In order to formalize our proposed loss function, we first define the global and relative pose terms. We define the global pose for image I_t as $\mathbf{p}_t = (\mathbf{x}_t, \mathbf{q}_t)$ with $\mathbf{x} \in \mathbb{R}^3$ denoting the position and $\mathbf{q} \in \mathbb{R}^4$ denoting the orientation in quaternion representation. The relative motion between the image pair (I_{t-1}, I_t) is then denoted by $\Delta\mathbf{p}_{t-1,t} = (\Delta\mathbf{x}_{t-1,t}, \Delta\mathbf{q}_{t-1,t})$, which can be computed from the global poses $\mathbf{p}_{t-1}$ and $\mathbf{p}_t$ using transformation geometry such that $\mathbf{p}_t = \mathbf{p}_{t-1} \oplus \Delta\mathbf{p}_{t-1,t}$, where $\oplus$ denotes the concatenation of transforms. Equation (4.1) details the relative motion loss term, in which we assume that the quaternion output of the network has been normalized a priori for ease of notation.

$$\begin{aligned}
\mathcal{L}_{Rel}\left(f\left(\theta \mid I_t\right)\right) = & \exp(-\hat{s}_{x_{Rel}})\mathcal{L}_{x_{Rel}}\left(f\left(\theta \mid I_t\right)\right) + \hat{s}_{x_{Rel}} \\
& + \exp(-\hat{s}_{q_{Rel}})\mathcal{L}_{q_{Rel}}\left(f\left(\theta \mid I_t\right)\right) + \hat{s}_{q_{Rel}} \\
\mathcal{L}_{x_{Rel}}\left(f\left(\theta \mid I_t\right)\right) := & \left\|\Delta\mathbf{x}_{t-1,t} - \Delta\hat{\mathbf{x}}_{t-1,t}\right\|_2 \\
\mathcal{L}_{q_{Rel}}\left(f\left(\theta \mid I_t\right)\right) := & \left\|\Delta\left(\mathbf{q}_{t-1,t}\right)^{-1}\Delta\hat{\mathbf{q}}_{t-1,t}\right\|_2 .
\end{aligned} \tag{4.1}$$

The terms $\hat{\mathbf{x}}, \hat{\mathbf{q}}$ denote the predicted position and rotation of the network, and $\hat{s}_{x_{Rel}}, \hat{s}_{q_{Rel}}$ denote the learnable weighting parameters for the relative motion loss term $\mathcal{L}_{Rel}$. Following the aforementioned notation, the Euclidean loss term can be defined as

$$\begin{aligned}
\mathcal{L}_{Euc}\left(f\left(\theta \mid I_t\right)\right) = & \exp(-\hat{s}_x)\mathcal{L}_x\left(f\left(\theta \mid I_t\right)\right) + \hat{s}_x \\
& + \exp(-\hat{s}_q)\mathcal{L}_q\left(f\left(\theta \mid I_t\right)\right) + \hat{s}_q \\
\mathcal{L}_x\left(f\left(\theta \mid I_t\right)\right) := & \left\|\mathbf{x}_t - \hat{\mathbf{x}}_t\right\|_2 \\
\mathcal{L}_q\left(f\left(\theta \mid I_t\right)\right) := & \left\|\left(\mathbf{q}_t\right)^{-1}\hat{\mathbf{q}}_t\right\|_2 .
\end{aligned} \tag{4.2}$$

The final GC loss term to be minimized is

$$\mathcal{L}_{GC}\left(f\left(\theta \mid I_t\right)\right) := \mathcal{L}_{Euc}\left(f\left(\theta \mid I_t\right)\right) + \mathcal{L}_{Rel}\left(f\left(\theta \mid I_t\right)\right). \tag{4.3}$$

By minimizing the aforementioned loss function, our network learns a model that is geometrically consistent with respect to the motion.

4.2.2 Semantic Visual Localization and Odometry

In this section, we describe the proposed VLocNet++ architecture [157] for jointly estimating the global pose, odometry and semantic segmentation from consecutive monocular camera images. Similar to the VLocNet architecture (Section 2.6), each of the task-specific models can be deployed independently during test time. In order to encode geometric and structural constraints into the pose regression network, we propose to employ our adaptive weighted fusion layer to incorporate information from the previous timestep, thereby accumulating motion-specific information based on the region activations. In order to reinforce the geometric constraints in the pose regression network, we employ the GC loss function (Section 4.2.1) which constrains the search space using the relative motion between two consecutive frames. As predicting robust and consistent semantics is an essential requirement for the proposed fusion framework, we present a new self-supervised warping technique which aggregates scene-level context information into the semantic segmentation stream. Our architecture depicted in Figure 4.2 consists of four convolutional neural network streams: a global pose regression stream, a Siamese-type double stream for visual odometry estimation and a semantic segmentation stream.

Given a pair of consecutive monocular images $I_{t-1}, I_t \in \mathbb{R}^{\rho}$, where $\rho = H \times W$ denotes the number of pixels in an input image, the global pose regression stream uses the current image I_t as input to predict the pose $\mathbf{p}_t$, the odometry stream predicts the relative motion $\Delta\mathbf{p}_{t-1,t}$ resulting from the input frames (I_{t-1}, I_t), and the semantic segmentation stream predicts a segmentation mask M_t mapping each pixel u to one of the C semantic classes. In the remainder of this work, we denote feature maps from layer l of a particular stream using $\mathbf{z}^l$. In the following, we describe the various components of our network and the underlying multitask learning framework.

4.2.2.1 Geometry-Aware Visual Localization

Our architecture for estimating the global pose is built upon the ResNet-50 architecture [82] with pre-activation residual units truncated before the last average pooling layer. Apart from utilizing pre-activation residual units, it follows the same structure as the VLocNet architecture. However, as opposed to employing inner product layers to directly fuse the previous predicted pose as in VLocNet, we adopt a more methodological approach to provide the network with this prior. Directly fusing the previous pose prediction as opposed to intermediate network representations from the previous timestep prevents the network from learning to correlate the underlying motion-specific temporal spatial relations. In order to enable our network to learn the geometric and spatial relations of the environment, we propose integrating the intermediate representation $\mathbf{z}_{t-1}^{5a}$ from the last downsampling stage (*Res5a*) of the previous timestep using our proposed adaptive weighted fusion layer detailed in Section 4.2.2.4. Our fusion scheme enables the network to learn the most favorable element-wise weighting for each component, and when trained

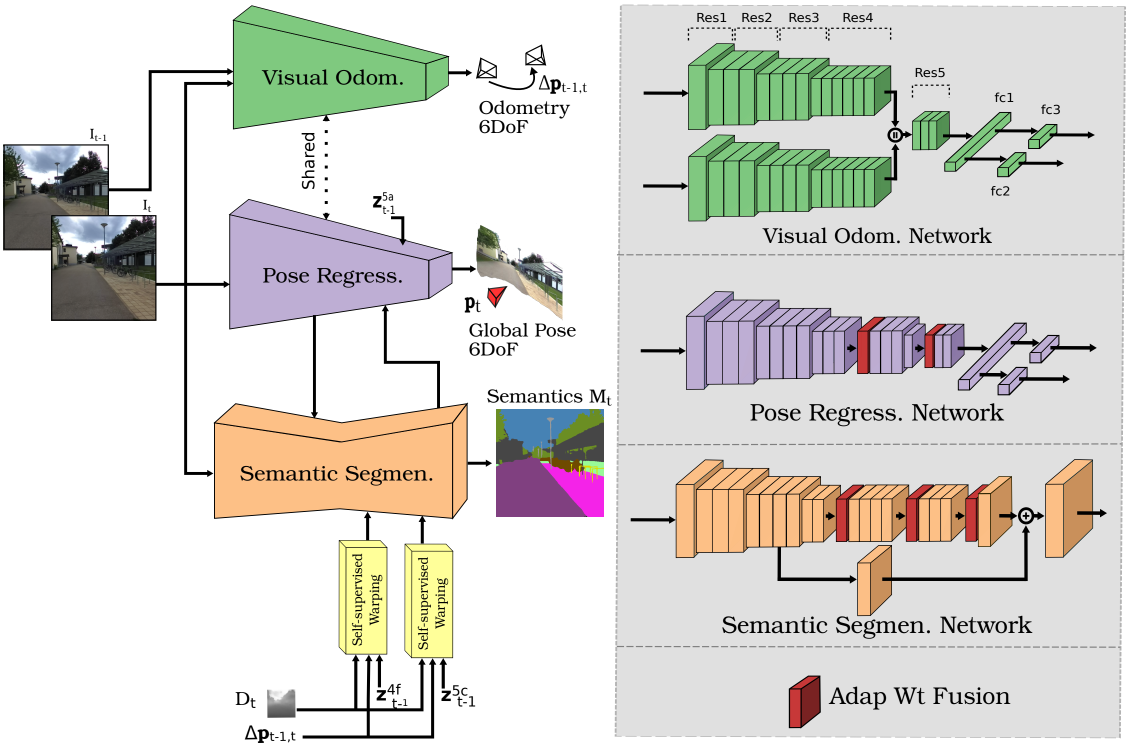


Figure 4.2: Schematic representation of our proposed VLocNet++ architecture [157]. The network takes two consecutive monocular images (I_{t-1}, I_t) as input and simultaneously predicts the global 6-DoF pose $\mathbf{p}_t$, odometry $\Delta\mathbf{p}_{t-1,t}$ and semantics M_t of the scene. The term $\mathbf{z}^l_{t-1}$ denotes the feature maps of layer l from the previous timestep and D_t denotes an externally predicted depth map used for representational warping in the semantic stream. The legend enclosed in gray shows the building blocks of the streams.

end-to-end with the GC loss, enables it to leverage the motion-specific feature cues across the temporal dimension. In Appendix A.4, we evaluate the performance of our fusion technique in comparison to standard methods for feature sharing. We denote the aforementioned architecture for global pose regression with the adaptive weighted fusion layer as VLocNet++$_{\text{STL}}$. Similar to our VLocNet architecture (Section 2.6), we employ the GC loss function during training to enable the network to learn geometrically and temporally consistent poses. Moreover, by employing a mechanism to aggregate motion-specific features temporally, we are able to efficiently leverage this information.

4.2.2.2 Visual Odometry Estimation

We use the same Siamese-type network architecture from VLocNet (Section 2.6) for visual odometry estimation. However, contrary to VLocNet, we use the full pre-activation ResNet architecture [82] instead of the standard residual units (Section 2.5.2). Given a pair of consecutive input images (I_{t-1}, I_t), the network predicts the relative motion

$\Delta\mathbf{p}_{t-1,t}$ by minimizing the Euclidean loss between the ground-truth and the predicted relative poses during training as follows:

$$\begin{aligned}
\mathcal{L}_{vo}\left(f\left(\theta \mid I_{t-1}, I_t\right)\right) :=& \exp(-\hat{s}_{x_{vo}})\mathcal{L}_x\left(f\left(\theta \mid I_{t-1}, I_t\right)\right) + \hat{s}_{x_{vo}} \\
&+ \exp(-\hat{s}_{q_{vo}})\mathcal{L}_q\left(f\left(\theta \mid I_{t-1}, I_t\right)\right) + \hat{s}_{q_{vo}} \\
\mathcal{L}_x\left(f\left(\theta \mid I_{t-1}, I_t\right)\right) :=& \left\|\Delta\mathbf{x}_{t-1,t} - \Delta\hat{\mathbf{x}}_{t-1,t}\right\|_2 \\
\mathcal{L}_q\left(f\left(\theta \mid I_{t-1}, I_t\right)\right) :=& \left\|\left(\Delta\mathbf{q}_{t-1,t}\right)^{-1}\Delta\hat{\mathbf{q}}_{t-1,t}\right\|_2 .
\end{aligned} \tag{4.4}$$

Similar to the GC loss function, we employ learnable weighting parameters $(\hat{s}_{x_{vo}}, \hat{s}_{q_{vo}})$ to balance the scale between the translational and rotational components in the loss term.

4.2.2.3 Temporally-Consistent Semantic Segmentation

We propose two variants of our semantic segmentation network: a single-task base architecture that takes as input a monocular image and outputs a predicted pixel-wise segmentation mask of the image and a multitask variant built upon the single-task model that incorporates our proposed self-supervised warping and adaptive weighted fusion layers (represented by the yellow and red blocks in Figure 4.2). In the following, we present the single-task base architecture followed by the proposed self-supervised warping layer.

Network Architecture

For our base network model, we employ the AdapNet [191] architecture which consists of a contractive and an expansive segment. Similar to both the global pose and visual odometry networks, the encoder segment of the network is based on the ResNet-50 architecture [81]. The encoder learns highly discriminative features and yields an output 16-times downsampled with respect to the input dimensions. Furthermore, the encoder incorporates multi-scale blocks to enable the generation of features at multiple scales throughout the network without increasing the number of parameters. This is achieved by replacing the 3×3 convolution inside the residual block with two parallel 3×3 convolutions with half the number of feature maps followed by an element-wise concatenation of the outputs resulting in the same number of channels as in the original block. In addition to enabling the network to learn features from different scales, the element-wise concatenation employed within the multi-scale blocks preserves the features thus enabling the network to learn combining features that are generated on different scales. The decoder consists of two deconvolution layers and a skip convolution from the encoder which fuses the high resolution feature maps as well as upsamples the downscaled feature maps back to the input resolution.

We define the set of training images $\mathcal{T} := \{(I_n, M_n) \mid n = 1, \ldots, N\}$, where $I_n = \{u_r \mid r = 1, \ldots, \rho\}$ denotes the input frame and ρ denotes the number of pixels. The corresponding ground-truth mask is denoted by $M_n = \{m_r^n \mid r = 1, \ldots, \rho\}$, where $m_r^n \in \{1, \ldots, C\}$ represents the set of semantic classes. Using θ to denote the internal parameters of the network, $s_j(u_r, \theta)$ denotes the score assigned to labeling pixel u_r with label j. We obtain the probabilities $\mathbf{p} = (p_1, \ldots, p_C)$ for all the semantic classes using the softmax function σ [37] as follows:

$$p_j(u_r \mid \theta, I_n) = \sigma\left(s_j\left(u_r, \theta\right)\right) = \frac{\exp\left(s_j\left(u_r, \theta\right)\right)}{\sum_{k=1}^{C} \exp\left(s_k\left(u_r, \theta\right)\right)} \tag{4.5}$$

The optimal network parameters are then estimated by minimizing the cross-entropy loss function (see Section 2.5.3).

Self-Supervised Warping

In order to aggregate scene-level context for learning consistent semantics, we employ the representational warping concept from multi-view geometry. By incorporating feature maps from multiple views and resolutions, we enable our model to be robust to camera angle deviations, object scale and frame-level distortions. We thereby also implicitly introduce feature augmentation hence facilitating faster convergence. We leverage the estimated relative pose from the odometry stream to warp the feature maps of the previous timestep into the current view using a predicted depth map of the image. We utilize DispNet [127] to obtain the depth map D_t for the current image I_t and fuse the warped features with the intermediate network representations of the current timestep as described in Section 4.2.2.4. The yellow and red blocks in Figure 4.2 represent the warping and fusion layers respectively at *Res4f* and *Res5c* that are used to first warp then fuse the feature maps $\mathbf{z}_{t-1}^{4f}$ and $\mathbf{z}_{t-1}^{5c}$ from the previous timestep into the network.

Utilizing the relative pose $\Delta\mathbf{p}_{t-1,t}$, the estimated depth map D_t and the projection function π, we formulate the warping as

$$\hat{u}_r := \pi\left(T\left(\Delta\mathbf{p}_{t-1,t}\right)\pi^{-1}\left(u_r, D_t\left(u_r\right)\right)\right), \tag{4.6}$$

where the warped pixel $\hat{u}_r$ is obtained from pixel u_r using the depth information $D_t(u_r)$ and the relative motion between the images $\Delta\mathbf{p}_{t-1,t}$. The function $T\left(\Delta\mathbf{p}_{t-1,t}\right)$ denotes the homogenous transformation matrix representing $\Delta\mathbf{p}_{t-1,t}$, π denotes the projection function transforming from world to camera coordinates such that $\pi : \mathbb{R}^3 \mapsto \mathbb{R}^2$ and π^{-1} denotes the transformation from camera to world coordinates using a depth map $D_t\left(u_r\right)$.

In order to facilitate the computation of the gradients necessary for back-propagation, we use bilinear interpolation as a sampling mechanism for warping. As the warping method is fully differentiable, our approach does not require any pre-computation for training and runs online. Furthermore, our self-supervised warping procedure adds minimal overhead as we only calculate the warping grid once at the input resolution in terms of pixels u_r and employ average pooling to apply the grid at multiple scales for transforming the feature maps $\mathbf{z}_{t-1}$ to their warped counterparts $\hat{\mathbf{z}}_{t-1}$. In Section 4.4.4.2, we investigate the effect of warping feature maps at different stages of the network on the accuracy of the predicted segmentation mask.

4.2.2.4 Multitask Learning

Our main motivation towards jointly learning to estimate the global pose, ego-motion and semantics of the scene is to enable the inductive transfer of domain specific knowledge across the different task-specific networks while simultaneously exploiting complementary features. Furthermore, it enables the inherent encoding of geometric and semantic knowledge in the global pose regression network during training, resulting in pose predictions that are semantically and geometrically consistent with the scene information. The multitask network framework is thus structured such that the network is interdependent on the intermediate representations and outputs of each of the learned tasks. As shown in Figure 4.2, we employ hard parameter feature sharing between the global pose regression stream and the visual odometry stream receiving the image from the current timestep until the end of the *Res3* block. This has the effect of both exploiting the task-specific similarities among both sub-tasks, and influencing the shared weights from the localization network to incorporate motion-specific features caused by the inductive bias due the relative motion estimation. Furthermore, it effectuates implicit attention on regions of the image that are more informative for relative motion estimation. In Appendix A.3, we evaluate the impact of the number of layers shared between the global pose regression and the odometry stream on the accuracy of the estimated poses.

Combining features from different layers or networks is most commonly performed through tensor concatenation and element-wise addition/multiplication. Such an approach is effective in cases where both tensors contain sufficient relevant information. However, the results are often suboptimal as the resulting tensor tends to accumulate irrelevant feature maps and the effectiveness of the combination becomes highly dependent on the stages at which the fusion is performed. Among the indispensable components of the proposed multitask learning framework is the novel adaptive weighted fusion layer [157]. The proposed fusion layer is comprised of an element-wise weighting of the input tensors based on the region activations, followed by non-linear feature pooling. Pooling across feature space as opposed to spatial pooling is regarded as a coordinate-dependent transformation that yields the same number of filters as the input tensor. Moreover, using region activations to weigh the tensors enables the framework to learn the most favorable weighting while discarding irrelevant information.

We formulate the mathematical representation of the adaptive weighted fusion layer [157] with respect to two input feature maps $\mathbf{z}^a$ and $\mathbf{z}^b$ from layers a and b, where both layers can belong to the same network or different task-specific networks. The following notation, nonetheless, can be extended to multiple maps in a straightforward manner. The output of the fusion layer can be represented as

$$\hat{\mathbf{z}}_{\text{fuse}} = \max\left(\mathbf{W} * \left((\mathbf{w}^a \odot \mathbf{z}^a) \oplus \left(\mathbf{w}^b \odot \mathbf{z}^b\right)\right) + \mathbf{b}, 0\right), \tag{4.7}$$

where $\mathbf{w}^a$ and $\mathbf{w}^b$ are learned weights having the same dimensions as $\mathbf{z}^a$ and $\mathbf{z}^b$, $\mathbf{W}$ and $\mathbf{b}$ are parameters of the non-linear feature pooling, $\odot$ and $\oplus$ denote the per-channel scalar multiplication and concatenation respectively, and $*$ denotes the convolution operation [157]. The above formulation entails the following steps: *(i)* each channel of the input feature maps is weighted using learnable weights, *(ii)* the output weighted maps are linearly combined together, *(iii)* non-linear feature pooling is applied on the resulting tensor. Non-linear feature pooling in this case can be realized using existing layers in the form of a 1×1 convolution followed by a non-linear activation function such as ReLU. The adaptive weighted fusion layer is incorporated at *Res4c* to fuse semantic features into the localization stream, and at the end of *Res3* and *Res4* blocks of the segmentation stream to fuse warped semantic features from the previous timestep into the current timestep. We denote our multitask architecture as VLocNet$++_{\text{MTL}}$ in the remainder of this chapter. In Appendix A.5, we demonstrate the effectiveness of our proposed fusion over simple concatenation for both inter and intra-task fusion.

Jointly learning all tasks is a challenging problem due to the diversity of the tasks which in turn results in varying units and scales for each loss term. Naively combining the task-specific losses through simple addition would result in no substantial benefit for any of the tasks as the task with the highest scale would dominate the training. As a solution to this problem, we use learnable scalar weights $\hat{s}_{loc}, \hat{s}_{vo}, \hat{s}_{seg}$ to balance the scale of each of the global pose regression, odometry and semantic segmentation loss terms respectively. In order to train our multitask framework, we minimize the following loss function:

$$\begin{aligned} \mathcal{L}_{multi} := & \exp(-\hat{s}_{loc})\mathcal{L}_{GC} + \hat{s}_{loc} \\ & + \exp(-\hat{s}_{vo})\mathcal{L}_{vo} + \hat{s}_{vo} \\ & + \exp(-\hat{s}_{seg})\mathcal{L}_{seg} + \hat{s}_{seg}. \end{aligned} \tag{4.8}$$

In Section 4.4.4.3, we investigate the effect of employing different optimization strategies as well as weighting techniques on the accuracy of the learned tasks and demonstrate the utility of our proposed learnable weighting method.

4.3 Datasets and Augmentation

Training supervised deep learning methods necessitates the availability of a large enough training data with corresponding ground-truth annotations. However, as previously stated, acquiring such data is laborious for real-world robotic scenarios. Moreover, the problem becomes more pronounced for multitask learning frameworks which require a large enough training dataset containing individual task-specific labels. Although a number of datasets exist that are commonly used for benchmarking semantic segmentation and visual localization tasks, to the best of our knowledge, there does not exist a large enough dataset containing both semantic and global localization ground-truth labels with multiple loop closures. As a solution to this problem, we introduce the challenging *DeepLoc* dataset which is captured at an outdoor urban environment. The dataset contains RGB-D images tagged with 6-DoF global poses and pixel-level semantic labels. Furthermore, we introduce the *DeepLocCross* dataset captured at a dynamic urban environment containing RGB-D images tagged with 6-DoF poses. Both *DeepLoc* and *DeepLocCross* are made publicly available [157]. In addition to the aforementioned datasets, we evaluate the performance of our localization architecture on the challenging Microsoft 7-Scenes benchmark [175]. Evaluating our multitask architecture on the aforementioned datasets demonstrates the robustness of our approach to scene structure, presence of dynamic objects as well as the capturing medium.

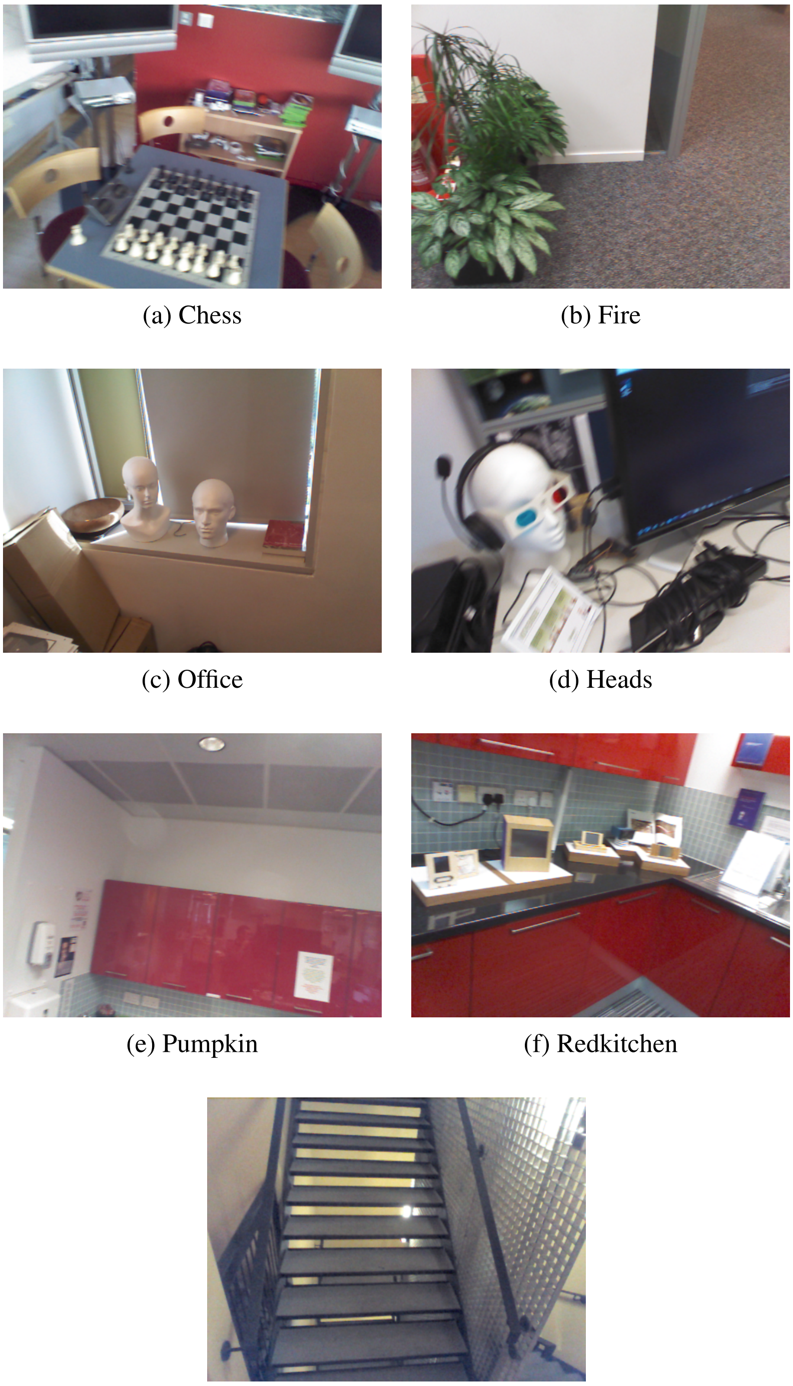
We experimented with augmenting the training images using pose synthesis [200] and synthetic view generation [138], however, employing them did not yield any performance improvement, but rather in some cases negatively impacted the pose accuracy. For learning semantics, we randomly apply image augmentations in the form of rotation, translation, scaling, skewing, cropping, flipping, contrast and brightness modulation.

4.3.1 Microsoft 7-Scenes

The Microsoft 7-Scenes dataset [175] is a commonly employed benchmark for camera relocalization and tracking. It is comprised of RGB-D images captured from seven different scenes in an indoor office environment: *Chess, Fire, Heads, Office, Pumpkin, RedKitchen* and *Stairs*. The images were captured with a handheld Kinect RGB-D camera at a resolution of 640×480 pixels and the ground-truth poses were generated using KinectFusion [175]. Each scene contains multiple sequences recorded in a room with different camera motions. Each of the sequences contains about 500 to $1{,}000$ frames. Figure 4.3 shows challenging images from each of the scenes and illustrates the various difficulties encountered benchmarking on this dataset. The challenges range from motion blur due to the camera movement, as shown in Figure 4.3(a, d), presence of repetitive structures such as in Figure 4.3(g), which increase the difficulty of the estimating the feature correspondences, and the presence of highly reflective surfaces as in Figure 4.3(e, f).

4.3.2 DeepLoc

We introduce a challenging urban outdoor localization dataset collected using our robotic platform presented in Section 2.1. We captured the data around a university campus spanning an area of $110\mathrm{m} \times 130\mathrm{m}$, that the robot traversed multiple times with different



(a) Chess

(b) Fire

(c) Office

(d) Heads

(e) Pumpkin

(f) Redkitchen

(g) Stairs

Figure 4.3: Challenging images from the Microsoft 7-Scenes benchmark, exhibiting significant motion blur (a, d), repetitive structures (g), highly reflective surfaces (e, f) and low-texture regions (b, c).

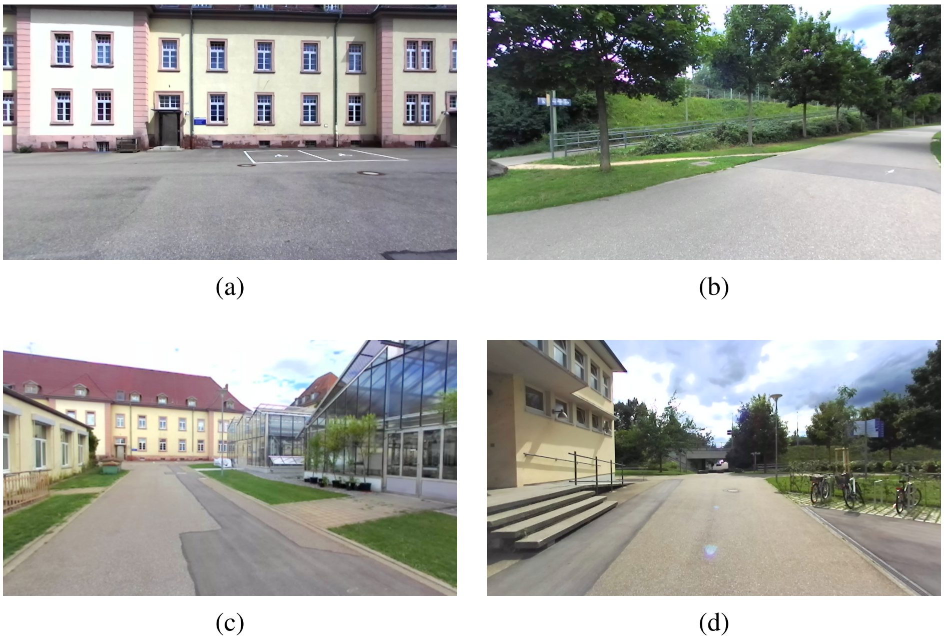


Figure 4.4: Example images from our DeepLoc dataset that show challenging scenarios for global pose regression, visual odometry estimation and semantic segmentation. Several sections of the traversed area contain buildings with repetitive patterns (a) and few distinctive features (b). The traversed trajectory contains multiple structures made solely of glass (c), buildings with large reflective glass surfaces (d), as well as partially occluded structures such as bikes attached to bike-stands (d). Note that the images are artificially brightened to facilitate viewing.

driving patterns such that there is minimum overlap in the trajectories. The dataset contains RGB-D stereo images captured at a resolution of $1{,}280\times720$ pixels at 20Hz. Note that while the dataset contains stereo image pairs, we however use monocular images for localization. We use the LiDAR-based SLAM system from Kümmerle *et al.* [108] to generate the ground-truth pose labels. Furthermore, we provide pixel-level semantic segmentation annotations for ten categories: *Background, Sky, Road, Sidewalk, Grass, Vegetation, Building, Poles & Fences, Dynamic and Other*.

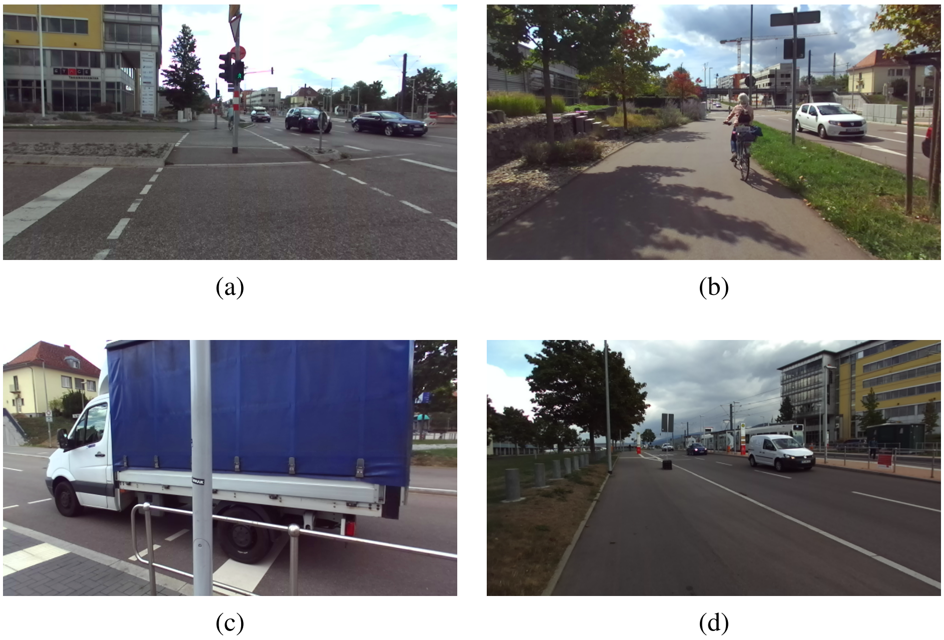
The dataset contains a total of ten sequences: seven of which were used for training with a total of $2{,}737$ images and the remaining three for testing containing $1{,}173$ images. We captured the data at varying times of the day which is reflected in the appearance of the images in terms of lighting conditions, glare, shadows and orange dawn sky. Additionally, a number of images contain significant motion blur caused by the motion of the robotic platform. The environment in which the dataset was collected further increases the difficulty of perception related tasks as it contains buildings with similar facades and repetitive structures (Figure 4.4(a, b), and translucent and reflective glass buildings (Figure 4.4(c, d)). Overall, the dataset can be very challenging for vision-based applications such as 6-DoF pose estimation, camera relocalization, semantic segmentation, ego-motion estimation and loop closure detection. We are convinced that this dataset enables future research in multitask and multimodal learning.

4.3.3 DeepLocCross

In addition to the DeepLoc dataset, we introduce the challenging DeepLocCross dataset which we captured using our robotic platform presented in Section 2.1. We collected the data around a highly dynamic road segment spanning an area of $158\mathrm{m}\times 90\mathrm{m}$ which contains a tram line as well as multiple pedestrian crossings and road intersections. Like the DeepLoc dataset, the DeepLocCross dataset contains RGB-D stereo images captured at $1{,}280\times 720$ pixels at a rate of $20\mathrm{Hz}$. The ground-truth pose labels are generated using the LiDAR-based SLAM system from Kümmerle *et al.* [108]. In addition to the 6-DoF localization poses of the robot, the dataset additionally contains tracked detections of the observable dynamic objects. Each tracked object is identified using a unique track ID, spatial coordinates, velocity and orientation angle. Furthermore, as the dataset contains multiple pedestrian crossings, we provide labels at each intersection indicating its safety for crossing. Note that in this chapter, we only utilize the RGB-D images and localization poses for evaluating our proposed multitask architecture. The dynamic tracking information and intersection safety labels are used in the context of motion prediction and intersection safety prediction in Chapter 5.

The dataset consists of seven training sequences with a total of $2{,}264$ images, and three testing sequences with a total of 930 images. The dynamic nature of the environment in which the dataset was captured, renders the tasks of localization and visual odometry estimation extremely challenging due to the varying weather conditions, presence of shadows and motion blur caused by the movement of the robot platform. Figure 4.5(a, b) show example images from the dataset depicting the presence of pedestrians, cars, trams and cyclists, each of which exhibiting different motion behavior than the ego-motion of the robot thereby rendering the visual odometry estimation rather challenging. Furthermore, the presence of multiple dynamic objects often results in partial and full occlusions of the informative regions of the image (Figure 4.5(c)), and the presence of repeated structures (Figure 4.5(d)) further increase the difficulty of the challenging task of pose estimation. Overall this dataset covers a wide range of perception related tasks such as loop closure detection, semantic segmentation, visual odometry estimation, global localization, scene flow estimation and behavior prediction. We make both the DeepLoc and DeepLocCross datasets publicly available* [157] to facilitate further progress in the field of multitask learning for robotics.

*VLocNet++ live demo and dataset are publicly available at:
`http://deeploc.cs.uni-freiburg.de`



(a)

(b)

(c)

(d)

Figure 4.5: Example images from our DeepLocCross dataset depicting challenging scenarios. The presence of multiple dynamic objects in the scene (a, b) such as cars, bicycles and tram cars render benchmarking on this dataset challenging. Additionally, the images exhibit significant occlusion of stable features by dynamic objects (c) and presence of repetitive structures (d) rendering the localization and visual odometry estimation tasks challenging. Note that the images are artificially brightened to facilitate viewing.

4.4 Experimental Evaluation

In this section, we evaluate our proposed multitask learning framework for the tasks of global pose regression, visual odometry estimation and semantic scene segmentation. We first quantify the performance of each of the single-task models by comparing against deep learning methods for each corresponding task, followed by an extensive evaluation of our multitask framework. Moreover, we compare against the VLocNet architecture with employing our proposed GC loss function in order to gain perspective on the effect of the architecture topology on the localization performance. Furthermore, we present extensive ablation studies and qualitative analysis demonstrating the efficacy of our approach in different scenarios as well as providing insights on the various architectural design choices and the representations learned by the network. Additional ablation studies can be found in Appendix A. In the following, we begin by detailing the training procedure employed.

4.4.1 Network Training

We train our models on all datasets using random crops of the image and test on the center crop. We found that using random crops acts as a better regularization method in comparison to synthetic augmentation techniques. We initialize the residual blocks of each task-specific network using the weights from the ResNet-50 model trained on the ImageNet dataset [64] and the remaining layers with Xavier initialization [74]. We employ the Adam solver [98] for optimization with parameter values $\beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 1^{-10}$. In order to train each multitask network, we follow a multi-stage procedure by initially training each task-specific model individually with a learning rate of $\lambda = 10^{-3}$ and mini-batch size of 32.

Training deep networks with limited labeled data is a challenging problem that is most commonly overcome through transfer learning approaches. However, in order to apply such methods, one must rely on initializing the network with pre-trained weights from a large-enough dataset on a semantically similar task. The approach proposed in this chapter is the first to address such a wide range of tasks, and as such pre-existing trained models cannot be employed to accelerate training. To this end, we evaluate the effect of various weight initializations and bootstrapping methods on the accuracy of our method in Section 4.4.4.3. Furthermore, the optimization strategy employed during training has a direct effect on the learned representations, with the most common strategies employed being alternate and joint training. In alternate training, we utilize a separate optimizer for each task and randomly alternate between executing each optimizer on the task-specific function. This has the advantage of allowing synchronized transfer of information among the tasks, thereby enforcing commonality between them. Alternate optimization strategies, however, have the disadvantage of introducing task-specific bias in the parameters towards the task that was selected first for optimization. On the other hand, joint optimization strategies entail the combination of all task-specific loss functions into a single loss function and using a single optimizer to train all network streams concurrently. While this has the advantage of maintaining the individuality of the tasks during training, weighting mechanisms need to be applied to prohibit the task with the largest scale from dominating the training. For the tasks at hand, we found that a joint optimization strategy with learnable weighting variables is most suitable. We further evaluate the performance gain over alternate optimization approaches in Section 4.4.4.3. Both multitask models of VLocNet and VLocNet++ are trained using an Adam optimizer with parameter values $\beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 1^{-10}$ and a learning rate of $\lambda = 10^{-4}$ for a maximum of $200{,}000$ iterations. We use the Tensorflow library [21] for the implementation and train the network on a single NVIDIA Titan X GPU.

4.4.2 Comparison with the State-of-the-Art

In the following, we show empirical evaluations comparing the performance of each of the single-task models VLocNet$_{\text{STL}}$ and VLocNet$++_{\text{STL}}$ using the proposed GC loss function with deep learning-based approaches for the tasks of global pose regression and visual odometry estimation. Note that we do not present results for the semantic scene segmentation task with the single-task model as that corresponds to the AdapNet architecture [191], rather we show the performance gain from incorporating the self-supervised warping layer to the network in Section 4.4.4.5.

4.4.2.1 Evaluation of Visual Localization

As a primary evaluation criterion, we report the localization performance in comparison to deep learning-based approaches on the Microsoft 7-Scenes, DeepLoc and DeepLocCross datasets. We analyze the performance in terms of the median translational and rotational errors for each scene using the training and test splits provided by the datasets. Table 4.1 shows the results on the Microsoft 7-Scenes dataset. Our proposed VLocNet$_{\text{STL}}$ consistently reduces the localization error for all scenes by an average of 77.14% in translation and 59.14% in rotation in comparison to the best performing model NNet [111]. Furthermore, by employing the adaptive weighted fusion layer to fuse previous pose information into the current network stream, VLocNet++$_{\text{STL}}$ achieves a further improvement of 54.1% and 63.4% in the translational and rotational components, respectively over VLocNet$_{\text{STL}}$. The performance improvements are most apparent in the perceptually hardest scenes that contain textureless and reflective surfaces such as Fire (Figure 4.3(b)), Pumpkin (Figure 4.3(e)) and scenes containing repetitive structures such as Stairs (Figure 4.3(g)).

Table 4.2 shows the results of our proposed single-task architecture on the DeepLoc dataset. On this dataset, our proposed VLocNet$_{\text{STL}}$ architecture outperforms the state-of-the-art deep learning-based methods, achieving a localization accuracy of $0.68\text{m}, 3.43°$. Despite the presence of perceptual aliasing in the scene, by incorporating the previous pose information and utilizing the GC loss, the network is able to learn a geometrically consistent model of the environment. Furthermore, by employing the proposed adaptive weighted fusion layer to incorporate the previous predicted pose, VLocNet$++_{\text{STL}}$ achieves a localization error almost half of that achieved by VLocNet$_{\text{STL}}$. This further demonstrates the utility of employing an adaptive method to fuse information as opposed to naive concatenation. Furthermore, it demonstrates that VLocNet$++_{\text{STL}}$ performs equally well in outdoor environments with significant perceptual aliasing as well as indoor textureless environments.

Table 4.3 shows the median translation and orientation error on the DeepLocCross dataset. VLocNet$_{\text{STL}}$ outperforms the baseline deep learning-based methods achieving a localization accuracy of $0.80\text{m}, 4.51°$. This further validates the impact of utilizing the previous pose and the relative motion information on the accuracy of the pose predictions. Furthermore, by incorporating our proposed fusion layer to adaptively weigh the feature maps from the previous pose, VLocNet$++_{\text{STL}}$ further improves on the localization accuracy by 13.5% in rotation despite the challenges faced in the dataset due to the large number of dynamic objects, occlusions and varying illumination conditions (Figure 4.5). This further corroborates the efficacy of our network in accurately estimating the visual

Table 4.1: Median localization error of VLocNet and VLocNet++ in comparison with existing deep learning models on the Microsoft 7-Scenes dataset [158].

Scene	PoseNet [95]	Bayesian PoseNet [93]	LSTM-Pose [199]	Hourglass-Pose [132]	BranchNet [200]	PoseNet2 [94]	NNnet [111]	VLocNet$_{\text{STL}}$ (Ours)	VLocNet++$_{\text{STL}}$ (Ours)
Chess	0.32m, 8.12°	0.37m, 7.24°	0.24m, 5.77°	0.15m, 6.53°	0.18m, 5.17°	0.13m, 4.48°	0.13m, 6.46°	0.036m, 1.71°	**0.023**m, **1.44**°
Fire	0.47m, 14.4°	0.43m, 13.7°	0.34m, 11.9°	0.27m, 10.84°	0.34m, 8.99°	0.27m, 11.3°	0.26m, 12.72°	0.039m, 5.34°	**0.018**m, **1.39**°
Heads	0.29m, 12.0°	0.31m, 12.0°	0.21m, 13.7°	0.19m, 11.63°	0.20m, 14.15°	0.17m, 13.0°	0.14m, 12.34°	0.046m, 6.64°	**0.016**m, **0.99**°
Office	0.48m, 7.68°	0.48m, 8.04°	0.30m, 8.08°	0.21m, 8.48°	0.30m, 7.05°	0.19m, 5.55°	0.21m, 7.35°	0.039m, 1.95°	**0.024**m, **1.14**°
Pumpkin	0.47m, 8.42°	0.61m, 7.08°	0.33m, 7.00°	0.25m, 7.01°	0.27m, 5.10°	0.26m, 4.75°	0.24m, 6.35°	0.037m, 2.28°	**0.024**m, **1.45**°
RedKitchen	0.59m, 8.64°	0.58m, 7.54°	0.37m, 8.83°	0.27m, 10.15°	0.33m, 7.40°	0.23m, 5.35°	0.24m, 8.03°	0.039m, 2.20°	**0.025**m, **2.27**°
Stairs	0.47m, 13.8°	0.48m, 13.1°	0.40m, 13.7°	0.29m, 12.46°	0.38m, 10.26°	0.35m, 12.4°	0.27m, 11.82°	0.097m, 6.48°	**0.021**m, **1.08**°
Average	0.44m, 10.4°	0.47m, 9.81°	0.31m, 9.85°	0.23m, 9.53°	0.29m, 8.30°	0.23m, 8.12°	0.21m, 9.30°	0.048m, 3.80°	**0.022**m, **1.39**°

Table 4.2: Median localization error for the task of visual localization on the DeepLoc dataset [157].

PoseNet [95]	Bayesian PoseNet [93]	SVS-Pose [138]	VLocNet$_{\text{STL}}$ (Ours)	VLocNet++$_{\text{STL}}$ (Ours)
2.42m, 3.66°	2.24m, 4.31°	1.61m, 3.52°	0.68m, 3.43°	**0.37**m, **1.93**°

Table 4.3: Median localization error for the task of visual localization on the DeepLocCross dataset.

PoseNet [95]	Bayesian PoseNet [93]	SVS-Pose [138]	VLocNet$_{\text{STL}}$ (Ours)	VLocNet++$_{\text{STL}}$ (Ours)
5.40m, 6.65°	3.43m, 5.39°	1.22m, 4.24°	**0.80**m, 4.51°	1.21m, **3.90**°

localization in multiple varying environments ranging from indoor static with distinct features to outdoor dynamic with textureless repetitive features.

4.4.2.2 Evaluation of Visual Odometry

In order to evaluate the performance of our single-task architecture on the task of visual odometry estimation, we report the average translational and rotational errors relative to the sequence length on each of the datasets. Since odometry has the problem of drifting over the distance traveled, reporting the pose errors per sequence length helps in providing an unbiased estimate facilitating the comparison of the various methods. Similar to

Table 4.4: Average translational and rotational error for the task of visual odometry on the Microsoft 7-Scenes dataset $[\%, \text{deg/m}]$ [157].

Scene	LBO [143]	DeepVO [135]	cnnBspp [133]	VLocNet$_{\text{STL}}$ (Ours)	VLocNet++$_{\text{STL}}$ (Ours)
Chess	1.69, 1.13	2.10, 1.15	1.38, 1.12	1.14, 0.75	**0.99**, **0.66**
Fire	3.56, 1.42	5.08, 1.56	2.08, 1.76	1.81, 1.92	**0.99**, **0.78**
Heads	14.43, 2.39	13.91, 2.44	3.89, 2.70	1.82, 2.28	**0.58**, **1.59**
Office	3.12, 1.92	4.49, 1.74	1.98, 1.52	1.71, 1.09	**1.32**, **1.01**
Pumpkin	3.12, 1.60	3.91, 1.61	1.29, 1.62	1.26, 1.11	**1.16**, **0.98**
RedKitchen	3.71, 1.47	3.98, 1.50	1.53, 1.62	1.46, 1.28	**1.26**, 1.52
Stairs	3.64, 2.62	5.99, 1.66	2.34, 1.86	1.28, 1.17	1.55, **1.10**
Average	4.75, 1.79	5.64, 1.67	2.07, 1.74	1.51, 1.45	**1.12**, **1.09**

Table 4.5: Average translational and rotational error on the DeepLoc dataset for the task of visual odometry $[\%, \mathrm{deg/m}]$ [157].

LBO [143]	DeepVO [135]	cnnBspp [133]	VLocNet$_{\mathrm{STL}}$ (Ours)	VLocNet++$_{\mathrm{STL}}$ (Ours)
$0.41, 0.053$	$0.33, 0.052$	$0.35, 0.049$	$0.15, 0.040$	$\mathbf{0.12}, \mathbf{0.024}$

Table 4.6: Average error per sequence length for the task of visual odometry on the DeepLocCross dataset $[\%, \mathrm{deg/m}]$.

LBO [143]	DeepVO [135]	cnnBspp [133]	VLocNet$_{\mathrm{STL}}$ (Ours)	VLocNet++$_{\mathrm{STL}}$ (Ours)
$1.02, 0.032$	$0.99, 0.029$	$0.79, 0.054$	$0.20, 0.033$	$\mathbf{0.18}, 0.038$

the global pose regression task, we use the same train and test splits provided by the datasets for each scene. In Table 4.4, we show the results on the Microsoft 7-Scenes dataset where our VLocNet$_{\mathrm{STL}}$ outperforms the compared methods achieving an error of 1.51%, $1.45\mathrm{deg/m}$ per sequence length. Employing the full pre-activation ResNet as the base of our architecture in VLocNet++$_{\mathrm{STL}}$ further improves the results achieving a translational error of 1.12% and rotational error of $1.09\mathrm{deg/m}$. Similarly on the DeepLoc dataset shown in Table 4.5, despite the textureless environment and varying lighting conditions, both our proposed VLocNet$_{\mathrm{STL}}$ and VLocNet++$_{\mathrm{STL}}$ architectures surpass the accuracy of the compared methods with a translational error of 0.12% and a rotational error of $0.024\mathrm{deg/m}$.

Table 4.6 shows the average translational and rotational error as a function of the sequence length on the DeepLocCross dataset. On this dataset, VLocNet$_{\mathrm{STL}}$ outperforms the compared approaches reducing the error by at least 1.6-times. Moreover, employing VLocNet++$_{\mathrm{STL}}$ further reduces the translational error reaching 0.18%. This improvement over the compared architectures can be attributed to employing the pre-activated ResNet as a base architecture which enables our network to learn more general representations of the environment while being tolerant to noise. The presence of multiple dynamic objects in motion in the DeepLocCross dataset (see Figure 4.5) renders the visual odometry estimation task quite challenging due to the presence of significant motion parallax. Benchmarking on this dataset validates the suitability as well as efficacy of our proposed architecture to be deployed in highly dynamic environments.

4.4.3 Evaluation of the Multitask Learning

Following the evaluation of our single-task architectures, in this section we present quantitative evaluations of our proposed multitask learning framework for the global pose regression, visual odometry estimation and semantic scene segmentation tasks. The goal of this experiment is to investigate the contribution of jointly learning the aforementioned tasks in the same settings to the overall performance. We compare the performance of both VLocNet$_{\text{MTL}}$ and VLocNet$++_{\text{MTL}}$ with state-of-the-art methods on each of the tasks on the Microsoft 7-Scenes, DeepLoc and DeepLocCross datasets. On the Microsoft 7-Scenes dataset, we provide empirical results for only the visual localization and odometry estimation tasks as no semantic labels are provided for this dataset. We further provide a qualitative evaluation of the generalization capabilities of our proposed self-supervised warping scheme on the DeepLocCross dataset in Section 4.4.4.5.

4.4.3.1 Evaluation of Visual Localization

We follow the evaluation procedure of the single-task models by utilizing the same training and test splits provided by each dataset, and reporting the median translational and rotational error for each scene. We benchmark the performance of our single-task architectures VLocNet$_{\text{STL}}$ and VLocNet$++_{\text{STL}}$ as well as the multitask variants VLocNet$++_{\text{MTL}}$, VLocNet$_{\text{MTL}}$ on the Microsoft 7-Scenes dataset by comparing against both local feature-based pipelines and learning-based techniques. Table 4.7 shows the median localization pose error. The results show that our architecture VLocNet$_{\text{MTL}}$ is the first deep learning-based approach to perform comparably with local feature-based learning methods achieving a median localization error of 0.04m and $3.09°$. By jointly learning to regress the relative motion in addition to the global pose, the network is able to efficiently incorporate motion-specific features that are necessary for accurate pose predictions. This in turn enables our network to achieve sub-centimeter and sub-degree accuracy for the majority of the scenes. Furthermore, incorporating the proposed adaptive weighted fusion for the previous pose further reduces the localization error by approximately 50% in comparison to VLocNet$_{\text{MTL}}$. Moreover, unlike local feature-based approaches, our proposed VLocNet$++_{\text{MTL}}$ is able to accurately estimate the global pose in environments containing repetitive and textureless structures.

In Figure 4.6, we present the median localization error metric and percentage of poses for which the error is below 5cm and $5°$. While VLocNet$_{\text{MTL}}$ is able to outperform SCoRe Forests [175] in terms of the number of images with pose error below 5cm and $5°$, it is still outperformed by DSAC2 [41]. However, by employing VLocNet++ as the base architecture and utilizing the proposed GC loss function, VLocNet$++_{\text{STL}}$ is able to achieve a localization accuracy of 96.4%. This amounts to an improvement of 20.3% over the previous state-of-the-art approach DSAC2 [41], and an order of magnitude compared to other deep learning methods [94, 111]. Furthermore, employing the proposed

Table 4.7: Benchmarking the median localization error of VLocNet and VLocNet++ on the Microsoft 7-Scenes dataset [158].

Scene	Active Search [169]	SCoRe Forest [175]	DSAC [42]	DSAC2 [41] w/o 3D	DSAC2 [41] w/ 3D	VLocNet$_{\text{STL}}$ (Ours)	VLocNet++$_{\text{STL}}$ (Ours)	VLocNet$_{\text{MTL}}$ (Ours)	VLocNet++$_{\text{MTL}}$ (Ours)
Chess	0.04m, 1.96°	0.03m, 0.66°	0.02m, 1.20°	0.02m, 0.70°	0.02m, 0.50°	0.03m, 1.71°	0.023m, 1.44°	0.03m, 1.69°	**0.018**m, 1.17°
Fire	0.03m, 1.53°	0.05m, 1.50°	0.04m, 1.50°	0.04m, 1.20°	0.02m, 0.80°	0.04m, 5.34°	0.018m, 1.39°	0.04m, 4.86°	**0.009**m, **0.61**°
Heads	0.02m, 1.45°	0.06m, 5.50°	0.03m, 2.70°	0.24m, 10.00°	0.01m, 0.80°	0.04m, 6.64°	0.016m, 0.99°	0.05m, 4.99°	**0.008**m, **0.60**°
Office	0.09m, 3.61°	0.04m, 0.78°	0.04m, 1.60°	0.03m, 0.80°	0.03m, 0.70°	0.04m, 1.95°	0.024m, 1.14°	0.03m, 1.51°	**0.016**m, 0.78°
Pumpkin	0.08m, 3.10°	0.04m, 0.68°	0.05m, 2.00°	0.04m, 1.10°	0.04m, 1.00°	0.04m, 2.28°	0.024m, 1.45°	0.04m, 1.92°	**0.009**m, 0.82°
RedKitchen	0.07m, 3.37°	0.04m, 0.76°	0.05m, 2.00°	0.05m, 1.30°	0.04m, 1.00°	0.04m, 2.20°	0.025m, 2.27°	0.03m, 1.72°	**0.017**m, 0.93°
Stairs	0.03m, 2.22°	0.32m, 1.32°	1.17m, 33.1°	0.27m, 5.40°	0.10m, 2.50°	0.10m, 6.48°	0.021m, 1.08°	0.07m, 4.96°	**0.010**m, **0.48**°
Average	0.05m, 2.46°	0.08m, 1.60°	0.20m, 6.30°	0.099m, 2.92°	0.04m, 1.04°	0.048m, 3.80°	0.022m, 1.39°	0.04m, 3.09°	**0.013**m, **0.77**°

Table 4.8: Median localization error of our multitask framework for the task of visual localization on the DeepLoc dataset [158].

PoseNet [95]	Bayesian PoseNet [93]	SVS-Pose [138]	VLocNet$_{\text{STL}}$ (Ours)	VLocNet++$_{\text{STL}}$ (Ours)	VLocNet$_{\text{MTL}}$ (Ours)	VLocNet++$_{\text{MTL}}$ (Ours)
2.42m, 3.66°	2.24m, 4.31°	1.61m, 3.52°	0.68m, 3.43°	0.37m, 1.93°	0.47m, 2.38°	**0.32**m, **1.48**°

Table 4.9: Median localization error for the task of visual localization of VLocNet$_{\text{MTL}}$ and VLocNet++$_{\text{MTL}}$ on the DeepLocCross dataset.

PoseNet [95]	Bayesian PoseNet [93]	SVS-Pose [138]	VLocNet$_{\text{STL}}$ (Ours)	VLocNet++$_{\text{STL}}$ (Ours)	VLocNet$_{\text{MTL}}$ (Ours)	VLocNet++$_{\text{MTL}}$ (Ours)
5.40m, 6.65°	3.43m, 5.39°	1.22m, 4.24°	0.80m, 4.51°	1.21m, 3.90°	**0.63**m, 4.75°	1.69m, **1.85**°

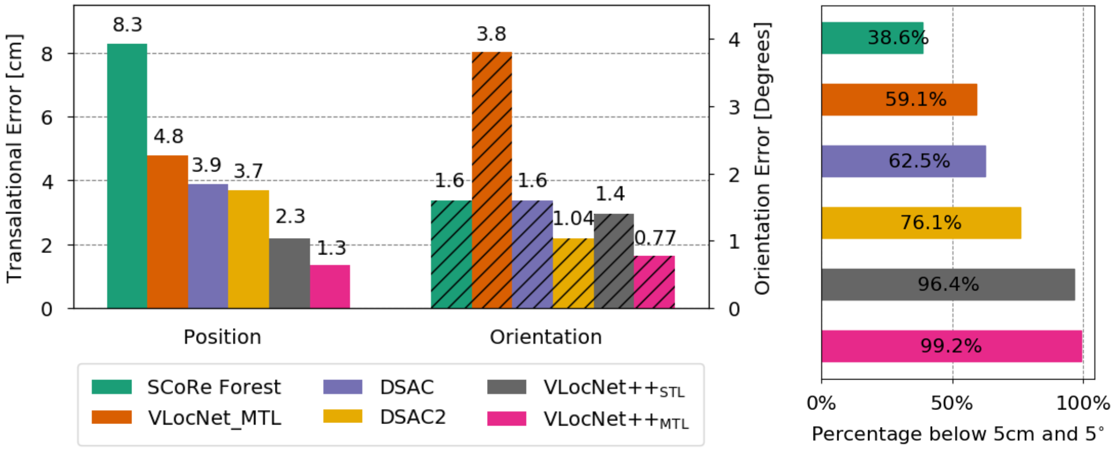


Figure 4.6: Benchmarking the visual localization performance on the Microsoft 7-Scenes dataset in terms of median localization errors (left) and percentage of test images with a pose error below 5cm and 5° (right) [157]. We compare against state-of-the-art methods utilizing the 3D model, depth data and RGB images. Our proposed approach uses only RGB images as input.

multitask learning framework VLocNet$++_{\text{MTL}}$ results in a further improvement in the localization accuracy, thus setting the new state of the art on this benchmark, at the time of writing this thesis, with an accuracy of 99.2%. Note that aside from VLocNet and VLocNet++, the methods shown in Figure 4.6 require a 3D model of the scene through RGB-D data, whereas both VLocNet and VLocNet++ require only monocular images as input. Furthermore, note that both variants of DSAC [42] and DSAC2 [41] that utilize only RGB images demonstrate a lower performance as shown in Table 4.7. The performance achieved by VLocNet++ demonstrates the efficacy of utilizing the proposed GC loss function in combination with the adaptive weighted fusion layer in enabling the network to leverage the motion-specific features, thereby learning a geometrically and temporally consistent motion model of the scene.

Table 4.8 shows the median localization error on the DeepLoc dataset. While our single-task architecture VLocNet$_{\text{STL}}$ outperforms the compared approaches, the results show that jointly learning the ego-motion as an auxiliary task in VLocNet$_{\text{MTL}}$ improves the localization accuracy by 30.9% in translation and 30.6% in rotation in comparison to the single-task variant. Furthermore, employing our proposed adaptive weighted fusion layer to incorporate the previous pose information, as well as semantic knowledge of the environment, results in further reduction of the localization error achieving 0.32m and 1.48°. In Table 4.9, we present the median localization error on the DeepLocCross dataset. Jointly learning the localization and ego-motion estimation tasks improves upon the single-task VLocNet$_{\text{STL}}$ by 21.2% in the translational component of the pose. This improvement, however, comes at the cost of the rotational accuracy which decreases by

5.3%. Utilizing our adaptive weighted fusion layer to dynamically weigh the feature maps from the previous pose in VLocNet++$_{\mathrm{STL}}$ is able to mitigate this imbalance and improve the rotational pose accuracy by 13.5%. Furthermore, by jointly learning both tasks in VLocNet++$_{\mathrm{MTL}}$, the rotational pose accuracy improves by 52.6% over the single-task variant, thus achieving an overall localization accuracy of $1.69\mathrm{m},\ 1.85°$. This corroborates the synergy between the global localization and ego-motion estimation tasks especially in dynamic environments. Through learning to estimate the visual odometry task, the pose regression network is able to better estimate the location as the attention of the network is drawn to the more informative parts of the image, thus enabling it to better leverage the temporal motion features for a more accurate localization estimate.

4.4.3.2 Evaluation of Visual Odometry

Following the evaluation procedure from the single-task architecture, we report the average translation and orientation error as a function of the trajectory length for each scene. Table 4.10 shows the visual odometry pose error on the Microsoft 7-Scenes dataset. We observe that jointly learning the global location in VLocNet$_{\mathrm{MTL}}$ further improves the ego-motion estimation achieving an error of $1.46\%,\ 1.31\mathrm{deg/m}$ in translation and rotation respectively. This validates our hypothesis that employing parameter sharing between the global pose stream and the visual odometry stream results in mutual benefit for both tasks, specifically in scenes with significant motion blur and low textures such as Chess and Fire. Furthermore, by utilizing the pre-activated ResNet as a base architecture and utilizing the adaptive weighted fusion layer to incorporate previous poses into the pose regression network, VLocNet++$_{\mathrm{MTL}}$ further outperforms VLocNet$_{\mathrm{MTL}}$ with an average error of 1.08% and $1.03\mathrm{deg/m}$ in translation and orientation respectively. We present the results on the DeepLoc dataset in Table 4.11. We observe that employing the multitask version of VLocNet does not result in significant performance improvements on this dataset, which can be attributed to the difficult nature of this dataset caused by the presence of multiple repetitive structures and low texture regions. Nonetheless, VLocNet++$_{\mathrm{MTL}}$ is able to outperform all of the compared methods as well as improve upon the performance of its task-specific variant, achieving a final error of $0.10\%,\ 0.002\mathrm{deg/m}$ in translation and rotation. Moreover, Table 4.12 shows the average odometry pose error on the DeepLocCross dataset. Similar to the DeepLoc dataset, while VLocNet$_{\mathrm{MTL}}$ performs comparatively similar to its single-task variant, VLocNet++$_{\mathrm{MTL}}$ is able to not only learn a balance between the translational and rotational components of the motion, but also learns temporally invariant feature maps that are tolerant to high dynamic noise in the environment caused by the presence of multiple objects in motion. Furthermore, by employing the adaptive weighted fusion layer to incorporate the previous pose information into the global pose regression network, the network is able to better capture the temporal information which in turn improves the ego-motion estimation through the parameter

Table 4.10: Average translational and rotational error for the task of visual odometry on the Microsoft 7-Scenes dataset $[\%, \mathrm{deg/m}]$ [158].

Scene	LBO [143]	DeepVO [135]	cnnBspp [133]	VLocNet$_{\mathrm{STL}}$ (Ours)	VLocNet++$_{\mathrm{STL}}$ (Ours)	VLocNet$_{\mathrm{MTL}}$ (Ours)	VLocNet++$_{\mathrm{MTL}}$ (Ours)
Chess	1.69, 1.13	2.10, 1.15	1.38, 1.12	1.14, 0.75	0.99, 0.66	1.09, 0.73	**0.97**, **0.63**
Fire	3.56, 1.42	5.08, 1.56	2.08, 1.76	1.81, 1.92	0.99, 0.78	1.77, 1.89	**0.98**, **0.73**
Heads	14.43, 2.39	13.91, 2.44	3.89, 2.70	1.82, 2.28	0.58, 1.59	1.75, 2.01	**0.54**, **1.50**
Office	3.12, 1.92	4.49, 1.74	1.98, 1.52	1.71, 1.09	1.32, 1.01	1.68, 0.99	**1.31**, **0.97**
Pumpkin	3.12, 1.60	3.91, 1.61	1.29, 1.62	1.26, 1.11	1.16, 0.98	1.25, 1.08	**1.11**, **0.94**
RedKitchen	3.71, 1.47	3.98, 1.50	1.53, 1.62	1.46, 1.28	1.26, 1.52	1.43, 1.29	**1.24**, 1.40
Stairs	3.64, 2.62	5.99, 1.66	2.34, 1.86	1.28, 1.17	1.55, 1.10	1.26, 1.19	1.40, **1.05**
Average	4.75, 1.79	5.64, 1.67	2.07, 1.74	1.51, 1.45	1.12, 1.09	1.46, 1.31	**1.08**, **1.03**

Table 4.11: Average translational and rotational error on the DeepLoc dataset for the task of visual odometry of VLocNet$_{\mathrm{MTL}}$ and VLocNet++$_{\mathrm{MTL}}$ $[\%, \mathrm{deg/m}]$ [158].

LBO [143]	DeepVO [135]	cnnBspp [133]	VLocNet$_{\mathrm{STL}}$ (Ours)	VLocNet++$_{\mathrm{STL}}$ (Ours)	VLocNet$_{\mathrm{MTL}}$ (Ours)	VLocNet++$_{\mathrm{MTL}}$ (Ours)
0.41, 0.053	0.33, 0.052	0.35, 0.049	0.15, 0.040	0.12, 0.024	0.15, 0.039	**0.10**, **0.020**

Table 4.12: Average error normalized by sequence length for the task of visual odometry of our multitask network on the DeepLocCross dataset $[\%, \mathrm{deg/m}]$.

LBO [143]	DeepVO [135]	cnnBspp [133]	VLocNet$_{\mathrm{STL}}$ (Ours)	VLocNet++$_{\mathrm{STL}}$ (Ours)	VLocNet$_{\mathrm{MTL}}$ (Ours)	VLocNet++$_{\mathrm{MTL}}$ (Ours)
1.02, 0.032	0.99, 0.029	0.79, 0.054	0.20, 0.032	0.18, 0.038	0.21, 0.031	**0.18**, **0.031**

Table 4.13: Comparison of semantic segmentation performance in terms of IoU score with state-of-the-art approaches on our DeepLoc dataset [157].

Approach	Sky	Road	Sidew.	Grass	Veg.	Build.	Poles	Dyn.	Other	Mean
	🟦	⬛	🟨	🟩	🟩	🟫	🟥	⬛	🟨	IoU
FCN-8s [121]	94.65	98.98	64.97	82.14	84.47	87.68	45.78	66.39	47.27	69.53
SegNet [28]	93.42	98.57	54.43	78.79	81.63	84.38	18.37	51.57	33.29	66.05
UpNet [145]	95.07	98.05	63.34	81.56	84.79	88.22	31.75	68.32	45.21	72.92
ParseNet [119]	92.85	98.94	62.87	81.61	82.74	86.28	27.35	65.44	45.12	71.47
DeepLab v2 [53]	93.39	98.66	76.81	84.64	88.54	93.07	20.72	66.84	52.70	67.54
DeepLab v3 [54]	93.51	98.80	77.63	85.78	88.62	93.56	24.66	67.75	53.86	76.02
AdapNet [191]	94.65	98.98	64.97	82.14	84.48	87.68	45.78	66.40	47.27	78.59
VLocNet++$_{\text{MTL}}$ (Ours)	**95.84**	**98.99**	**80.85**	**88.15**	**91.28**	**94.72**	**45.79**	**69.83**	**58.59**	**80.44**

sharing scheme employed.

4.4.3.3 Evaluation of Semantic Segmentation

In this section, we evaluate the performance of VLocNet++$_{\text{MTL}}$ for semantic scene segmentation on the DeepLoc dataset. We benchmark our performance against several state-of-the-art deep learning-based methods: FCN-8s [121], SegNet [28], UpNet [145], ParseNet [119], DeepLab v2 [53], DeepLab v3 [54] and AdapNet [191]. We use the Jaccard index, also known as Intersection over Union (IoU) as the metric for evaluating the performance of the models. Table 4.13 shows the individual category IoU as well as the mean IoU for our network in comparison to state-of-the-art methods. VLocNet++ achieves a mean IoU of 80.44%, consistently outperforming the baselines in all categories. This improvement can be attributed to both the self-supervised warping as well the inductive transfer that occurs from the training signals of the localization network, as the AdapNet model, which we build upon, achieves a lower performance without our proposed improvements. In addition, this enables the model to converge in about $26{,}000$ iterations, whereas AdapNet requires $120{,}000$ iterations to converge.

Analyzing the individual class IoU shows that the largest improvement is in the class Sidewalk with 15.9% over the AdapNet baseline. The ability of the network to accurately distinguish narrow structures such as sidewalks further corroborates the significance of incorporating the self-supervised warping into our architecture. Similarly, both the Grass and Vegetation classes receive a significant improvement over the AdapNet baseline which can be attributed to fusing learned representations from the localization network using our adaptive weighted fusion layer, thereby enabling our model to learn a more accurate disambiguation between the different categories. Employing the self-supervised warping procedure enables the network to accurately distinguish the Dynamic objects within the scene image despite their shape irregularity. In Section 4.4.4.5, we present an extended qualitative analysis on the segmentation masks produced by the network.

4.4.4 Ablation Study

In this section, we present an extensive ablation study investigating the different design choices as well as providing qualitative and generalization analysis of the representations learned by the network. We begin with an evaluation of the proposed loss function, followed by in-depth evaluations of the initialization and optimization strategies. Finally, we conclude with a qualitative evaluation of the representations learned by the network and the generalization capabilities of the network to new data. Additional experiments detailing the various multitask learning design choices of VLocNet++$_{\text{MTL}}$ are presented in Appendix A.

4.4.4.1 Evaluation of the Loss Function

In the following, we investigate the effect of the loss function on the visual localization accuracy. We employ the Pre-activation ResNet-50 as the base architecture, and show the improvements for the following variants:

- M1: Pre-activation ResNet-50 base architecture with ReLUs, Euclidean loss for translation and rotation with $\beta = 1$
- M2: Pre-activation ResNet-50 base architecture with ELUs, Euclidean loss for translation and rotation with $\beta = 1$
- M3: Pre-activation ResNet-50 base architecture with ELUs and previous pose fusion using $\mathcal{L}_{GC}$ loss with $\beta = 1$
- M4: Pre-activation ResNet-50 base architecture with ELUs and previous pose fusion using $\mathcal{L}_{GC}$ loss with $\hat{s}_x$, $\hat{s}_q$, which corresponds to our single-task VLocNet++$_{\text{STL}}$ architecture.

Table 4.14 shows the median error in global pose estimation of the aforementioned variants on the DeepLoc dataset. Employing ELU as an activation function results in an improvement of 12.3% in the rotational pose component. This, however, comes at the cost of the translational component of the pose whose accuracy reduces by a factor of two. Replacing the standard Euclidean loss function with our proposed GC loss enables the M3 model to improve the translational accuracy while further improving the rotational accuracy. Finally the most notable improvement is achieved by replacing the weighting parameter from a constant value β to learnable parameters $\hat{s}_x$, $\hat{s}_q$. Our final

Table 4.14: Comparative analysis of the loss function and weighting parameters on the median localization error of VLocNet++$_{\text{STL}}$ on the DeepLoc dataset [193].

Model	Activation	Loss Function	Weighting Parameter	Median Error
M1	ReLU	$\mathcal{L}_{Euc}$	β	0.57m, 2.44°
M2	ELU	$\mathcal{L}_{Euc}$	β	1.71m, 2.14°
M3	ELU	$\mathcal{L}_{GC}$	β	0.56m, 2.06°
M4 (VLocNet++$_{\text{STL}}$)	ELU	$\mathcal{L}_{GC}$	$\hat{s}_x$, $\hat{s}_q$	$\mathbf{0.37}$m, $\mathbf{1.93}$°

M4 (VLocNet++$_{\text{STL}}$) model achieves a localization error reduction of 35.1%, 20.9% in the translational and rotational pose components, respectively in comparison to the M1 model. This validates our hypothesis that utilizing the GC loss function to constrain the search space and incorporating the relative pose information during training, enables the network to learn a model that is more representative of the environment and thus provides more accurate estimates. Furthermore, by utilizing learnable parameters for weighting the translational and rotational components of the pose, the network learns a more favorable weighting without increasing the number of tunable hyperparameters.

4.4.4.2 Evaluation of the Self-Supervised Warping

In the following, we investigate the effect of the self-supervised warping method on the accuracy of the semantic segmentation task. We conduct experiments to determine the stage at which the warping is most effective. There exist several aspects to this problem which we examine. The first is whether warping should be applied at the end of a residual block or at the beginning. We hypothesize that warping the feature maps at the end of a residual block before the next downsampling stage would be more effective and beneficial when compared to warping at the beginning immediately after the downsampling. The second aspect we wish to investigate is the impact of introducing the warping procedure at multiple downsampling stages. Our hypothesis is that warping the feature maps at multiple stages enables the network to better generalize to the different scales of the objects present in the scene.

Table 4.15 shows the mean IoU achieved by different warping positions in VLocNet++$_{\text{MTL}}$ on the DeepLoc dataset. In order to determine whether the warping should be conducted at the beginning or end of a residual block, we experiment with adding the self-supervised warping layer at both *Res3a* and *Res3d* layers. The results validate our hypothesis that performing the warping at the end of the residual block is more beneficial with an improvement of 0.13% in the mean IoU over warping at the beginning of the block. Following this, we experiment with including multiple warping

Table 4.15: Improvement in the semantic segmentation performance due to warping feature maps from the previous timestep. The warping layer denotes where the warping is performed in the segmentation stream. The results are shown for the DeepLoc dataset [158].

Warping Layer	mIoU
No warping	78.59%
Res3a	80.03%
Res3d	80.19%
Res2c, Res3d	80.09%
Res3d, Res5c	80.34%
Res4f, Res5c	**80.44%**
Res3d, Res4f, Res5c	80.31%

layers at the end of different residual blocks: (*Res2c, Res3d*), (*Res3d, Res5c*), (*Res4f, Res5c*) and (*Res3d, Res4f, Res5c*). We observe that warping feature maps at later stages results in more improvement of the mean IoU. This can be attributed to the fact that the features learned at earlier stages of the network are more abstract in comparison to the end, thus warping at (*Res2c, Res3d*) results in a lower mean IoU than warping at (*Res3d, Res5c*). The highest mean IoU is achieved by warping at (*Res4f, Res5c*) with an improvement of 1.85% over the base AdapNet model.

4.4.4.3 Evaluation of Initialization and Optimization Strategies

In order to evaluate the effect of the optimization strategy and different initializations on the accuracy of the trained model, in this section we conduct experimental evaluations targeting each of the aforementioned design choices. We measure the performance of each choice by evaluating the visual localization task, while we consider the remaining tasks as auxiliary. As described in Section 4.4.1, the training procedure can follow either a joint or alternate optimization scheme. In order to determine the most suitable optimization procedure, we explore using either scheme to minimize the loss function. Furthermore, for joint optimization, we experiment with the weighting parameters employed for balancing the scales between the different loss terms. More precisely, we compare the performance of utilizing constant equal weighting terms for each of the loss terms against learnable weighting parameters. Table 4.16 shows the median localization error of the various optimization strategies on the Microsoft 7-Scenes dataset. The results show that using an alternate optimization strategy, the average localization error is 28.8% and 18.5% lower in translation and rotation, respectively, when compared to the joint optimization with constant equal weight terms. The lower performance of the joint optimization scheme

Table 4.16: Comparative analysis of the optimization strategy on the median localization error of VLocNet++$_{\text{MTL}}$ on the Microsoft 7-Scenes dataset [193].

Optimization Strategy	Median Translational Error	Median Rotational Error
Alternate	0.042m	$3.09°$
Joint Constant Weighting	0.059m	$3.79°$
Joint Learnable Weighting	$\mathbf{0.022}\text{m}$	$\mathbf{1.39}°$

with equal weights can be attributed to the difference in scales of the loss values for each task, which in turn results in the optimization procedure becoming more biased towards minimizing the global pose regression error at the cost of having suboptimal relative pose estimates. This, however, results in worse accuracy for both tasks. Employing the learnable weights in the joint optimization strategy achieves the best localization accuracy with an improvement of 47.6% in the translational components and 55.0% in the rotational components of the pose over the alternate optimization strategy. Using learnable weighting parameters enables the network to maintain the individuality of each task during training while prohibiting the task with the largest scale from dominating the training.

The first training phase of our multitask learning framework includes training each of the task-specific sub-networks separately. During this phase, each network alters the weights of its convolutional layers in a manner that best minimizes the loss function. In the later phases, in order to effectively combine each of the task-specific sub-networks, we require an initialization strategy that enables efficient feature sharing. Towards this goal, we evaluate various initialization strategies using the single-task global pose sub-network VLocNet$_{\text{STL}}$ as a baseline. We report the effect of the various initializations of the joint model on the localization accuracy. More precisely, we compare the effect of the following variants:

- **MTL-GLoc**: initializing the global pose regression stream from the task-specific VLocNet$_{\text{STL}}$ and the remaining layers with Xavier initialization.
- **MTL-VO**: initializing the visual odometry stream using the task-specific weights from VLocNet$_{\text{STL}}$ and using the Xavier initialization for the remaining layers.
- **MTL-Dual**: utilizing the combined weights from each task-specific network to initialize the overall model.

Figure 4.7 shows the median localization error of the aforementioned initialization strategies for VLocNet$_{\text{MTL}}$ on the Microsoft 7-Scenes dataset. Jointly learning both

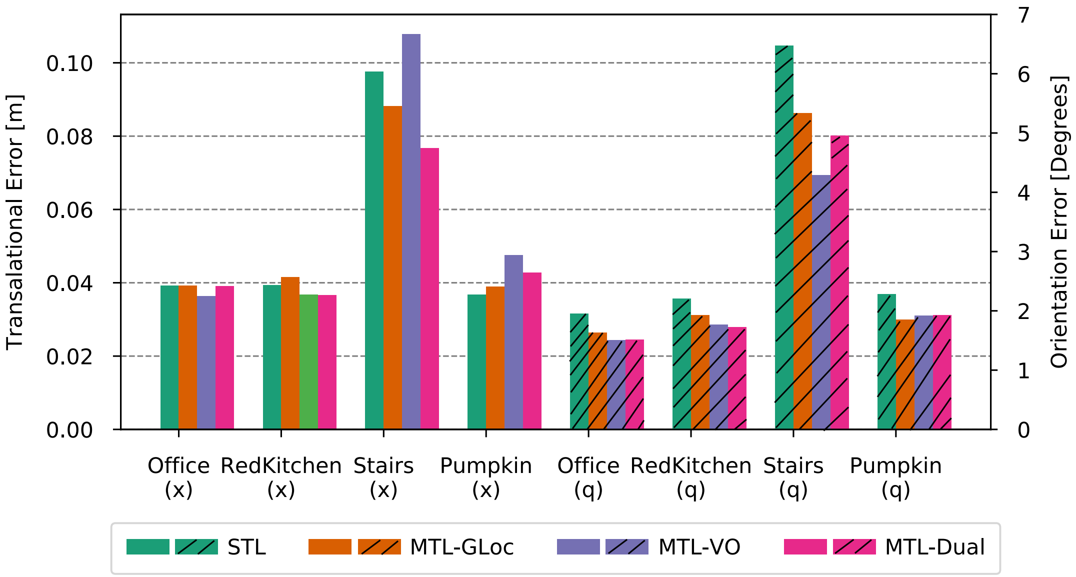
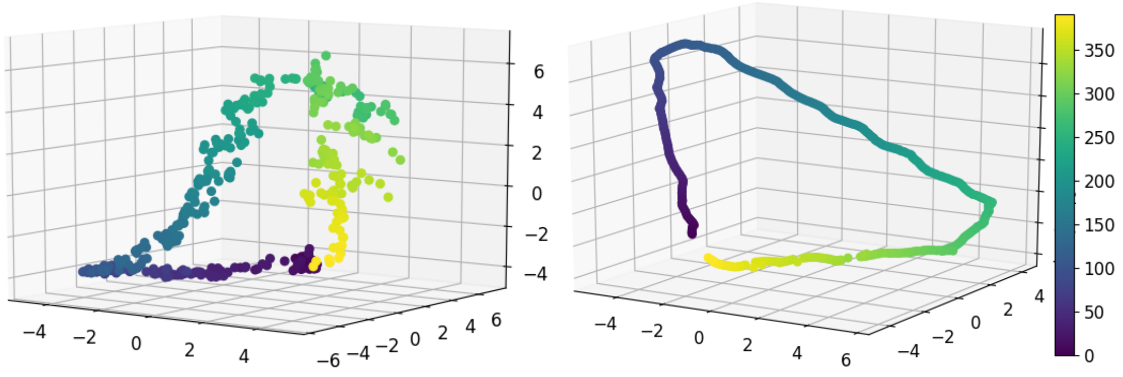


Figure 4.7: Performance of our single-task model in comparison to the multitask model with different weight initializations, on the Microsoft 7-Scenes dataset. The terms (x) and (q) denote the translation and orientation components [193].

localization and odometry tasks results in a higher localization accuracy in comparison to the task-specific sub-network, thereby validating the efficacy of employing a joint learning procedure. Examining the results shows that the best performance is achieved by the dual initialization procedure, as opposed to initializing only one of the sub-networks and learning the other from scratch. This effect can be attributed to the limited amount of data available for training. Furthermore, we observe that initializing only the visual localization stream (MTL-GLoc) yields the lowest improvement in pose accuracy compared to the single-task model. When utilizing this initialization strategy, the visual odometry stream needs to be trained from scratch. During the early training epochs, the predicted relative motion by the network is often inaccurate. However, as the localization sub-network relies on the predicted relative motion information concurrently during the joint training, the incorrect predictions at the early stages inadvertently result in worse performance for the localization as the network can no longer benefit from the motion-specific features.

4.4.4.4 Evaluations of the Learned Representations

The high dimensionality of the representations learned by deep learning approaches deters the capability to fully understand the behavior and choices made by the network. In an attempt to counteract this issue, we employ various approaches that facilitate



(a) PoseNet [95]

(b) VLocNet++$_{\text{MTL}}$

Figure 4.8: Plot of the features learned by PoseNet [95] and VLocNet++$_{\text{MTL}}$ on the DeepLoc dataset using 3D multi-dimensional scaling (MDS) [105]. Inputs are images from the testing seq-01 loop and the points shown are chronologically colored. Features learned by VLocNet++$_{\text{MTL}}$ show precise correlation with the trajectory (Figure 4.10(h)), whereas PoseNet fails to capture the distribution especially for the poses near the glass buildings [157].

the evaluation of the learned representations, thereby enabling us to better interpret the achieved results. Feature visualization and dimensionality reduction techniques are most commonly employed with the goal of facilitating the evaluation of the learned representations. Such techniques transform the data from high dimensional spaces to ones of lower dimensions by decomposing the data along a set of principal axes. For the task of localization, preserving the global geometry of the features is of higher importance over finding clusters and sub-clusters in the data. Therefore, we apply 3D metric Multi-Dimensional Scaling (MDS) [105] to the features learned by the penultimate layer of our VLocNet++$_{\text{MTL}}$ to visualize the underlying distribution.

Figure 4.8 shows the down-projected features after applying MDS to the features learned by VLocNet++$_{\text{MTL}}$ on the DeepLoc dataset in comparison to the features learned by PoseNet [95]. Inspecting the results shows a direct correspondence between the features learned by VLocNet++$_{\text{MTL}}$ and the ground-truth trajectory in Figure 4.10(h). On the other hand, the features learned by PoseNet fail to capture the ground-truth pose distribution in several areas of the trajectory. In the upcoming section, we provide further qualitative results depicting the accuracy of the poses learned by our model.

We further investigate the effect of encoding semantic feature maps into the localization sub-network by visualizing the activation maps of the network for both the single-task and multitask variants of VLocNet++ using using Grad-CAM++ [52]. Grad-CAM++ employs a weighted combination of the positive partial derivatives of the feature maps at the penultimate layer of the network to produce an activation map that acts as a visual explanation of the network predictions. In Figure 4.9, we depict two sample images from

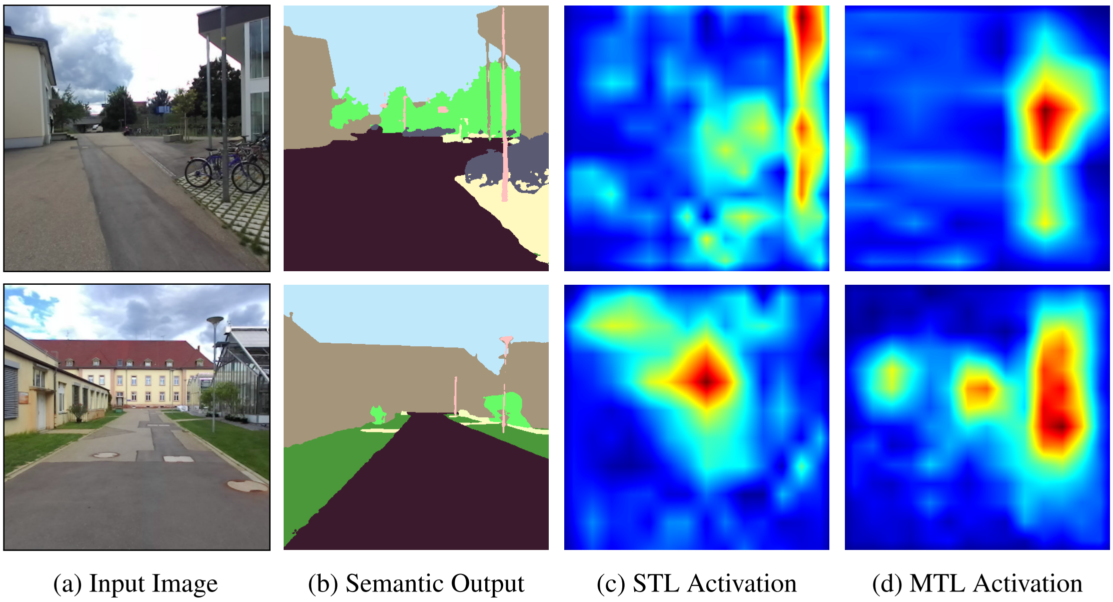


Figure 4.9: Qualitative analysis of the predicted segmentation output along with a visualization of the regression activation maps [52] for both the single-task (STL), and multi-task (MTL) variant of VLocNet++ on the DeepLoc dataset [157].

the DeepLoc dataset that contain glass facades and optical glare. For each of the images, we further depict the segmentation output and regression activation masks produced by Grad-CAM++ for both VLocNet++$_{\mathrm{STL}}$ and VLocNet++$_{\mathrm{MTL}}$. Despite the challenging nature of both images, our model is able to segment both scenes with high granularity. The activation maps generated from VLocNet++$_{\mathrm{MTL}}$ show less noisy activations when compared to the masks from VLocNet++$_{\mathrm{STL}}$. Further examination of the activation masks produced by VLocNet++$_{\mathrm{MTL}}$ shows that the network places more attention on static distinguishable features of the scene that can facilitate the pose regression task such as the pole in the top image, and the glass building in the bottom.

4.4.4.5 Qualitative Evaluation

Our experiments have thus far demonstrated the capability of VLocNet++ in terms of the performance metrics. In this section, however, we present a qualitative analysis of our proposed architecture for the various tasks, in addition to the run-time capabilities of our method. While achieving accurate pose estimates is crucial for any localization approach, the run-time requirements and complexity deploying the model play important roles in its ease of use on various robotic systems. In order to evaluate the feasibility of deploying our VLocNet++$_{\mathrm{MTL}}$ on robotic platforms, we compare the run-time, pose accuracy, median localization error and input requirements of our approach with the previous state-of-the-art local feature-based method on the Microsoft 7-Scenes benchmark in Table 4.17.

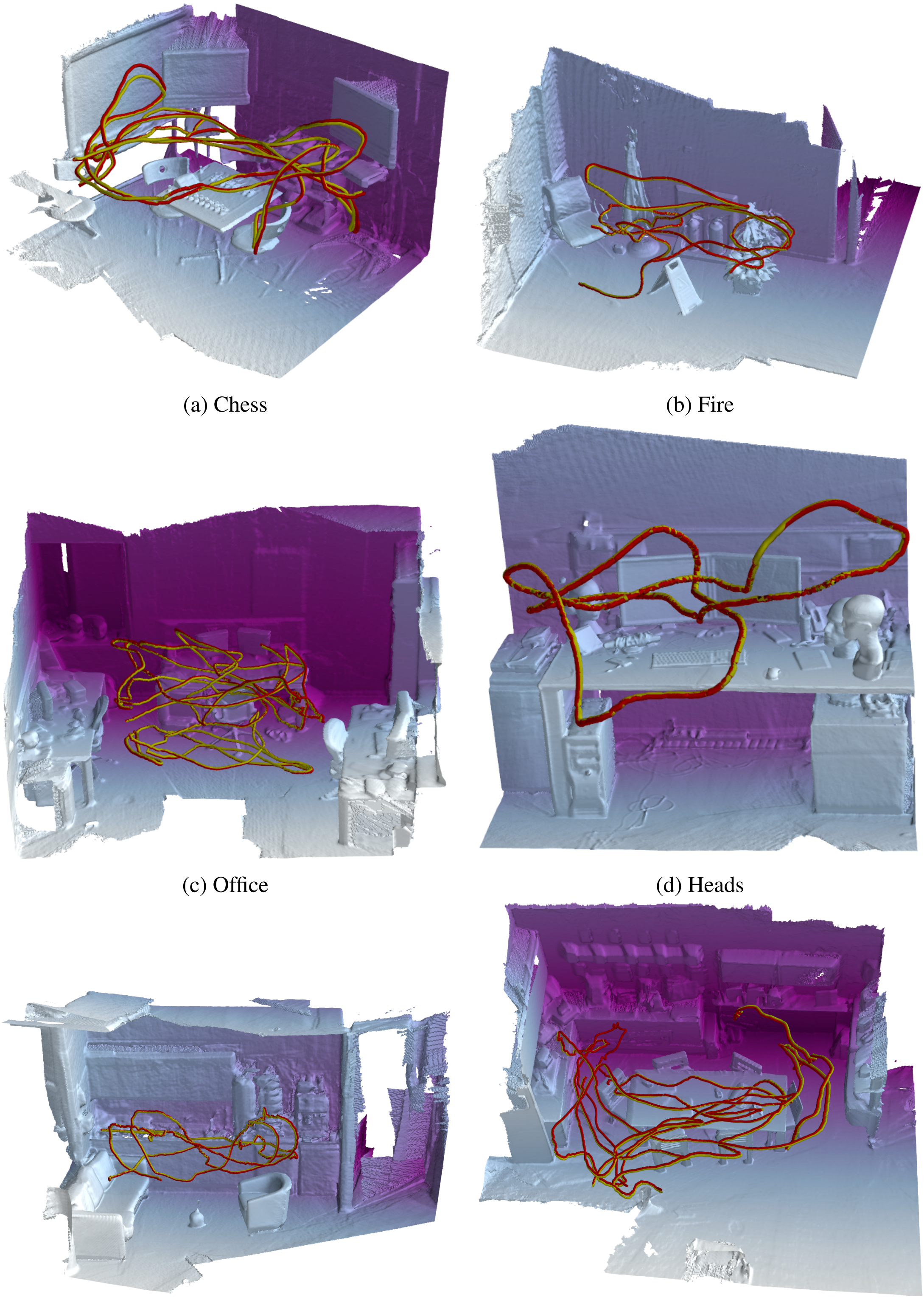
Table 4.17: Comparison with the state of the art on the Microsoft 7-Scenes dataset. We evaluate the performance in terms of pose accuracy, run-time and input requirements [158].

Method	Input	Median Translational Error	Median Rotational Error	Pose Accuracy	Run-time
DSAC2 [41]	w/ 3D	0.04m	$1.04°$	76.1%	200ms
VLocNet++$_{\text{MTL}}$ (Ours)	Monocular	$\mathbf{0.013}$m	$\mathbf{0.77}°$	$\mathbf{99.2}\%$	$\mathbf{79}$ms

While DSAC2 [41] is currently considered the state of the art on the Microsoft 7-Scenes dataset, the results in Table 4.17 demonstrate that our VLocNet++$_{\text{MTL}}$ exceeds the state-of-the-art localization accuracy by 67.5% in the translational and 25.9% in the rotational components of the pose. Furthermore, unlike DSAC2 [41], our proposed method does not require a 3D model of the scene. This in turn facilitates ease of deployment, in addition to occupying less space for the model. Moreover, the run-time of VLocNet++$_{\text{MTL}}$ is 60.5% faster (run on a single consumer grade GPU) than that of DSAC2 [41], rendering our method well suited for real-time deployment in an online manner, as well as on resource restricted platforms [157].

We further analyze the localization accuracy of the poses predicted by VLocNet++$_{\text{MTL}}$ by depicting visual representations of the predicted poses and the ground-truth poses on the three benchmarking datasets; Microsoft 7-Scenes, DeepLoc and DeepLocCross. Results from this experiment are shown in Figure 4.10, where the predicted poses are shown in yellow and the ground-truth poses in red. Note that for each of the scenes, we depict the 3D scene for visualization purposes solely, as our framework takes only monocular images of the scene as an input. We show only the first test loop for both the DeepLoc and DeepLocCross datasets, as visualizing all test loops in one scene results in an intertwined output that is visually difficult to analyze. Interactive visualizations of the 3D scene models, depicting the ground-truth and predicted trajectories can be found at `http://deeploc.cs.uni-freiburg.de`.

VLocNet++$_{\text{MTL}}$ accurately estimates the global pose in both indoor (a-g) and outdoor (h, i) environments while being robust to textureless regions (Figure 4.3(b, c)), dynamic objects (Figure 4.5), repetitive as well as reflective structures (Figure 4.3(e, f, g)) and motion blur (Figure 4.3(a, d)). Our proposed multitask learning framework is able to accurately predict the poses regardless of the aforementioned challenges. Furthermore, through employing the proposed adaptive weighted fusion layer for fusing features across multiple timesteps and tasks, VLocNet++ is able to accurately correlate the motion-specific spatial features crucial for the localization task using only monocular input images of the scene.



(a) Chess

(b) Fire

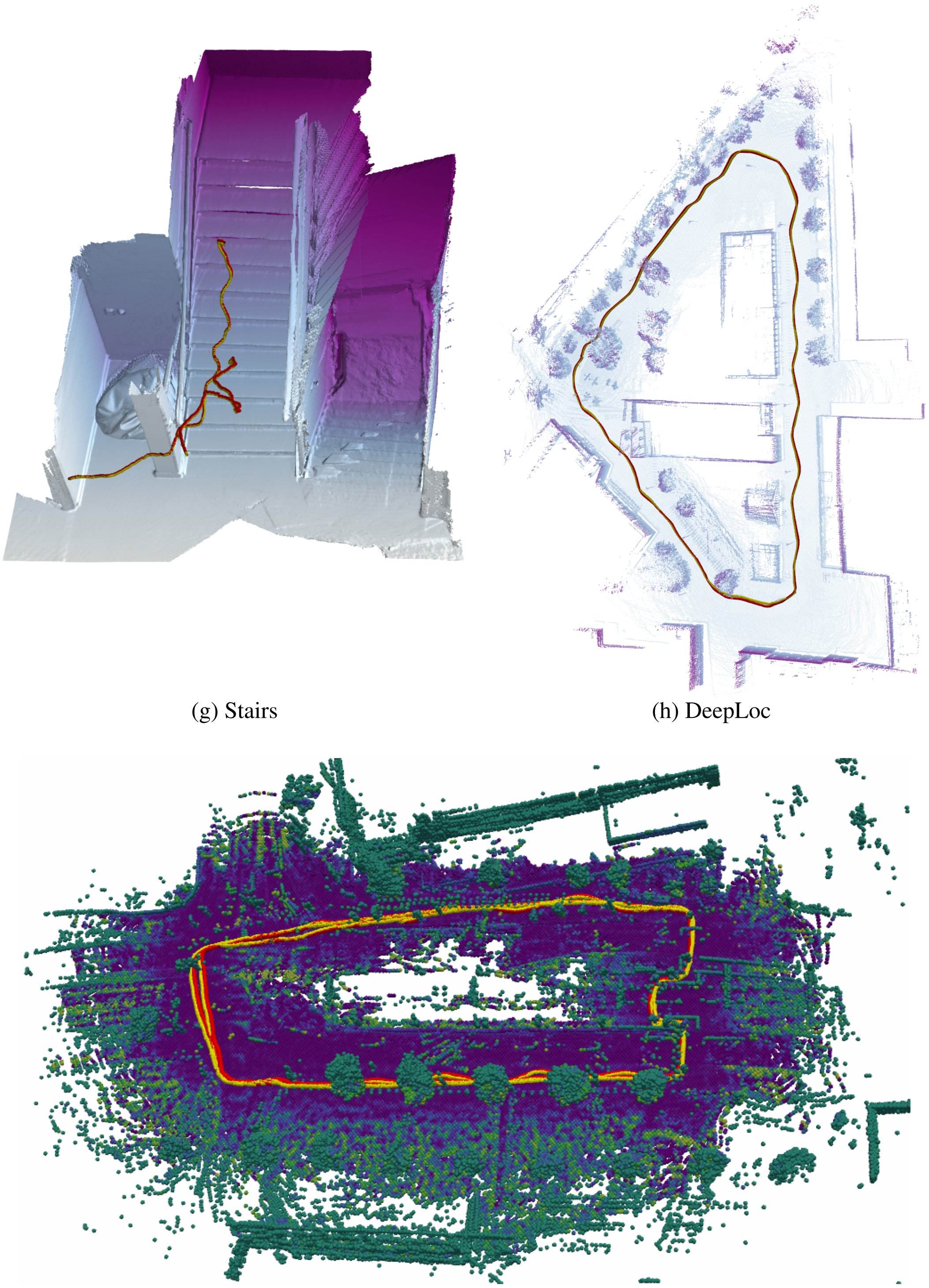
(c) Office

(d) Heads

(e) Pumpkin

(f) RedKitchen

Figure 4.10: Qualitative results depicting the predicted global pose (yellow trajectory) versus the ground-truth pose (red trajectory) plotted with respect to the 3D scene model for visualization on the Microsoft 7-Scenes, DeepLoc and DeepLocCross datasets [157]. Interactive visualizations: `http://deeploc.cs.uni-freiburg.de`.



(g) Stairs

(h) DeepLoc

(i) DeepLocCross

Figure 4.10: Qualitative results depicting the predicted global pose (yellow trajectory) versus the ground-truth pose (red) plotted with respect to the 3D scene model for visualization on the Microsoft 7-Scenes, DeepLoc and DeepLocCross datasets (continued) [157]. Interactive visualizations: `http://deeploc.cs.uni-freiburg.de`.

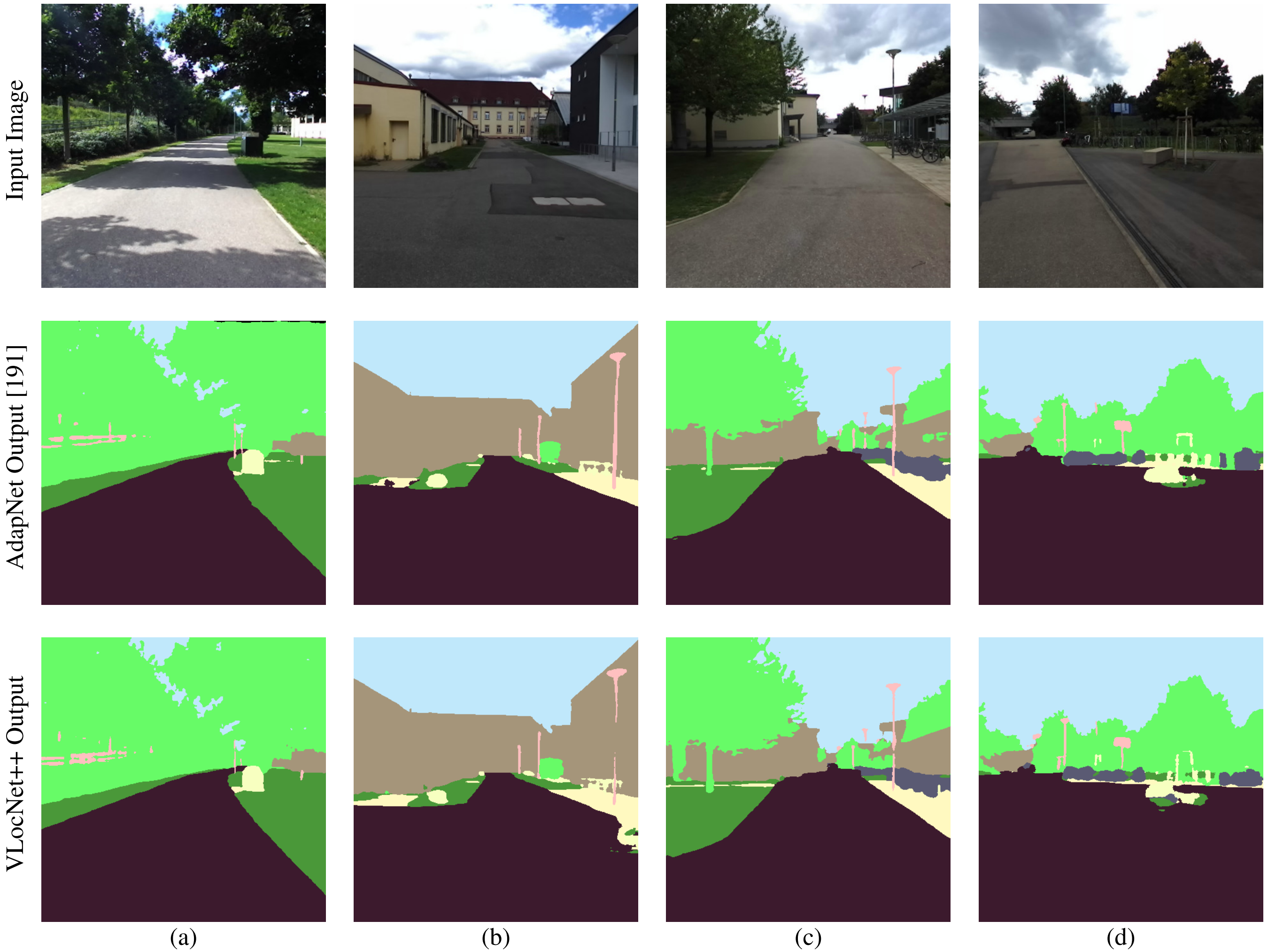


Figure 4.11: Qualitative comparison of semantic segmentation results obtained using AdapNet [191] versus VLocNet++ on the DeepLoc dataset. The semantic categories are: ■ Sky, ■ Road, ■ Sidewalk, ■ Grass, ■ Vegetation, ■ Building, ■ Poles, ■ Dynamic and ■ Other [157].

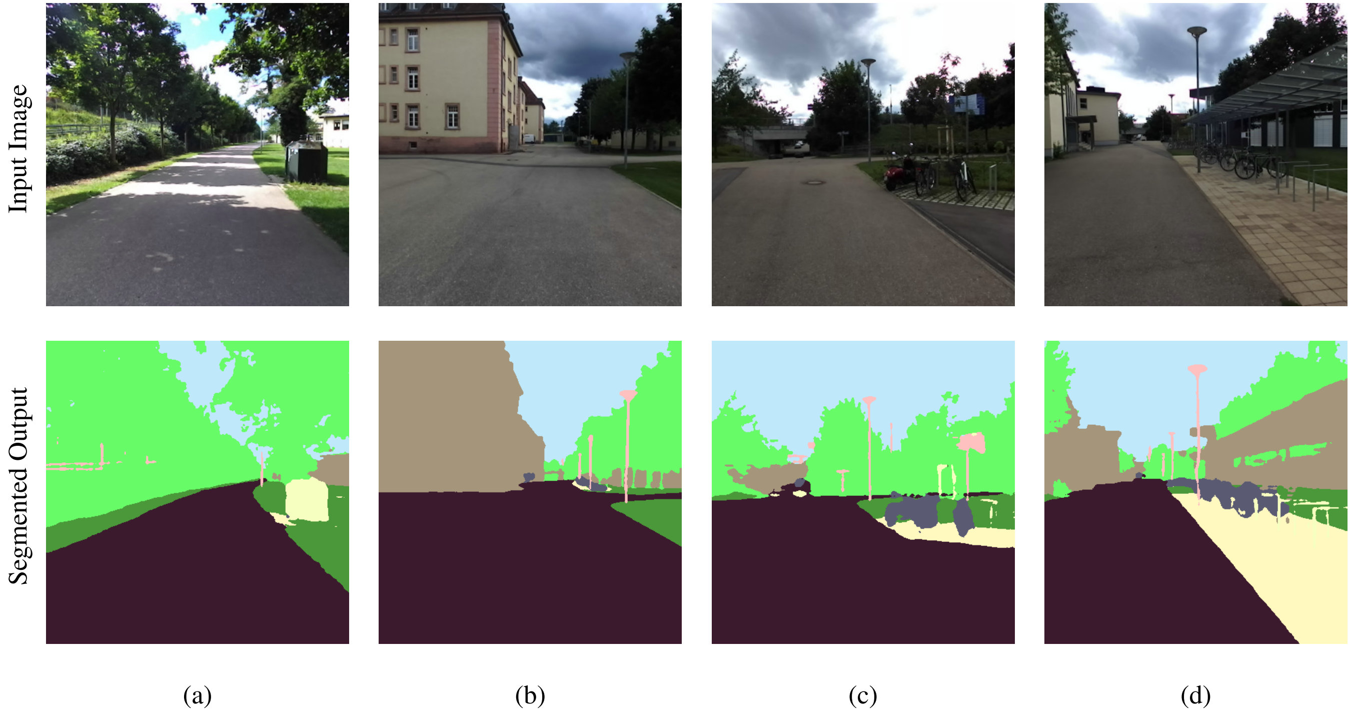


Figure 4.12: Qualitative evaluation of the segmentation results on the DeepLoc dataset in varying degrees of difficulty. The semantic categories are color coded as follows: ■ Sky, ■ Road, ■ Sidewalk, ■ Grass, ■ Vegetation, ■ Building, ■ Poles, ■ Dynamic and ■ Other. VLocNet++ is able to accurately segment the scene while being robust to varying lighting conditions (a, b, e), shadows (a), reflective/translucent glass surfaces (e, d, f, h), thin and partially occluded structures (b, c, d, e) [157].

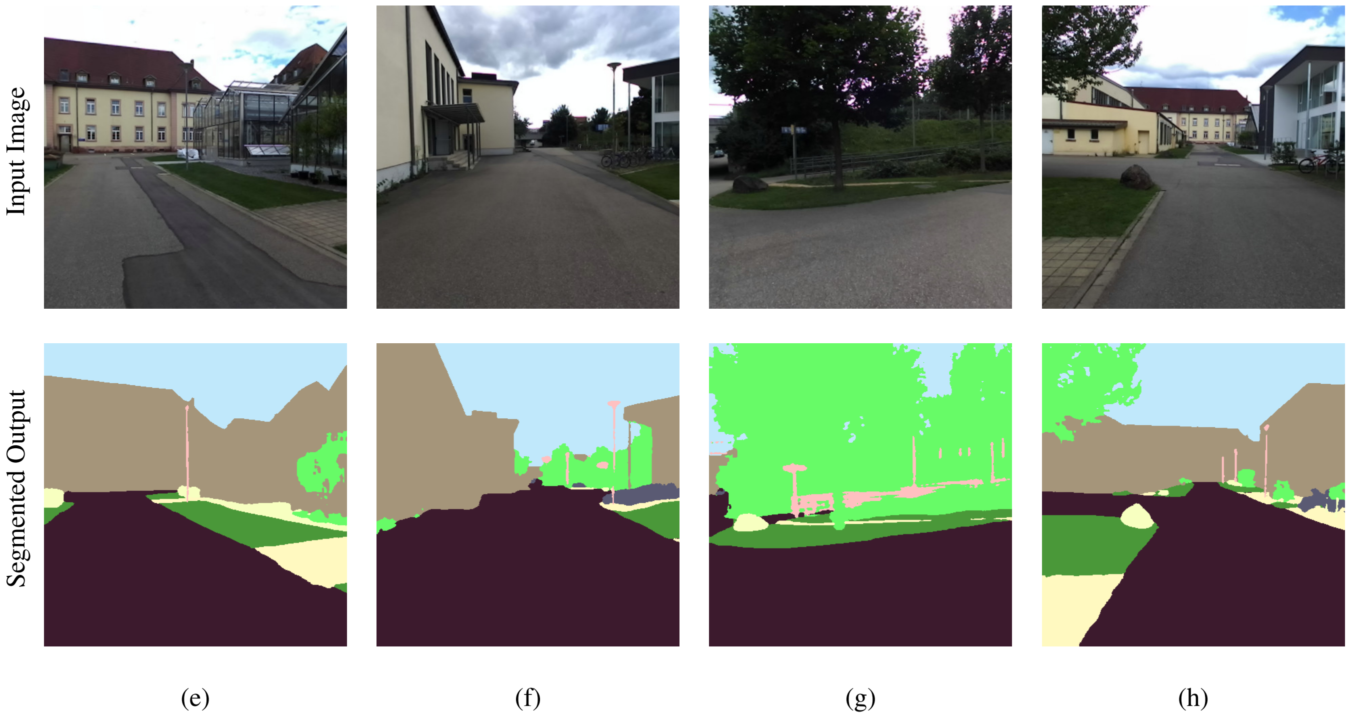


Figure 4.12: Qualitative evaluation of the segmentation results on the DeepLoc dataset in varying degrees of difficulty (continued). The semantic categories are color coded as follows: ■ Sky, ■ Road, ■ Sidewalk, ■ Grass, ■ Vegetation, ■ Building, ■ Poles, ■ Dynamic and ■ Other. VLocNet++ is able to accurately segment the scene while being robust to varying lighting conditions (5(a, b, e)), shadows (5(a)), reflective/translucent glass surfaces (5(e, d, f, h)), thin and partially occluded structures (5(b, c, d, e)) [157].

We further provide a qualitative analysis to examine the improvement in the segmentation predictions of the network due to the incorporation of the proposed self-supervised warping technique. Figure 4.11 shows the output segmentation masks of VLocNet$++_{\text{MTL}}$ in comparison to AdapNet [191] on the DeepLoc dataset. Figure 4.11(a, c) show distant sidewalk paths in the middle of the grass and structures partially occluded by vegetation which are quite challenging to detect given the small size of the input image. While the segmentation mask produced by AdapNet [191] often fails to capture the entire path/structure, VLocNet++ is able to precisely capture the boundary between between grass and sidewalk.

Figure 4.11(b, d) contain multiple thin structures such as poles and bike stands, which are either completely absent or only partially detected in the output of AdapNet [191]. Our proposed method is, however, able to accurately and precisely segment thin pole-like structures by aggregating the previous observations using the proposed self-supervised warping scheme. Figure 4.11(d) shows another challenging example, wherein a dark image is produced by the camera due to direct sunlight. While AdapNet [191] incorrectly classifies grass as a bench due to this artifact, VLocNet++ is able to reliably distinguish between the distinct categories while being tolerant to the various weather and illumination conditions.

We present additional qualitative segmentation results of our VLocNet$++_{\text{MTL}}$ in various challenging scenarios from the DeepLoc dataset in Figure 4.12. Figure 4.12(a, b, e, g) show various illumination conditions causing shadows, glare, over- and underexposure due to the sunlight. Nonetheless, in all these cases, VLocNet++ yields an accurate representation of the scene overcoming the disturbances. Furthermore, despite the small size of the input image and the abundance of thin pole-like structures such as lamp posts, signs and fences in the dataset, which contribute major challenges for any segmentation network, VLocNet++ is still able to detect the entire structure of the objects as can be seen in Figure 4.12(b, c, d, g). We attribute this to the ability of the network to aggregate information from previous observations using the dynamic warping scheme.

Among the challenging aspects of our DeepLoc dataset is the presence of reflective and translucent glass buildings as in Figure 4.12(d, e, f, h). Despite the presence of multiple glass-constructs, our method is able assign to them the correct semantic class. The ability to identify the boundary between grass and vegetation is a difficult problem even for humans. Figure 4.12(a, g) depict examples where our network was able to accurately predict such boundaries. Moreover, Figure 4.12(e, g) show images containing narrow paths surrounded by vegetation which are difficult to observe as the distance from the path increases. Inspecting the segmentation outputs of VLocNet++ for those cases shows the capability of our model to capture such distant thin passages.

Table 4.18: Evaluation of the semantic segmentation performance of our VLocNet++$_{\mathrm{MTL}}$ on the Cityscapes dataset.

Approach	Sky	Building	Road	Sidew.	Fence	Veg.	Pole	Car	Sign	Person	Cyclist	Mean
	🟦	⬛	🟪	🟪	🟫	🟩	⬜	🟦	🟨	🟥	⬛	IoU
FCN-8s [121]	76.51	83.97	93.82	67.67	24.91	86.38	31.71	84.80	50.92	59.89	59.11	59.97
SegNet [28]	73.74	79.29	92.70	59.88	13.63	81.89	26.18	78.83	31.44	45.03	43.46	52.17
FastNet [145]	77.69	86.25	94.97	72.99	31.02	88.06	38.34	88.42	52.34	61.76	61.83	68.52
ParseNet [119]	77.57	86.81	95.27	74.02	33.31	87.37	38.24	88.99	53.34	63.25	63.87	69.28
DeconvNet [144]	89.38	83.08	95.26	68.07	27.58	85.80	34.20	85.01	27.62	45.11	41.11	62.02
DeepLab v2 [53]	74.28	81.66	90.86	63.3	26.29	84.33	27.96	86.24	44.79	58.89	60.92	63.59
DeepLab v3 [54]	92.40	89.02	96.74	78.55	41.00	90.81	49.74	91.02	64.48	66.52	66.98	75.21
AdapNet [191]	93.20	90.46	97.38	81.11	54.85	91.36	52.72	92.53	68.07	72.36	70.69	78.61
VLocNet++$_{\mathrm{MTL}}$ (Ours)	**93.57**	**90.71**	**97.48**	**81.72**	**54.91**	**91.72**	**55.47**	**92.84**	**69.15**	**73.43**	**71.20**	**79.29**

4.4.4.6 Generalization Capabilities

In order to further demonstrate the benefit of utilizing the self-supervised warping for the segmentation accuracy, we provide qualitative analysis of the performance of our multitask model on the DeepLocCross dataset in comparison to AdapNet [191]. As we do not provide pixel-wise semantic annotations for this dataset, we train our model on the Cityscapes dataset [58] and evaluate the performance of the model on the DeepLocCross dataset due to the similarities in the object classes between both datasets. We train our VLocNet++$_{\text{MTL}}$ model for 12 categories of the Cityscapes dataset: *Background*, *Sky*, *Building*, *Road*, *Sidewalk*, *Fence*, *Vegetation*, *Pole*, *Car*, *Sign*, *Person* and *Cyclist*. We provide a qualitative evaluation of the performance of our VLocNet++$_{\text{MTL}}$ in comparison to several state-of-the-art methods for semantic segmentation: FCN-8s [121], SegNet [28], FastNet [145], ParseNet [119], DeconvNet [144], DeepLab v2 [53], DeepLab v3 [54] and AdapNet [191]. Table 4.18 shows the per-class and mean IoU for each of the aforementioned methods on the Cityscapes dataset.

Our proposed VLocNet++$_{\text{MTL}}$ outperforms state-of-the-art methods achieving a mean IoU of 79.29%. We observe the biggest improvement over the base AdapNet architecture [191] in some of the more difficult classes: the *Pole* category improves by 2.75%, the *Sign* class improves by 1.08% and similarly the *Cyclist* category improves by 0.51%. This further corroborates our hypothesis that by incorporating the self-supervised warping layer, our network is able to better distinguish narrow structures such as *Pole* as well as distinguish between semantically difficult categories such as *Person* and *Cyclist*.

Figure 4.13 shows the output segmentation masks of VLocNet++$_{\text{MTL}}$ in comparison to AdapNet [191] on the Cityscapes dataset. Figure 4.13(a) shows a complex scene image with multiple pedestrians and cyclists, which are particularly difficult to differentiate due to the small size of the image, the far distance at which they were observed and the partial occlusions within the image. Unlike AdapNet, our proposed VLocNet++$_{\text{MTL}}$ is able to accurately distinguish between the two categories despite the aforementioned challenges as can be seen from the image. Figure 4.13(b, c) show multiple poles and signs which are either entirely missing or only partially detected in the output of AdapNet. However, VLocNet++$_{\text{MTL}}$ is able to precisely capture the structure of these objects by aggregating the previous temporal features using the self-supervised warping layer. Figure 4.13(d) contains multiple short pole-like structures on the side of a building which are only partially segmented in the output of AdapNet. However, despite the dark image quality, by incorporating the temporal features using our proposed self-supervised warping scheme, VLocNet++$_{\text{MTL}}$ is able to accurately and precisely classify both the pole-like structures and the boundary between the building and sidewalk.

Finally, we evaluate the generalization capabilities of our proposed warping scheme by testing our VLocNet++$_{\text{MTL}}$ architecture trained on the Cityscapes dataset on images from the test sequence of the DeepLocCross dataset. Despite the similarity in the object

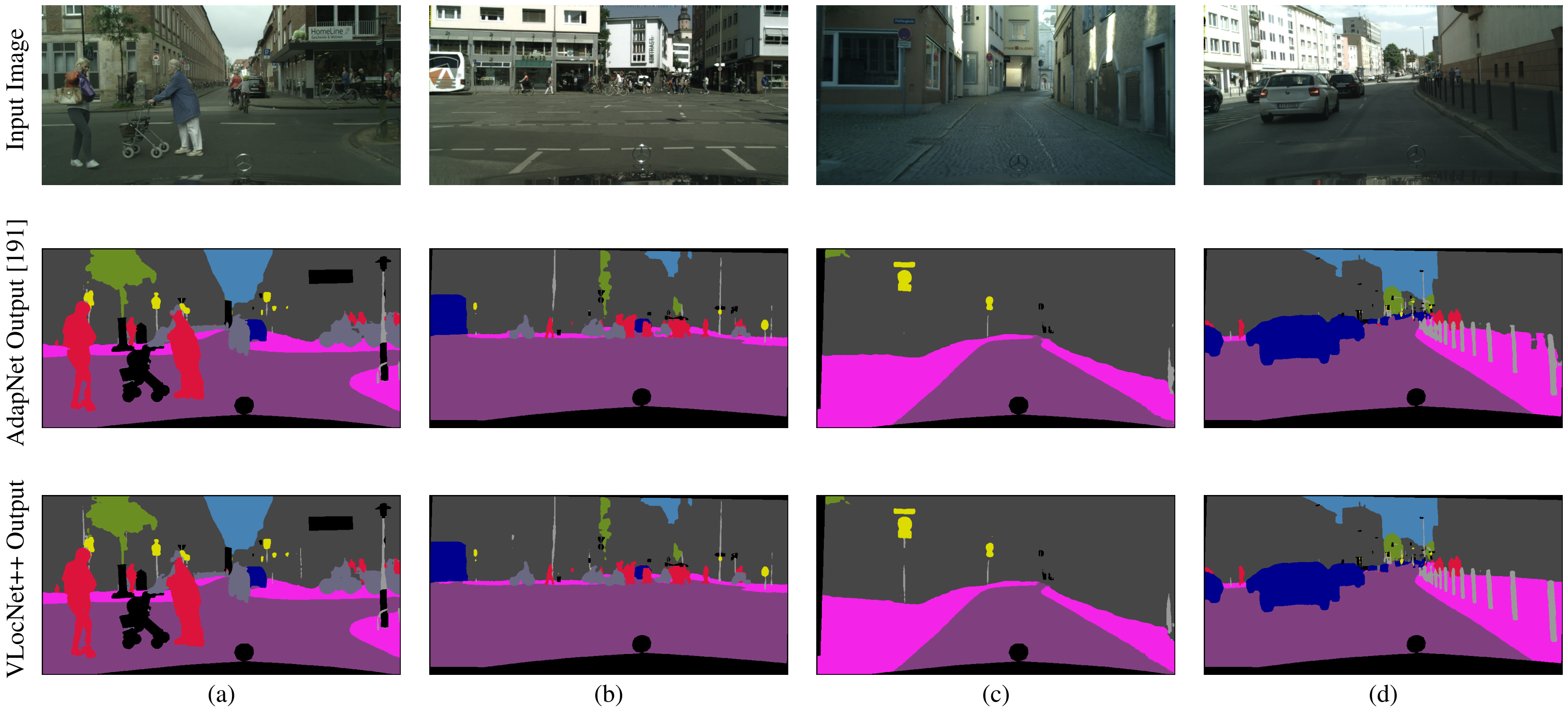


Figure 4.13: Qualitative comparison of semantic segmentation results obtained using AdapNet [191] versus VLocNet++ on the Cityscapes dataset. The semantic categories are color coded as follows: ■ Sky, ■ Building, ■ Road, ■ Sidewalk, ■ Fence, ■ Vegetation, ■ Pole, ■ Car, ■ Sign, ■ Person and ■ Cyclist.

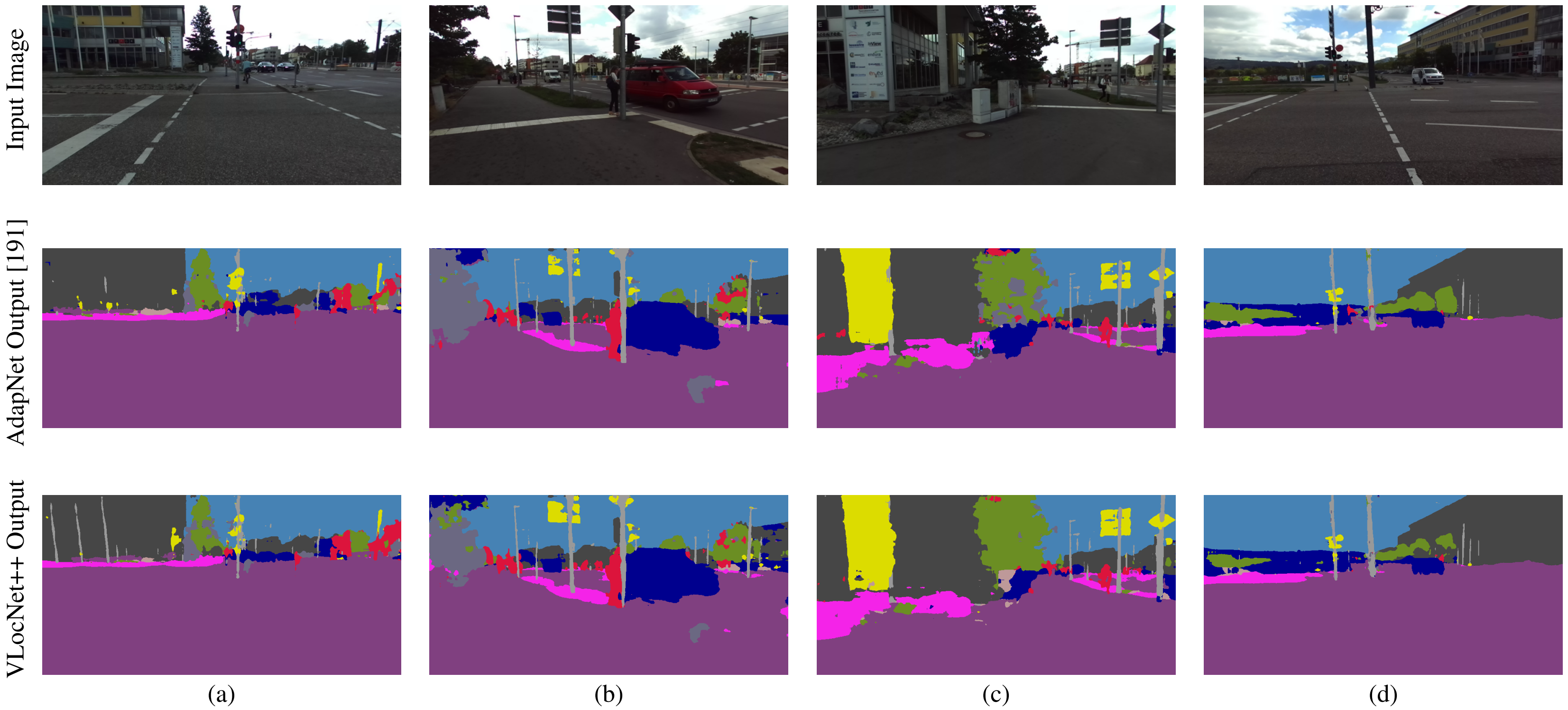


Figure 4.14: Qualitative comparison of semantic segmentation results obtained using AdapNet [191] versus VLocNet++ on the DeepLocCross dataset demonstrating the generalization capabilities of the network. The semantic categories are color coded as follows: ■ Sky, ■ Building, ■ Road, ■ Sidewalk, ■ Fence, ■ Vegetation, ■ Pole, ■ Car, ■ Sign, ■ Person and ■ Cyclist.

classes observed in both datasets, the perspective from which the images were captured is completely different. The Cityscapes dataset was captured with a camera mounted to a moving vehicle as it traversed the streets of various cities. On the other hand, the DeepLocCross dataset captures data from a pedestrian perspective as the robot traverses on the sidewalk. We expect this change in perspective to affect the segmentation output of the network. Figure 4.14 shows four sample images from the DeepLocCross dataset. For each image, we show the segmentation mask predicted by AdapNet [191] and our VLocNet++$_{\mathrm{MTL}}$. Figure 4.14(a, d) show scenes with multiple poles, cars and signs. VLocNet++$_{\mathrm{MTL}}$ is able to accurately segment the poles and remaining structures in the scene while showing better generalization capabilities in comparison to AdapNet.

Figure 4.14(b, c) show challenging images where the robot is traversing the sidewalk, however, due to the variance between the Cityscapes dataset and the images from the DeepLocCross dataset, the sidewalk is misclassified as road. This occurs as the training images did not contain any examples where the car is traversing over the sidewalk. We believe, however, that the inclusion of training images containing different perspectives should remedy this issue. Nonetheless, we observe that the classes *Poles*, *Signs* and *Person* are more accurately and precisely segmented by our proposed self-supervised warping scheme. Overall, we observe that despite the perspective differences, utilizing our proposed warping scheme enables the network to better generalize to different environments as can be seen from the predicted segmentation masks when compared to the predictions of AdapNet.

4.5 Related Work

Over the past decade there has been a gradual shift from employing traditional handcrafted pipelines to learning-based methods particularly for perception related tasks. In this section, we discuss some of the recent learning-based approaches for multitask learning, pose regression and semantic segmentation.

Multitask Learning is becoming more popular over the years, with applications covering a wide range of tasks including mage understanding [206], sentiment prediction [92], semantic segmentation [182] and recently even on learning from demonstration [159]. Multitask learning can be defined as an inductive transfer mechanism that improves generalization by leveraging domain specific information from related tasks [51]. In [36], Bilen *et al.* propose the use of an instance normalization network to train a network that recognizes objects across multiple visual domains including digits, signs and faces. In [174], the authors introduce a model with a sparsely-gated mixture of experts layer containing thousands of feed-forward sub-networks for the task of language modeling and machine translation. For combining different loss functions in a multitask model, Kendall *et al.* [96] propose a loss function based on maximizing the Gaussian likelihood using a homoscedastic task uncertainty. While in most of the aforementioned approaches, the parts of the network that learn low-level features are shared among the different tasks, followed by separate task-specific branches, our proposed method employs a novel weighted fusion layer to inductively share representations across streams through learning a favorable weighting among the feature maps for the mutual benefit of the tasks.

Visual Localization methods are commonly classified as either metric or appearance-based approaches. Appearance-based localization methods [168, 185] employ image retrieval techniques to find closely matching images to the query image from an image/feature database, thereby providing a coarse estimate of the location of the query image. On the contrary, metric-based localization approaches estimate correspondences between local features in the image, with the goal of estimating the full 6-DoF pose. Once the correspondences are found, Structure-from-Motion (SfM) or Simultaneous Localization and Mapping (SLAM) [130, 167] is applied to estimate the geometric relation between the input image and the 3D model or map of the scene. In this chapter, we primarily focused on metric-based localization methods. While the predominant metric-based localization approaches rely on local sparse feature correspondences, the success of deep convolutional neural networks in classification and semantic segmentation has led to a surge in the number of deep learning-based methods adapted for pose regression.

Sparse feature-based localization approaches learn a set of feature descriptors from the training images. The learned features are then employed to learn a codebook of 3D descriptors against which a query image can be matched [79, 169]. In order to efficiently find feature correspondences within the codebook, Shotton *et al.* [175] and Valentin *et al.* [195] train regression forests on 3D scene data and employ RANSAC [68] to infer the final location of the query image. Donoser *et al.* [66] propose a discriminative classification approach using random ferns, which improves the pose accuracy while allowing for faster run-time. While sparse feature-based localization methods are able to provide accurate pose estimates, the overall run-time of such approaches depends on the size of the 3D model and number of feature correspondences found. This in turn results in suboptimal performance in textureless environments and scenes with repetitive structures.

The first deep learning-based localization approach was proposed by Kendall *et al.* [95]. The authors introduced PoseNet, a CNN architecture that given a monocular input image estimated the 6-DoF pose. The emergence of PoseNet has led to a subsequent surge in deep learning-based localization methods. Subsequent improvements included using Dropout units to act as a regularizer for estimating the uncertainty of the network predictions [93], incorporating Long-Short Term Memory (LSTM) units for dimensionality reduction [199], supervising the representations learned by the network through using an encoder-decoder architecture during training [132], and proposing loss functions that leverage the scene geometry [94]. Laskar *et al.* [111] propose a CNN approach that combines both metric and appearance-based localization methods. The authors train a network on relative camera pose estimation, and utilize the features learned by the network to identify closely matching images from a database. Brachmann *et al.* [42] propose a differentiable version of RANSAC, dubbed DSAC. Their proposed approach modifies upon the standard RANSAC by replacing the deterministic pose hypothesis either a soft argmax selection or a probabilistic selection, thereby rendering the method differentiable and suitable for online learning through back-propagation. In subsequent work [41], the authors introduce a second version of the method that employs an entropy controlled soft inlier count for scoring the pose hypotheses predicted by the network.

A majority of the deep learning-based localization methods utilize pre-trained classification networks as an architecture backbone, and modify the network through the addition of inner product layers coupled with a Euclidean loss function for pose regression. On the one hand, the features learned by the networks are, unlike local feature-based methods, robust to motion blur and perceptual aliasing. However, the accuracy of the CNN-based methods remains substantially lower than local feature-based localization methods. In this chapter, we proposed adaptively fusing the previous motion information into the pose regression network to enable the network to learn poses that are consistent with the motion model. Furthermore, we employed a novel loss function that enables the aggregation of motion-specific features across the temporal domain, thereby enabling the network to learn a model that is globally consistent.

Visual Odometry: Estimating the motion of the camera due to the robot motion, or ego-motion estimation, is a closely related task to that of visual localization. Among the earlier approaches to tackle this problem is that of Konda *et al.* [101], in which the authors employ a CNN with a softmax layer to infer the relative transformation between the input images. In their work, the authors treat the task as a classification problem, and attempt to infer the transformation from a prior discretized set of velocities and directions. Using inputs as both images and LiDAR data, Nicolai *et al.* [143] proposed a Siamese architecture with alternating convolutional and pooling layers to estimate the transformations from consecutive point clouds. Their proposed approach projects the point cloud data on the 2D image and subsequently feeds this information to a neural network which estimates the visual odometry.

Mohanty *et al.* [135] propose an AlexNet-based [104] Siamese architecture called DeepVO for odometry estimation from monocular images and their corresponding FAST features [163]. Their architecture employs a Euclidean loss function during training with equal weight values to regress the translational and rotational pose components. In similar work, Melekhov *et al.* [133] add a weighting term to balance the translational and rotational components of the loss, yielding an improvement to the predicted pose. In order to render their approach robust to varying image resolutions, their network architecture incorporates a spatial pyramid pooling layer. Taking inspiration from the aforementioned works, and the success of residual networks in various vision-based tasks, we proposed a Siamese-type dual stream architecture built upon the ResNet-50 [82] model for visual odometry estimation.

Semantic Segmentation: Fully Convolutional Neural Networks (FCNs) [121] was the first approach to propose the encoder-decoder model replacing inner product layers in classification networks with convolutional layers to enable pixel-wise classification. This lead to a tremendous increase in the performance of various scene parsing tasks, with several networks building upon FCNs by introducing more refinement stages to improve the granularity of the segmentation [145], employing efficient non-linear upsampling schemes [28], adding global context [119] and pyramid pooling for context aggregation [209]. Yu *et al.* [207] proposed a context module utilizing dilated convolutions in order to enlarge the receptive field. DeepLab [53] proposed the use of multiple parallel dilated convolutions with different sampling rates for multi-scale learning in addition to employing Conditional Random Fields for post-processing the predictions of the network. Valada *et al.* [191] introduced the AdapNet architecture built on ResNet, in which they introduce multi-scale residual blocks containing dilated parallel convolutions, thereby enabling faster inference times without compromising on the performance.

Ma *et al.* [124] proposed an RGB-D semantic mapping method which incorporates a warping procedure for warping frames with no ground-truth into nearby frames with ground-truth annotation. For learning consistent semantics in VLocNet++, we build upon AdapNet's model and fuse feature maps from the preceding frame into the current frame through warping by utilizing the predictive relative pose from the odometry stream. Unlike the approach of Ma *et al.* [124] which incorporates the warped feature maps to aid in calculating the supervised loss for frames without ground-truth information, in this chapter, we warp the feature maps of the preceding frame into the current frame at multiple downsampling stages in order to enable multi-view aggregation which subsequently leads to improved accuracy and faster convergence time.

In this chapter, we proposed a multitask deep convolutional neural network architecture for 6-DoF visual localization, visual odometry estimation and semantic scene segmentation from consecutive monocular camera images. In order to leverage the inherent interdependencies between the three tasks, we proposed an adaptive weighted fusion layer that learns a favorable weighted combination of the feature maps by utilizing the region activations. Furthermore, we proposed a loss function that enables the network to regress poses consistent with the true motion model. We additionally proposed a self-supervised warping method to enable the multi-view aggregation of features at different downsampling stages thereby enabling the network to predict more accurate segmentation masks while reducing the training time.

4.6 Conclusion

In this chapter, we introduced an end-to-end trainable multitask convolutional neural network that addresses the problem of visual pose regression. We built upon the VLocNet architecture, which employs auxiliary learning techniques to jointly regress the 6-DoF visual localization and 6-DoF visual odometry from consecutive monocular images. Using an efficient sharing scheme and a joint optimization strategy, we enabled the network to exploit the inter-task correlations for the mutual benefit of both tasks. We introduced the VLocNet++ architecture for simultaneously learning 6-DoF visual localization, semantic segmentation and odometry estimation, with the goal of exploiting the interdependencies within these tasks for their mutual benefit. We proposed an approach for incorporating geometric and structural priors into the visual localization network through aggregating semantic and motion-specific features learned through a joint optimization strategy and an efficient sharing scheme. In order to efficiently and adaptively share features, we proposed an adaptive weighted fusion layer that learns the most favorable weighting for fusion based on the region activations. We further presented a self-supervised warping technique for scene-level context aggregation in semantic segmentation networks that improves the segmentation accuracy, decreases the convergence time while adding minimal computational overhead. In order to enable the efficient encoding of the geometric features into the pose regression network, we introduced the Geometric Consistency loss function. By minimizing this loss function, in combination with incorporating the previous pose information using our adaptive weighted fusion layer, we enable the model to efficiently leverage the pose information and thus learn a model that is geometrically consistent with respect to the motion.

In order to evaluate the performance of our proposed model, we introduced two large-scale outdoor localization datasets with multiple loops captured in different settings, which we make publicly available. Both datasets contain challenging scenarios for perception related tasks ranging from motion blur, substantial illumination changes, perceptual aliasing, thin/translucent structures and presence of multiple dynamic objects. Using extensive experimental evaluations on an indoor benchmark dataset, in addition to both outdoor datasets, we show that both our single-task as well as multitask models achieve state-of-the-art performance compared to existing deep learning-based approaches. Furthermore, on the challenging Microsoft 7-Scenes dataset, our method outperforms the previous state-of-the-art by 67.5%, 25.9% in the translational and rotational pose components, while being 2.5-times faster. Thereby, our method does not only close the gap between local feature-based and deep learning-based methods, but is also the first method to outperform the state of the art while simultaneously predicting multiple tasks. We presented extensive ablation and qualitative studies justifying the various design choices as well as visualizing the representations learned by the network.

Chapter 5

Mutlimodal Interaction-Aware Motion Prediction

The ability to safely cross street intersections is essential for mobile robots navigating on sidewalks. Most existing approaches rely on the recognition of the traffic light signal to make an informed crossing decision. Although these approaches have been crucial enablers for urban navigation, the capabilities of robots employing such approaches are still limited to navigating only on streets that contain signalized intersections. In this chapter, we address this challenge and propose a multimodal convolutional neural network framework to predict the safety of a street intersection for crossing. Our architecture consists of two sub-networks: an interaction-aware trajectory estimation stream (IA-TCNN), that predicts the future states of all observed traffic participants in the scene, and a traffic light recognition stream (AtteNet). Learned representations from the traffic light recognition stream are fused with the estimated trajectories from the motion prediction stream to learn a crossing decision that is invariant to the type of the intersection. Moreover, incorporating the uncertainty information from both modules enables our architecture to learn a likelihood function that is robust to noise and mispredictions from either sub-network. We further introduce the Freiburg Street Crossing dataset which contains sequences captured at multiple intersections of varying types, demonstrating complex interactions among the traffic participants as well as various weather conditions. Extensive experimental evaluations on public benchmark datasets and our Freiburg Street Crossing dataset demonstrate that our network achieves state-of-the-art performance for each of the sub-tasks, as well as for the street crossing safety prediction.

5.1 Introduction

Thus far in this thesis, we proposed various approaches for mobile robots operating in urban environments to robustly localize themselves, thereby facilitating their deployment for day-to-day tasks whether as navigational aids, autonomous driving vehicles or last-mile delivery agents. In most of these applications, as mobile robots navigate closely around humans, it is essential that they follow our navigational conventions while also being robust to unexpected situations. In this context, the ability to autonomously navigate across street intersections is one of the main situations that robots should be able to handle safely.

In order to decide if a street intersection is safe for crossing, humans are taught from an early age to follow a rigorous decision making process which is comprised of checking and waiting for the traffic light signal, followed by looking in both directions to ensure the safety of the intersection for crossing. Hence, solely relying on the traffic light information to make the crossing decision is suboptimal as not only is the traffic light recognition task challenging in itself, the signal alone does not ensure the intersection safety for crossing. For example, when a speeding vehicle such as an ambulance or a firetruck approaches an intersection, it has the right of way as it does not necessarily follow the traffic regulations. Traffic participants such as pedestrians and vehicles are required to wait until the intersection becomes clear. This problem becomes even more challenging with the varying types of intersections and the associated rules on how to cross each variant. For instance, the standard convention at zebra crossings is that the pedestrian has the priority for crossing the intersection, whereas the oncoming traffic slows down and stops until they have crossed. On the other hand, at unmarked intersections such as a side street merging into a main road, there is neither a traffic light to regulate the crossing nor does the pedestrian have the right of way. Further complicating the problem, the topology of the road such as street width, presence of a middle island and direction of traffic play an important role in determining the crossing procedure. Hence, hard-coding a set of behavioral rules for a mobile robot to abide by at intersections is not only infeasible, but also requires constant upkeep and tailoring to suit varying scenarios that change with each region or city.

In this chapter, we propose a convolutional neural network framework to address the problem of autonomous street crossing while considering the dynamicity of the scene as well as factors that influence the crossing decision such as the presence of a traffic light. Our network consists of two streams, an interaction-aware motion prediction stream to estimate the future states of all traffic participants in the vicinity and a traffic light recognition stream to predict the state of the traffic light. Our framework fuses feature maps from both network streams to learn the crossing decision in an end-to-end manner, rendering it tolerant to noise and mispredictions by either sub-network, as well as making it inherently agnostic to the type of intersection.

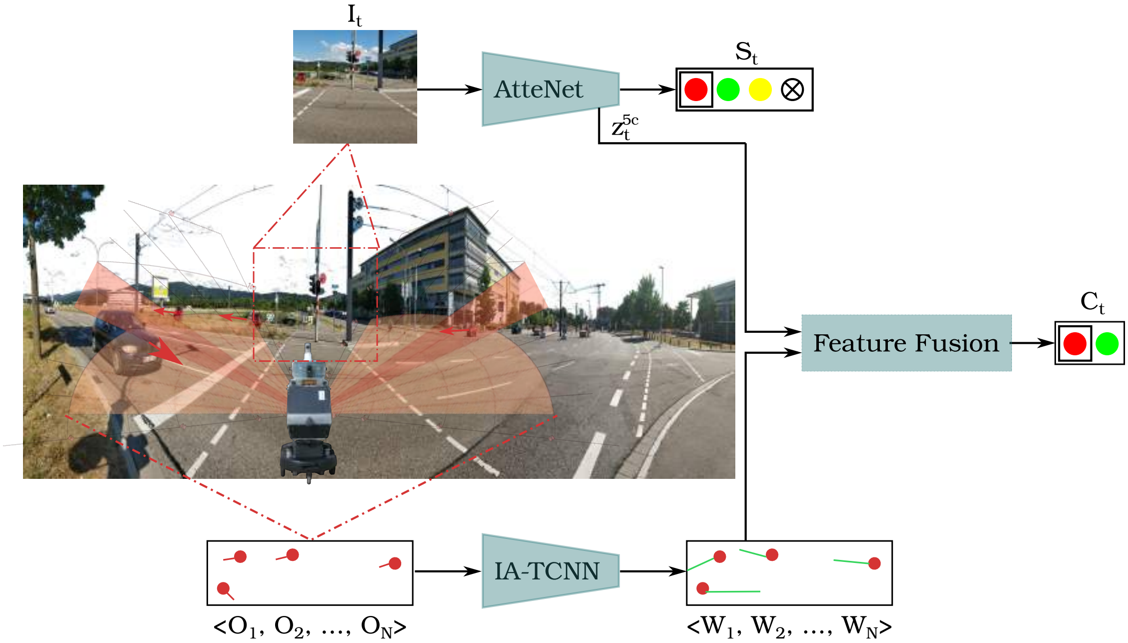


Figure 5.1: Schematic representation of our proposed system for autonomous street crossing. Our approach is comprised of two main modules: a traffic light recognition network and an interaction-aware motion prediction network. Feature map outputs of both modules are utilized to predict the safety of the intersection for crossing.

Predicting and modeling the behavior of agents, whether pedestrians or vehicles, is an extremely challenging problem that requires understanding the navigation conventions as well as the complex interactions among the various agents. For humans, identifying and following these conventions during navigation is a skill learned over several years that often needs readjustment depending on the environment. Hence, formalizing a set of behavioral rules for a mobile robot to uphold is both complex and taxing, requiring constant maintenance for each new environment. Recently, learning based motion prediction approaches [24, 97] have shown considerable robustness in modeling interactions among agents in real-world scenarios. However, as the density of the scene increases, their run-time and representational capabilities decrease, as they rely on modeling each agent separately by considering only their local neighborhood.

In order to address these problems, we propose the novel Interaction-aware Temporal Convolutional Neural Network (IA-TCNN) architecture for interaction-aware motion prediction to jointly estimate the future trajectories of all observed agents in the scene. By utilizing a data driven method to represent the behavior of the different agents, we enable our approach to leverage the inherent interdependencies in their motion, thereby learning interactions without manually specifying a set of behavioral rules [84, 148]. While, sequence modeling problems such as trajectory estimation have been mostly tackled using recurrent neural networks, recent studies have shown that temporal convolutional neural networks are able to more effectively model such tasks [29].

We propose the novel AtteNet architecture for traffic light recognition that is robust to varying weather and lighting conditions. Our architecture incorporates Squeeze-Excitation blocks [86], thereby enabling it to learn a robust feature recalibration method that explicitly models the complex interdependencies between the channels of the various feature maps. This allows the network to actively suppress irrelevant features in the scene and highlight the most relevant features, which in turn enables it to learn representations that are robust to noise. Our aim by learning the traffic light signal is to incorporate the information into the street crossing predictor, thereby enabling our method to learn a model that acts in accordance with the navigational norms.

We introduce a real-world dataset captured at different intersections in Freiburg, which we also make publicly available. The data contains over $1,200$ annotated scenes of crossing scenarios, tracked detections for nearby traffic participants, and RGB images of pedestrian traffic lights in challenging weather conditions.

Figure 5.1 depicts the proposed architecture for intersection safety prediction along with the constituting sub-networks. The input to our network is an RGB image of the scene and the trajectories for all observed dynamic agents over an interval of time. Our network simultaneously predicts the traffic light signal, the future states of all traffic participants over a prediction window and the safety of the intersection for crossing during this interval. As our method does not rely on structural knowledge of the environment or any form of communication with the surrounding traffic participants, it can be applied independently of the intersection type. We benchmark our IA-TCNN architecture on several publicly available datasets, namely ETH [148], UCY [115] and L-CAS [204], in addition to our own Freiburg Street Crossing (FSC) dataset. For the traffic light recognition task, we benchmark on the Nexar [142] and Bosch [35] datasets as well as the Freiburg Street Crossing dataset. While for the autonomous crossing prediction, we perform extensive experimental evaluations on the Freiburg Street Crossing dataset. The results demonstrate that our architecture achieves state-of-the-art performance on each of the tasks.

In summary, our key contributions are:

- A novel multimodal convolutional neural network architecture for intersection crossing safety prediction that jointly predicts the state of the traffic light and the future trajectories of all traffic participants in the vicinity. Our network then utilizes information from both sub-networks to learn a crossing decision that is invariant to the intersection types as well as the underlying road topologies.

- The novel IA-TCNN architecture for interaction-aware motion prediction which employs causal convolutions to model the complex behavior and interactions among all observed agents in a scene while maintaining a fast inference time and being efficiently deployable in robots with limited resources.

- The novel AtteNet architecture for traffic light recognition that utilizes Squeeze-Excitation blocks to learn robust representations by leveraging global information to adaptively select relevant features from the input data.
- The Freiburg Street Crossing dataset captured at various intersections with annotations for the traffic light state, trajectory annotations for the tracked dynamic objects and labels for the intersection safety for crossing which we make publicly available to encourage future research in interaction-aware motion prediction and autonomous street crossing.
- We present extensive qualitative and quantitative analysis on each of the proposed modules on various publicly available benchmarks, in addition to our real-world dataset, demonstrating their efficacy in challenging real-world scenarios.

5.2 Technical Approach

In this section, we detail our proposed system for predicting the safety of the intersection for crossing by jointly learning to predict the future motion of the observed traffic participants and simultaneously recognizing the traffic light state. Our framework fuses the predicted future states as well as the uncertainties in the trajectories of the traffic participants from the motion prediction stream with feature maps from the traffic light recognition stream in order to predict the safety of the intersection for crossing. Note that the proposed networks for interaction-aware motion prediction and traffic light recognition can be deployed separately for their respective tasks.

Our proposed architecture, depicted in Figure 5.1, consists of two convolutional neural network streams: an interaction-aware motion prediction stream IA-TCNN and a traffic light recognition stream AtteNet. The learned representations from both streams are concatenated channel-wise and passed to the road crossing module, which in turn produces a likelihood over the crossing decision. Given an RGB image at the current timestep and the trajectory information for each dynamic object over a window of time, the output of our model is the traffic light state, the predicted trajectory for each object over the prediction interval and the crossing decision. We define dynamic objects as objects that have non-zero velocity during the observation interval. In the following sections, we will first detail each of the networks, followed by the fusion procedure for predicting the safety of the intersection for crossing.

5.2.1 Interaction-Aware Motion Prediction

Given the trajectory information for each dynamic object over a certain time period, the output of our model is the corresponding trajectory for each dynamic object over the prediction interval. We define the trajectory $\mathcal{O}_i$ for object i during an observation interval $T_{\text{obs}} = \{1, \ldots, t_{\text{obs}}\}$ as:

$$\mathcal{O}_i = \left\{ \left(x_i^t, y_i^t, v_i^t, qw_i^t, qz_i^t\right) \in \mathbb{R}^5 \mid t \in T_{\text{obs}} \right\}, \tag{5.1}$$

where each trajectory point is represented by the spatial coordinates (x_i^t, y_i^t), the velocity v_i^t and the yaw angle in the velocity direction in normalized quaternion representation $\mathbf{q}_i^t$. Since we do not utilize the roll and pitch angles, we drop their respective components in the quaternion representation leading to $\mathbf{q}_i^t = (qw_i^t, qz_i^t)$. Our network produces the predicted trajectory $\mathcal{W}_i$ over the interval $T_{\text{pred}} = \{t_{\text{obs}} + 1, \ldots, t_{\text{pred}}\}$ such that:

$$\mathcal{W}_i = \left\{ \left(x_i^t, y_i^t, v_i^t, qw_i^t, qz_i^t\right) \in \mathbb{R}^5 \mid t \in T_{\text{pred}} \right\}. \tag{5.2}$$

In order to represent this problem as a sequence-to-sequence modeling task, the predicted output at timestep $t \in T_{\text{pred}}$ can only depend on inputs from $t' \in T_{\text{obs}}$. In other words, predictions cannot depend on future states of traffic participants. Moreover, we predict the future trajectories for an interval greater than or equal to the observation interval, as estimating the trajectories for an interval shorter than the observation interval is comparatively trivial. Instead we strive to accurately predict the future states of dynamic agents for an interval longer than the observation interval.

We propose the IA-TCNN architecture depicted in Figure 5.2(a), which fulfills the above criteria. Our network consists of three causal blocks, where each block contains zero-padding followed by n dilated causal convolutions, cropping and a tanh activation function. In each block, we employ zero padding and cropping layers to fulfill the requirement of predicting a trajectory with length greater than or equal to the observed trajectory. We utilize causal convolutions where the output at each timestep is convolved with elements from earlier timesteps, thereby preventing information leak across different layers.

Although the amount of previous information utilized by causal convolutions is linear to the network depth, increasing the depth or using extremely large filter sizes increases the inference time as well as the training complexity. We overcome this problem by employing dilated causal convolutions to increase the size of the receptive field without increasing the depth of the network. We use a constant kernel size of 30×30 for each of the convolutional layers with filter sizes of 128 each and increase the dilation rate by 1 for each following convolution. We model the predicted spatial coordinates and velocity of each traffic participant using a multivariate Gaussian distribution in order to obtain a measure of confidence over the output of the network. We do not include the yaw angle in the multivariate distribution due to wrap-around issues. Obtaining a confidence measure over the angular prediction can be, however, addressed by using a von Mises distribution. In this work, we adopt the approach of estimating the yaw angle separately using the quaternion components. The output of the last block is passed to a

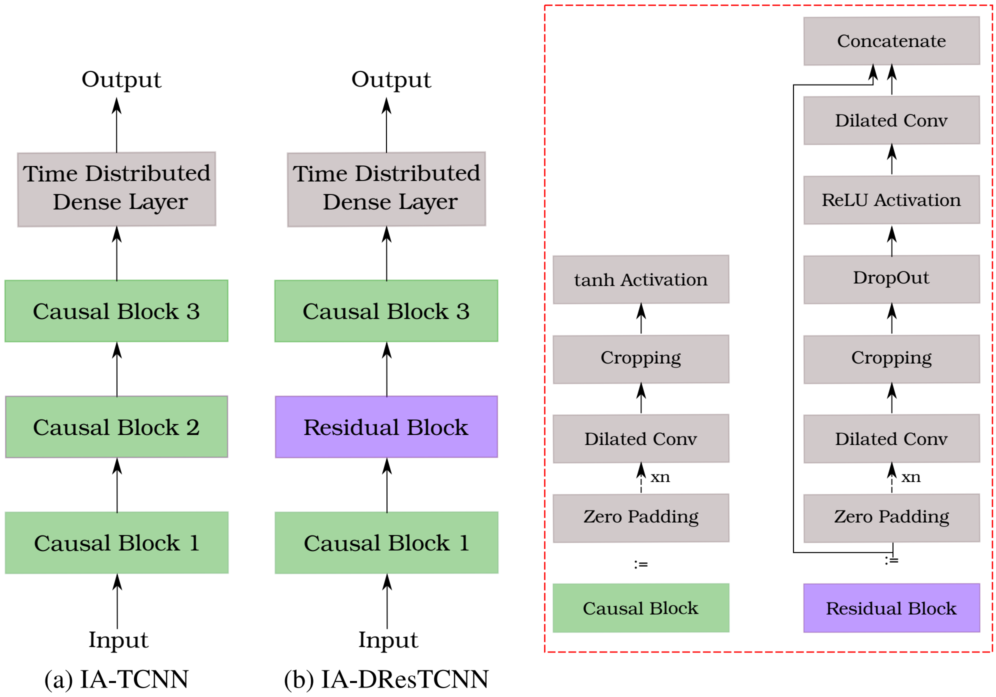


Figure 5.2: Illustration of the proposed network architecture for interaction-aware motion prediction. We propose two variants of our architecture (a)IA-TCNN and (b) IA-DResTCNN. The legend enclosed in red dashed lines shows the constituting layers for each block.

time-distributed fully connected layer of size 9 to produce temporal predictions for each timestep of the prediction interval, where for each dynamic object the network predicts the mean $\boldsymbol{\mu}_i^t = (\mu_x, \mu_y, \mu_v)_i^t$, standard deviation $\boldsymbol{\sigma}_i^t = (\sigma_x, \sigma_y, \sigma_v)_i^t$, correlation coefficient $\boldsymbol{\rho}_i^t = (\rho_{xy}, \rho_{xv}, \rho_{yv})_i^t$ and quaternion components (qw_i^t, qz_i^t).

We propose two variants of our method depicted in Figure 5.2 to further investigate the suitability of the proposed architecture for the sequence modeling task. IA-LinConv closely resembles the IA-TCNN architecture with the exception of setting the dilation rate $r = 1$ and the number of dilated convolutions $n = 1$, thus obtaining a single standard convolution per causal block. We propose this variant to investigate the effect of adding a dilation factor on improving the representational learning ability of the network. In the second variant IA-DResTCNN, we replace the middle causal block with a residual causal block and the tanh activation function with a ReLU [137]. By introducing residual connections in the network, we investigate if the current depth, filter size and dilation factor affect the stability of the architecture.

We train our model by minimizing the weighted combination of the negative log likelihood loss of the ground-truth position (x_i^t, y_i^t, v_i^t) under the predicted Gaussian distribution parameters $(\hat{\boldsymbol{\mu}}_i^t, \hat{\boldsymbol{\sigma}}_i^t, \hat{\boldsymbol{\rho}}_i^t)$ and the $\mathcal{L}_2$ loss of the orientation in normalized quaternion representation $\hat{\mathbf{q}}_i^t$ as follows:

$$\begin{aligned}
\mathcal{L}_\gamma(i,t) &= \left\| \left(\mathbf{q}_i^t\right)^{-1} \hat{\mathbf{q}}_i^t \right\|_2 \\
\mathcal{L}_p(i,t) &= -\log\left(P\left(x_i^t, y_i^t, v_i^t \mid \hat{\boldsymbol{\mu}}_i^t, \hat{\boldsymbol{\sigma}}_i^t, \hat{\boldsymbol{\rho}}_i^t\right)\right) \qquad (5.3) \\
\mathcal{L}_{MP}(i,t) &= \hat{s}_p + \hat{s}_\gamma + \sum_i^N \sum_t^{t_{\text{pred}}} \exp(-\hat{s}_p)\mathcal{L}_p(i,t) + \exp(-\hat{s}_\gamma)\mathcal{L}_\gamma(i,t),
\end{aligned}$$

where N is the number of dynamic agents, and $\hat{s}_p$, $\hat{s}_\gamma$ are learnable weighting variables for balancing the translational and rotational components of the predicted pose.

In real world data, the trajectories of different dynamic agents have varying lengths due to the limited sensor range which potentially restricts the amount of information available during training. In order to enable our method to leverage all trajectory information available, we train our proposed IA-TCNN with dynamic sequence lengths by using binary activation masks predicted by the network to signify the end of a trajectory. This in turn implicitly enables the network to learn when a pedestrian or vehicle exits the field-of-view of the sensor as well as enables it to learn a more realistic model of the trajectories. The predicted trajectory is then first multiplied by the activation mask before computing the prediction error. Moreover, as opposed to explicitly selecting the set of dynamic agents likely to affect the behavior of an agent, our proposed model utilizes information from all agents during the observation interval to predict the trajectory for each of the observed agents. This has the advantage of eliminating the need for creating handcrafted definitions which attempt to explicitly model how the behavior of a dynamic agent is affected by the surrounding agents. Furthermore, it expedites the information flow throughout the various layers of the network, hence facilitating fast trajectory estimation for all the dynamic agents in the scene.

5.2.2 Traffic Light Recognition

In this section, we describe the architecture of our traffic light recognition sub-network, which given an input RGB image I_t predicts the state of the traffic light $S_t \in S = \{\textit{Red}, \textit{Green}, \textit{Yellow}, \textit{No Light}\}$. We build upon the ResNet-50 architecture [82] with pre-activation residual units which allow unimpeded information flow throughout the network thus enabling the construction of more complex models that are easily trainable (Section 2.5.2). Our proposed AtteNet consists of five bottleneck residual blocks with multiple pre-activation residual units. We replace the traditional ReLU activation function in the residual unit with ELUs which enable our network to be more robust to noise in the training data while achieving shorter convergence time. Note that unlike the IA-DResTCNN architecture for interaction-aware motion prediction, we utilize the bottleneck

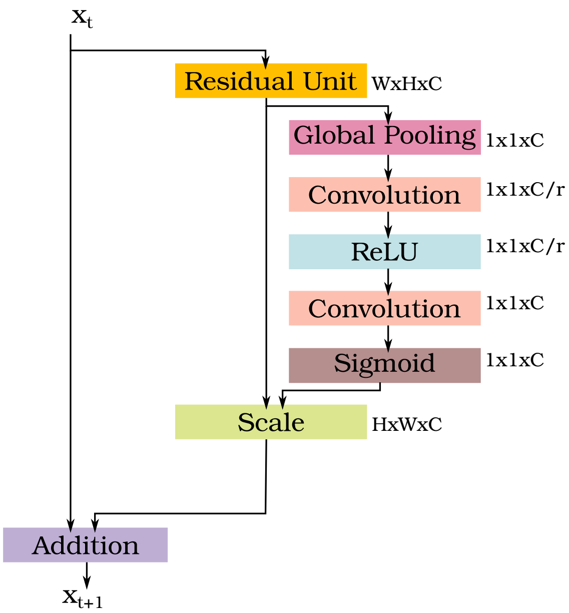


Figure 5.3: Illustration of a Squeeze-Excitation (SE) residual block. The output of the residual block is first passed through a squeeze operation to aggregate the feature maps across the spatial dimensions, then through an excitation operation to emphasize the informative features and suppress the irrelevant features.

residual units as the building block of our network due to their ability to aid in training deeper architectures without significantly increasing the number of parameters.

In order to improve the representational learning abilities of our network, we introduce Squeeze-Excitation (SE) blocks into our network [86]. Using SE blocks enables the network to perform feature recalibration, which in turn allows the network to utilize the global information in the images to selectively emphasize and suppress features depending on their usefulness for the task at hand. Instead of equally weighting all channels while creating the output feature maps, each SE block employs a content aware mechanism which learns an adaptive weighting for each channel with a minimum computational cost. A SE block, depicted in Figure 5.3 is comprised of two operations: squeeze and excite. During the squeeze operation feature maps from the previous layer are aggregated across the spatial dimension. Thus embedding the global distribution of the features to be leveraged by upcoming layers in the network. The excitation operation emphasizes the informative features and suppresses the irrelevant ones thus aiding in learning sample-specific activations for each channel. We replace the fully connected layers in the SE blocks with 1×1 convolutional layers and add a global average pooling layer after the fifth residual block, followed by a fully connected layer of size 4 which produces the prediction of the network. Our final architecture is shown in Figure 5.4. During training, we minimize the softmax cross entropy loss between the labels and the predicted logits.

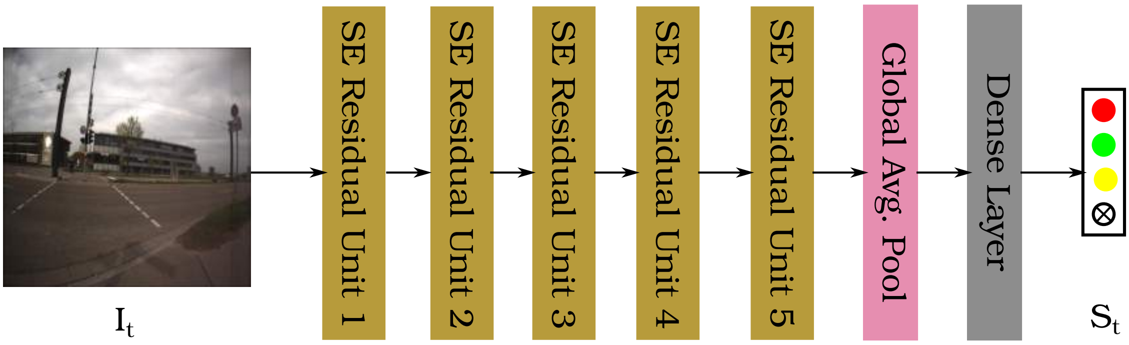


Figure 5.4: Schematic depiction of our proposed AtteNet architecture for traffic light recognition. Given an RGB image I_t, our network predicts the state of the traffic light S_t by aggregating the interdependencies between the different channels across the layers.

5.2.3 Learning To Cross The Road

In the following, we present the proposed method for learning to predict the safety of a street intersection for crossing. Before delving into the details of our approach, we first present a baseline approach to better analyze the performance of our proposed method.

5.2.3.1 Baseline

We formulate the problem of safe autonomous street crossing as a binary classification task. The input to the classifier is the sensor data from the most recent k-second interval, while the output is a binary value as to whether it is safe to cross the street. The trajectory $\mathcal{O}_i$ of each tracked dynamic object i is represented using the features stated in Equation (5.1), namely the spatial coordinates, velocity and yaw angle in the velocity direction for the length of the interval.

We create a feature vector for each time interval with a size of $m \times 5 \cdot k$, where m is the number of observed dynamic objects and k is the interval length. Dynamic objects are arranged in the feature vector with respect to their detection time, followed by distance to the robot with the closest object first. Under this representation, the final feature vector has the following format:

$$F = \begin{pmatrix} \mathcal{O}_1^{t1} & \mathcal{O}_1^{t2} & \mathcal{O}_1^{t3} & \cdots & \mathcal{O}_1^{tk} \\ \mathcal{O}_2^{t1} & \mathcal{O}_2^{t2} & \mathcal{O}_2^{t3} & \cdots & \mathcal{O}_2^{tk} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \mathcal{O}_m^{t1} & \mathcal{O}_m^{t2} & \mathcal{O}_m^{t3} & \cdots & \mathcal{O}_m^{tk} \end{pmatrix}$$

We pass each feature vector as a training/testing sample to the classifier, along with the label $\{$*Cross*, *Don't Cross*$\}$ representing the safety of the intersection for crossing. In order to learn the decision of when to safely cross the street intersection, we propose to utilize a Random Forest classifier [43] as well as a Naive Crossing Predictor as baseline approaches. The Naive Crossing Predictor iterates over all detected objects within an interval, and independent of their temporal behavior decides if it is safe to cross by utilizing the spatial coordinates of the object and its detected velocity to compute the time to collision assuming the velocity of the object remains constant. If the computed time is below a certain threshold for any of the objects throughout the time interval, the entire interval is considered unsafe.

5.2.3.2 Autonomous Street Crossing Predictor

In order to learn a crossing strategy that is robust to the type of intersection, we propose fusing the output predictions from the trajectory estimation sub-network and the traffic light recognition sub-network. Incorporating the traffic light recognition information is crucial at signalized intersections as the robot is expected to act within the behavioral norms obeying the intersection crossing rules such as crossing only when the traffic light is green. At the same time, in certain situations, one cannot rely solely on the traffic light information to cross such as when an ambulance or police car is speeding towards an intersection. In such cases, despite the green pedestrian traffic light, the robot is expected to wait at the sidewalk until the intersection becomes safe for crossing. Similarly at unsignalized intersections, the robot is expected to identify safe crossing intervals from unsafe intervals.

In these situations, utilizing the information from the trajectory estimation module is crucial to ensure safe crossing prediction. To achieve this goal, we perform element-wise concatenation of the feature maps from the traffic light recognition stream and the motion prediction stream. More specifically, the predicted Gaussian distribution parameters from IA-TCNN are first passed to a fully connected layer of dimension D, the output of which is reshaped to $H \times W \times C$ which corresponds in shape to the output of layer *Res5c* of AtteNet. Both tensors are then concatenated and fed to a fully connected layer *fc1* with N units. This is then followed by another fully connected layer *fc2* with softmax activation and two output units signalizing the intersection safety state $\{Cross, Don't\ Cross\}$.

Utilizing the Gaussian distribution parameters and orientation to model the trajectories, enables our model to incorporate the confidence information regarding the likelihood of the predictions, which in turn improves the robustness of our method to the prediction accuracy. We train the model by minimizing the softmax cross entropy loss function. In Section 5.4.5.2, we evaluate the impact of incorporating information from each of the streams and the number of units in *fc1* on the accuracy of the learned crossing decision.

5.3 Freiburg Street Crossing Dataset

We introduce a large-scale dataset captured at different intersections in Freiburg, Germany which we make publicly available* [156]. We captured the dataset using our robotic platform presented in Section 2.1. During capturing this dataset, we relied only on three laser scanners: the Velodyne HDL-32E, the tilting Hokuyo and the vertically mounted SICK scanner, in addition to two Delphi ESR radar sensors which are mounted to the left and right sides of the robot. In order to collect the data, we placed the robot on the side of the road facing the street, and recorded live traffic data from both sides of the road.

Figure 5.5 shows birds-eye-view images of the different intersections captured in this dataset. The dataset consists of tracked detections of cars, cyclists and pedestrians recorded at different intersections over the course of two weeks and it is divided into 18 different sequences containing approximately over $2{,}000$ tracked objects. Each object is identified by a unique track ID, spatial coordinates, velocity and orientation angle. Additionally we provide annotation information in the form of intervals indicating the safety of the intersection for crossing as well as annotations of the camera images regarding the state of the pedestrian traffic light signal $\{$*Red, Green, No Light*$\}$.

Annotating the data to indicate intervals in which the intersection is safe for crossing proved to be a challenging problem. First, the decision to cross or not must be made using only the information from this time interval without any knowledge of future or past intervals in order to prevent bias and ensure the generalizability of the behavior. Second, the time period for which an individual observes oncoming traffic before making a decision varies from one person to the other, rendering it difficult to assign a predetermined fixed value. In addition, depending on the traffic flow people often change their decision of crossing on the spot. Finally, different individuals have different crossing behaviors; in the same situation at an intersection, some might decide to cross while others choose a more conservative approach and wait for the next opportunity. Adding more difficulty to the problem, the crossing behavior varies within the same person depending on the type of intersection and the width of the street. These factors combined made the labeling procedure a rather tedious task, where we attempted to eliminate as much non-determinism as possible in order to enable our classifier to learn a meaningful classification strategy as close to human behavior as possible.

In order to facilitate the process of annotation, during capturing the dataset we mounted two GoPro cameras on the left and right sides of the robot above the radars which in addition to the Bumblebee camera captured images of oncoming traffic. We developed a graphical user interface that displayed the synchronized images from all three cameras to aid in labeling the data. Figure 5.6 presents a screenshot from the interface, where the user

*The dataset is publicly available at:
`http://aisdatasets.cs.uni-freiburg.de/streetcrossing/`

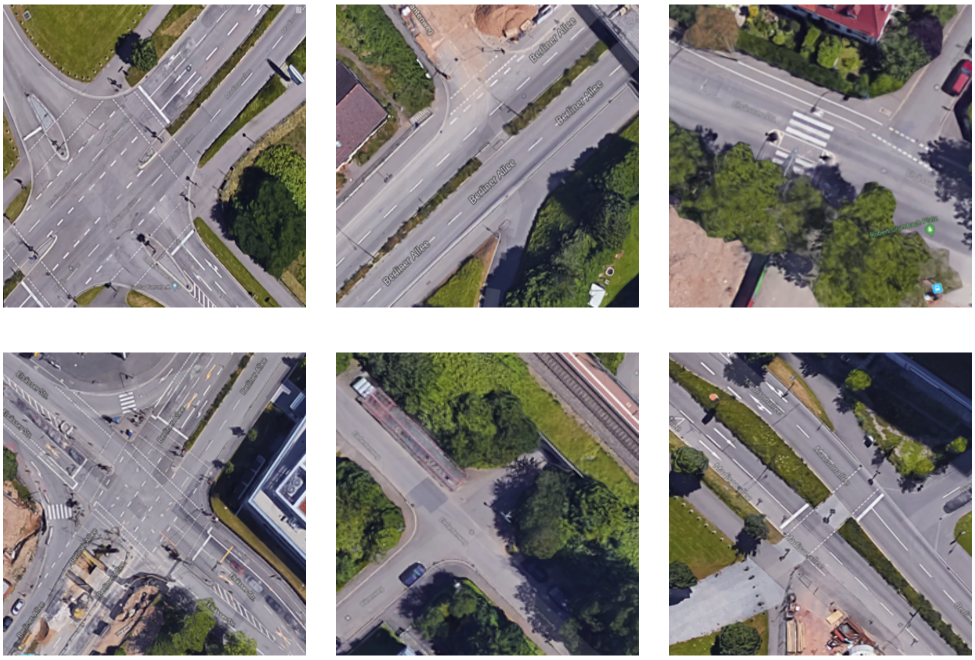


Figure 5.5: Birds-eye-view images from some of the road intersections included in the FSC dataset. The dataset covers a wide range of intersections including signalized, zebra crossings and crossings with middle islands, as well as different road curvature and streets merging.

can replay each interval several times before submitting the labeling decision. The replay button was added in order to accommodate for the short length of the decision interval and the randomness with which data samples are presented to the annotator. The data was labeled by five human annotators. For each data sample, the decision to cross is made at the end of the interval. In case of disagreement between the annotators, we chose majority voting over their decision. In order to quantify the agreement between the annotators, we employed Cohen's Kappa [57] which is one of the frequently used metrics for measuring inter-rater agreements. Within this metric, a value of 0 indicates no agreement between the raters, 1 means full agreement and negative values indicate agreement worse than random guessing. Using Cohen's Kappa, we found the inter-annotator agreement to be 0.47, showing moderate agreement between the annotators.

The Freiburg Street Crossing dataset can be used for intersection safety prediction for crossing, traffic light recognition or motion prediction tasks. Several factors make benchmarking on this dataset extremely challenging including large number of traffic participants, varying motion dependencies among different dynamic objects (Figure 5.8(d)), motion blur in the images, presence of reflecting surfaces and varying lighting conditions as shown in Figure 5.7(e,f). In order to train our network for the motion prediction task on this dataset, we apply a leave-one-out procedure by randomly selecting trajectories

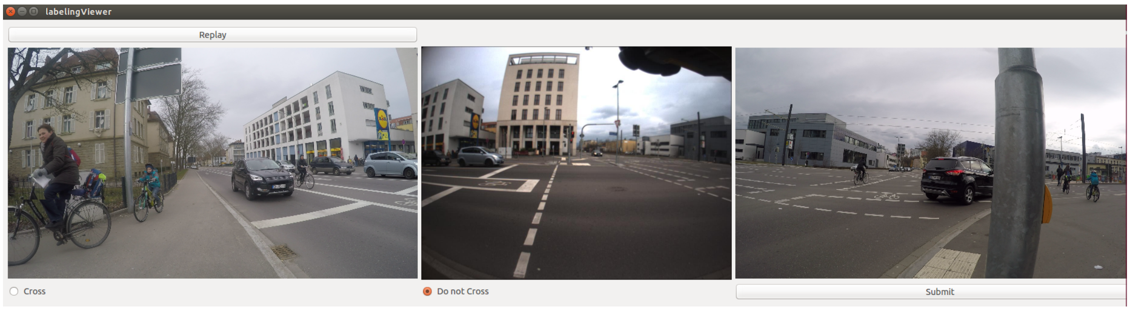


Figure 5.6: The graphical user interface used for labeling the FSC dataset. Each image shows the view from one of the cameras mounted on either the left, right or front side of the robot. Each sample interval can be replayed multiple times before making the decision of the safety of the intersection for crossing.

from all sequences except the testing sequence. For the traffic light recognition task, we divide the data into a $4{:}1$ split, and apply random brightness and contrast modulations as an augmentation procedure for the training images. Similarly, for the street intersection safety prediction task, we use 10 sequences for training and test on the remaining 8 sequences.

5.4 Experimental Evaluation

In order to evaluate our proposed system for predicting the safety of the intersection for crossing, we first evaluate each of the constituting subtasks followed by detailed results on the performance of the combined model. We evaluate our approach on multiple publicly available datasets and provide comprehensive details of our evaluation protocol to facilitate comparison and benchmarking. In the following section, we discuss in detail each of the datasets used for evaluation as well as any pre-processing or augmentation procedure applied.

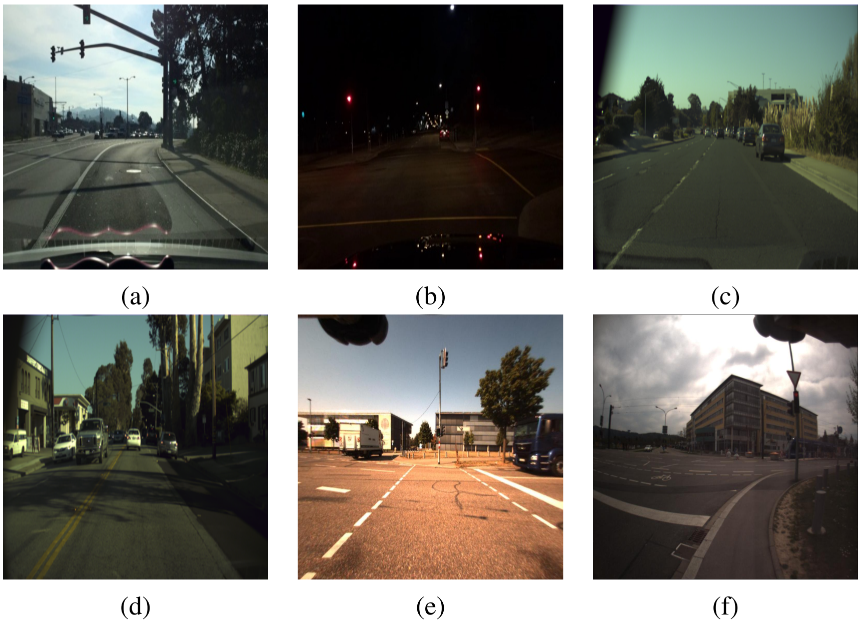
5.4.1 Datasets and Augmentation

Apart from the FSC dataset and the DeepLocCross dataset Section 4.3.3, we evaluate our approach on the following standard benchmarks:

Nexar Traffic Lights dataset consists of over $18{,}000$ RGB images captured in varying weather and lighting conditions. The dataset was released as part of a challenge to recognize the traffic light state in images taken by drivers using the Nexar app [142]. Each image is labeled with the state of the traffic light signal in the driving direction, between $\{$*Red, Green, No Light*$\}$. Several factors make benchmarking on this dataset extremely challenging such as the varying light conditions, the presence of substantial motion blur and the presence of multiple traffic lights in the image. Figure 5.7(a, b) shows sample images from the dataset. In addition to the aforementioned challenges, the evaluation criteria for this dataset was selected to be the classification accuracy and model size, with a minimum success criteria of 95.0% in terms of accuracy for submission acceptance. In order to train our method, we split the data into a training and a validation set using a split ratio of $4{:}1$ and perform augmentations on the training split in the form of random applications of brightness and contrast modulations.

Bosch Small Traffic Lights dataset contains RGB images at a resolution of $1{,}280 \times 720$ pixels captured in the San Francisco Bay Area and Palo Alto, California [35]. The training set consists of over $5{,}000$ images which are annotated at a 2 second interval, while the test set consists of over $8{,}000$ images annotated at a frame rate of 15 fps. Each image contains multiple labeled traffic lights amounting to a total of over $10{,}000$ annotated traffic lights in the training set and $13{,}000$ in the test set, with a median traffic light width of 8.6 pixels. For each image, the label file includes the bounding box coordinates of the traffic light, the status of the light $\{$*Red, Green, Yellow, No Light*$\}$, and whether the light is occluded by any object. This dataset is among the challenging benchmarks for detecting and recognizing traffic lights due to the small size of the lights in the image as well as the varying lighting conditions, presence of shadows and occlusions. We show example images from this dataset in Figure 5.7(c, d). We use the same training and test split provided by the authors and apply augmentations on the training set in the form of random brightness and contrast modulations. As our approach only predicts the status of the traffic light and not its position, we pre-process each image by masking out all but one traffic light using the bounding box information from the label file. To learn identifying when no traffic light is present in the image, we additionally mask out all the traffic lights, thereby creating from each image $N + 1$ images where N is the number of non-occluded traffic lights.

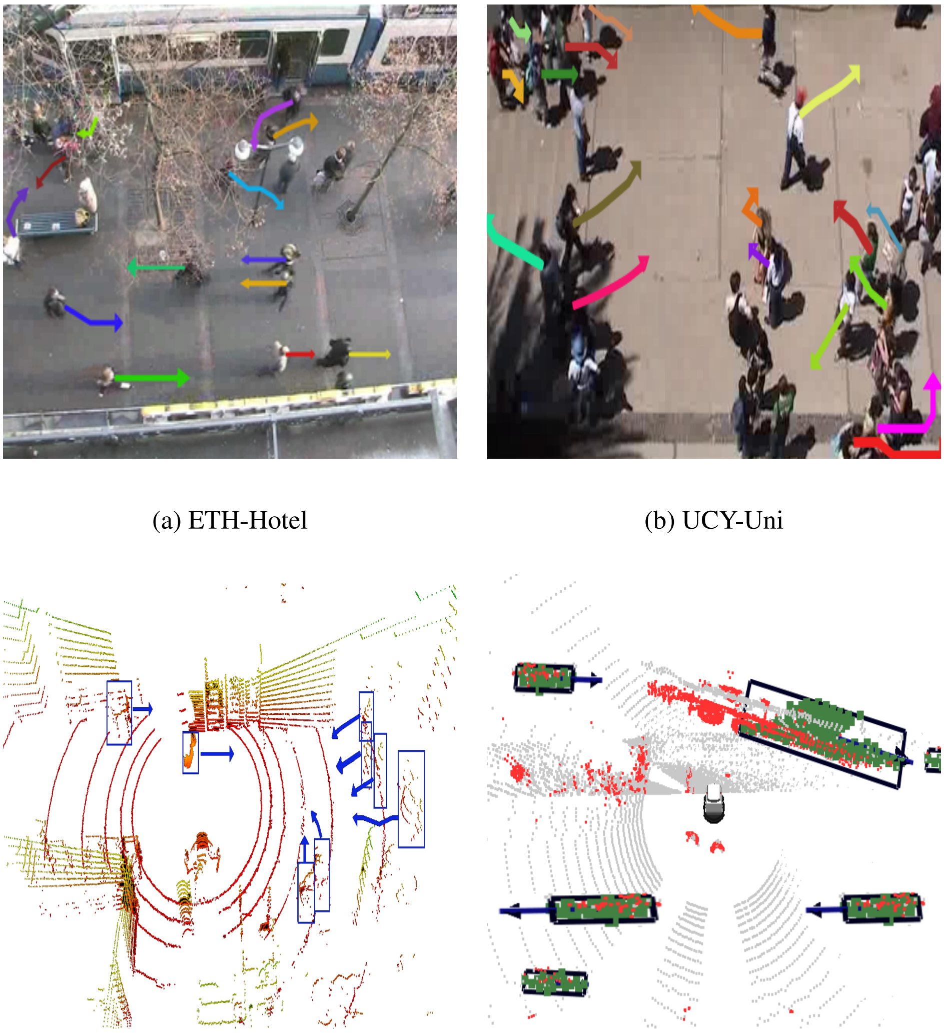
L-CAS dataset is a recently proposed benchmark for pedestrian motion prediction [204]. The data was captured using a 3D LiDAR scanner mounted on top of a Pioneer robot placed inside a university building. It consists of over 900 pedestrian tracks, each with an average length of 13.5 seconds and is divided into a training and testing split. Each pedestrian is identified by a unique ID, a time frame at which they are detected, the spatial coordinates, and their orientation angle. Some of the interesting scenarios captured in this dataset which make benchmarking challenging include people pushing trolleys, children running and groups dispersing. Figure 5.8(c) shows an example scan from the dataset, where pedestrians are marked by bounding boxes with arrows showing their trajectories for a sample interval. We use the same training and test splits provided by the authors for this dataset to facilitate comparison with other approaches.


(a) (b) (c)

(d) (e) (f)

Figure 5.7: Example images from the traffic light benchmark datasets used for evaluation namely: the Bosch Traffic Light dataset, the Nexar Challenge dataset and the FSC dataset. The datasets cover a wide range of challenges for the task of traffic light recognition including occlusions, varying lighting and weather conditions, presence of multiple traffic lights in the image and motion blur.

ETH crowd set dataset consists of two scenes: Univ and Hotel, containing a total of approximately 750 pedestrians exhibiting complex interactions [148]. For each scene, the dataset contains an obstacle map file providing the static map information of the surroundings, and an annotations file which provides the trajectory information for each pedestrian. Each tracked pedestrian is identified by a pedestrian ID, the frame number at which they were observed, the spatial coordinates and velocity with which they were traveling. The dataset additionally provides a groups file that provides information on pedestrians forming a group and a destinations file reporting the assumed destinations of all subjects in the scene. The dataset is one of the widely used benchmarks for pedestrian tracking and motion prediction as it represents real world crowded scenarios with multiple non linear trajectories, covering a wide range of group behavior such as crossings, dispersing and forming of groups. We show an example image from the Hotel sequence in Figure 5.8(a), where arrows represent the trajectories of the pedestrians for a sample interval. The sequence is recorded near a public transport stop. It captures the complex behavior of pedestrians as they enter/exit the vehicle as well as surrounding pedestrians navigating the scene. For this dataset, we utilize only the information from the annotations file, keeping track of the spatial coordinates of each pedestrian at each time frame. Furthermore, we assume no knowledge of the destination of each pedestrian, nor do



(a) ETH-Hotel

(b) UCY-Uni

(c) L-CAS

(d) Freiburg Street Crossing (FSC)

Figure 5.8: Sample trajectories from the various datasets employed for benchmarking the interaction-aware motion prediction sub-network. The benchmarking datasets include both camera captured sequences (ETH, UCY datasets) and LiDAR and radar captured sequences (L-CAS and FSC datasets). Overall, the datasets cover a wide range of motions among the various participants such as group behavior, trolley pushing and crowd navigation.

we utilize any information regarding group behavior or the structure of the environment.

UCY dataset consists of three scenes: Zara01, Zara02 and Uni, with a total of approximately 780 pedestrians [115]. For each scene, the dataset provides an annotations file comprised of a series of splines each describing the trajectory of a pedestrian using the spatial coordinates, frame number and the viewing direction of the pedestrian. This dataset in addition to the ETH dataset are widely used in conjunction as benchmarks for motion prediction and pedestrian tracking due to the wide range of non linear trajectories and pedestrian interactions exhibited including group behavior and pedestrians idling nearby shop fronts. Figure 5.8(b) shows a sample image from the Uni sequence, where pedestrian trajectories are represented by arrows. This particular sequence is the most challenging among the three sequences forming this dataset due to the large number of pedestrians observed concurrently, in addition to the complex crowd behavior demonstrated. We combine both this dataset with the ETH dataset similar to previous works [24, 197] and apply a leave-one-out procedure during training by randomly selecting trajectories from all scenes except the scenes used for testing. Furthermore, in order to facilitate the combination of the datasets, we predict only the 2D spatial coordinates for each pedestrian.

5.4.2 Training Schedule

In the following, we describe the training procedure used for each of the motion prediction, traffic light recognition and intersection safety prediction tasks. In order to train our IA-TCNN model such that it is robust to the varying number of pedestrians observable in each interval, we introduce a variable to represent the maximum number of distinct trajectories observed within an interval and initially set it to the maximum observed in all the datasets. During training and testing, we use an activation mask to encode the positions of valid trajectories and discard all remaining information. We train our model for 100 epochs with a mini-batch size of 12. We employ the Adam solver [98] for optimization, with a learning rate of 5×10^{-4} and apply gradient clipping. Details regarding the sequence length used for training, as well as the observation and prediction lengths used for testing are covered in Section 5.4.3.2.

We train our AtteNet model for traffic light recognition on random crops of size 224×224 and test on the center crop which we found adds more regularization to the network and helps learning a more generalized model. We use the SGD solver with momentum to optimize our AtteNet model, using an initial learning rate of 4×10^{-3} and a polynomial weight decay of 2×10^{-4}. We train our approach for 100 epochs using a mini-batch size of 32 and dropout probability of 0.2. In order to learn the final model for predicting the intersection safety for crossing, we initially bootstrap the training of both IA-TCNN and AtteNet using transfer learning from each of the aforementioned optimization procedures. We combine each of the task-specific loss functions using learnable weighting parameters and use a single optimizer to train all sub-networks concurrently. Training all tasks jointly aims at finding optimal weights that satisfy the constraints of each task as well as their interdependencies. Moreover, employing learnable weighting parameters ensures the proper balancing between the distinct tasks. We set the number of units in *fc1* of the Autonomous Street Crossing Predictor to 512. We employ the Adam optimizer with an initial learning rate of 5×10^{-5}. The final model is trained for 100 epochs with a mini-batch size of 10. In order to determine the parameters of the Random Forest Classifier baseline, we evaluated different configurations using a leave-one-out cross validation approach on the training data and opted for a maximum tree depth of 100, a minimum sample size of 50 and an active variable size of 100. As for the Naive Crossing Predictor baseline classifier, the minimum time to collision threshold was set to the same value as the sequence length employed for training the motion prediction task. All experiments are conducted using the Tensorflow library [21] on a single Nvidia Titan X GPU.

5.4.3 Evaluation of the Motion Prediction

In this section, we present extensive experimental evaluation of our IA-TCNN on the motion prediction task. We provide quantitative results comparing the performance of our proposed architecture with state-of-the-art methods on multiple publicly available datasets. In the following section, we provide a qualitative analysis of the predicted trajectories in addition to ablation studies on the impact of the various parameters on the accuracy of the predicted trajectories.

5.4.3.1 Comparison with the State of the Art

We benchmark the performance of our approach against several state-of-the-art methods for motion prediction including Social-LSTM [24], Social-Attention [197], Pose-LSTM [179], SGAN [77] and SoPhie [166]. Furthermore, we compare against the Social Forces model [84] and a basic LSTM implementation as baselines. Note that for each of the methods, we report the numbers directly from the corresponding manuscripts, with the exception of the Social Forces model for which we report the numbers from [24] as the original manuscript does not report evaluations using the metrics employed by the aforementioned methods. Furthermore, we use our own implementation for the LSTM baseline. We evaluate the accuracy of our motion prediction model by reporting the following metrics:

- *Average Displacement Error*: mean squared error over all predicted and ground-truth points in the trajectory.
- *Final Displacement Error*: distance between the predicted and ground-truth poses at the end of the prediction interval.

Table 5.1: Average displacement error of IA-TCNN on the task of motion prediction in comparison to existing methods on the L-CAS dataset.

Dataset	Social-LSTM [24]	Pose-LSTM [179]	IA-LinConv	IA-DResTCNN	IA-TCNN (Ours)
L-CAS	1.19m, NAN	0.95m, 35.0°	0.34m, 23.8°	0.46m, 33.1°	**0.11**m, **21.7**°

On the L-CAS, ETH and UCY datasets, we follow the standard evaluation procedure [24, 179] of training using a sequence length of 20 frames and using observation and prediction lengths of 8 frames (3.2s) and 12 frames (4.8s) respectively during testing. Table 5.1 shows the average displacement error of our approach on the L-CAS dataset. As demonstrated by the results, both our proposed variants, IA-LinConv and IA-DResTCNN are able to outperform the standard recurrent-based approaches by 64.2% and 32.0% in the translational and rotational components respectively. This in turn corroborates the advantage of utilizing a causal convolutional architecture over the standard recurrent methods. Moreover, by utilizing our proposed IA-TCNN, we achieve an average displacement error of 0.11m and 21.7° further improving upon the results by 67.6% and 8.8% in translation and rotation respectively. The improvement over the results achieved by IA-LinConv is attributed to employing dilated convolutions which increase the size of the receptive field, thereby increasing the content captured in each layer. However, we observe that adding a residual block to our network to improve the feature discriminability, as in IA-DResTCNN, does not help in improving the prediction accuracy despite it being helpful for other sequence modeling tasks such as language modeling [29].

In Table 5.2, we present the average displacement error of our proposed methods in comparison to state-of-the-art approaches on different sequences from the ETH and UCY crowd set datasets. Due to the complexity of the pedestrian interactions demonstrated in this dataset, employing the IA-LinConv model does not yield significant improvement over recurrent-based approaches due to the small receptive field at each layer. By employing dilated convolutions in our IA-TCNN architecture, the network is able to better capture the interactions across the various pedestrians, thereby achieving an improvement of 29.6% in comparison to the previous state of the art. Similar to IA-LinConv, the accuracy of IA-DResTCNN is comparable to that of recurrent-based approaches. However, the convergence time of the network is 5-times more than IA-LinConv and IA-TCNN.

We report the final displacement error of our proposed IA-TCNN on the various sequences of the ETH and UCY datasets in Table 5.3. Despite the sparse amount of sequences available for each dataset and the complexity of the pedestrian interactions demonstrated, our method is able to achieve the lowest final displacement error on all sequences of the ETH and UCY datasets with an average improvement of 55.7%

Table 5.2: Average displacement error of IA-TCNN on the task of motion prediction in comparison to existing methods on the ETH and UCY datasets.

Dataset	Social Forces [84]	Basic LSTM	Social-LSTM [24]	Social-Attention [197]	SGAN [77]	SoPhie [166]	IA-LinConv	IA-DResTCNN	IA-TCNN (Ours)
ETH-Univ	0.41m	0.39m	0.50m	0.39m	0.81m	0.70m	0.27m	0.43m	**0.15**m
ETH-Hotel	0.25m	0.32m	0.11m	0.29m	0.72m	0.76m	0.28m	0.36m	0.16m
Zara01	0.40m	0.18m	0.22m	0.20m	0.34m	0.30m	0.34m	0.45m	**0.14**m
Zara02	0.40m	0.28m	0.25m	0.30m	0.42m	0.38m	0.38m	0.37m	**0.19**m
UCY-Uni	0.48m	0.30m	0.27m	0.33m	0.60m	0.54m	0.41m	0.36m	0.29m
Average	0.39m	0.29m	0.27m	0.30m	0.58m	0.54m	0.34m	0.39m	**0.19**m

Table 5.3: Final displacement error of IA-TCNN on the task of motion prediction in comparison to existing methods on the ETH and UCY datasets.

Dataset	Social Forces [84]	Basic LSTM	Social-LSTM [24]	Social-Attention [197]	SGAN [77]	SoPhie [166]	IA-LinConv	IA-DResTCNN	IA-TCNN (Ours)
ETH-Univ	0.59m	1.06m	1.07m	3.74m	1.52m	1.43m	0.27m	0.60m	**0.21**m
ETH-Hotel	0.37m	0.33m	0.23m	2.64m	1.61m	1.67m	0.32m	0.52m	**0.18**m
Zara01	0.60m	0.93m	0.48m	0.52m	0.69m	0.63m	0.54m	1.08m	**0.27**m
Zara02	0.68m	1.09m	0.50m	2.13m	0.84m	0.78m	0.47m	0.88m	**0.25**m
UCY-Uni	0.78m	1.25m	0.77m	3.92m	1.26m	1.24m	0.66m	1.03m	**0.46**m
Average	0.60m	0.93m	0.61m	2.59m	1.18m	1.15m	0.45m	0.82m	**0.27**m

in comparison to previous methods. It is worth noting that while other approaches incorporate information about nearby pedestrians or the surrounding environment to predict the trajectories, our proposed method is able to accurately infer the trajectories, surpassing the performance of state-of-the-art methods, without leveraging information about the structure of the scene or performing any pre/post-processing on the trajectory data.

We benchmark on the Freiburg Street Crossing (FSC) dataset largely due to the variety of motion trajectories and complex interactions. Furthermore, unlike the ETH, UCY and L-CAS datasets, the FSC dataset includes trajectories and interactions among various types of dynamic objects such as cyclists, vehicles and pedestrians which in turn both increases the difficulty of the prediction task, as well as renders the data more representative of real-world scenarios. On the FSC dataset, we train using a sequence length of $10\mathrm{s}$ and use observation and prediction lengths of $5\mathrm{s}$. As the radar sensor has a larger field-of-view than the LiDAR, and in order to observe the traffic participants in both sensors, we experimentally identified that a time window of $10\mathrm{s}$ is appropriate for correlating objects in both sensors on this dataset. We report the average displacement error of our proposed method in Table 5.4. The results show an improvement in employing IA-LinConv and IA-DResTCNN over the LSTM baseline, specifically in terms of rotation and velocity estimation. We attribute this to the increased complexity of the interactions demonstrated in this dataset, in addition to the presence of multiple types of dynamic objects which exhibit different interaction and motion behavior. Furthermore, we observe that by employing the IA-DResTCNN architecture, the rotational accuracy of the pose is further improved in comparison to the IA-LinConv architecture. We attribute this improvement partially to the bigger receptive field at each layer due to the dilation factor employed. The best performance is achieved by leveraging the proposed IA-TCNN architecture which is able to balance the motion-specific pose components for each dynamic object independent of their type, yielding an average displacement error of $0.20\mathrm{m}$, 6.71° and $0.84\mathrm{m/s}$.

5.4.3.2 Ablation Study & Qualitative Evaluation

In this section, we perform detailed studies on the influences of various components of our proposed architecture. Table 5.5 shows the effect of varying the observation and prediction lengths on the average displacement error of IA-TCNN on the Uni sequence of the UCY crowd set dataset. For short observation lengths $(2-4)$ frames, the error in the predicted trajectory linearly increases with the increase in the prediction length, with the lowest error achieved using a prediction interval smaller than or equal to the observation interval. This accounts for the increased difficulty of making accurate predictions given short trajectory information as future interactions cannot be reliably predicted. Concurrently, by increasing the observation length, the prediction accuracy gradually increases with small improvements between $(6-8)$ observation frames. This can be attributed to the

Table 5.4: Average displacement error of our motion prediction network IA-TCNN on the Freiburg Street Crossing (FSC) dataset.

Dataset	Basic LSTM	IA-LinConv	IA-DResTCNN	IA-TCNN (Ours)
Seq.-1	0.37m, 10.34°, 0.62m/s	0.22m, 8.66°, 0.38m/s	0.49m, 6.25°, 0.48m/s	**0.21**m, 6.95°, 0.43m/s
Seq.-2	0.28m, 13.15°, 0.74m/s	0.48m, 9.69°, 0.55m/s	0.64m, 8.57°, 0.68m/s	**0.28**m, **8.51**°, **0.41**m/s
Seq.-3	0.37m, 20.68°, 0.86m/s	0.46m, 14.24°, 0.80m/s	0.85m, 12.46°, 1.01m/s	0.48m, 13.43°, 0.91m/s
Seq.-4	0.27m, 16.54°, 0.85m/s	0.26m, 6.56°, 0.63m/s	0.46m, 5.87°, 0.73m/s	**0.26**m, **4.52**°, **0.60**m/s
Seq.-5	0.32m, 8.17°, 1.26m/s	0.21m, 7.40°, 0.40m/s	0.42m, 7.20°, 0.48m/s	**0.21**m, **6.83**°, 0.45m/s
Seq.-6	0.26m, 20.53°, 1.98m/s	0.38m, 16.37°, 2.26m/s	0.65m, 8.23°, 2.27m/s	0.31m, 16.72°, 2.08m/s
Seq.-7	0.24m, 3.29°, 0.69m/s	0.09m, 2.00°, 0.28m/s	0.11m, 1.58°, 0.29m/s	**0.05**m, **1.43**°, **0.23**m/s
Seq.-8	0.35m, 6.56°, 1.23m/s	0.16m, 5.90°, 1.08m/s	0.40m, 2.92°, 1.34m/s	0.21m, 3.60°, **1.06**m/s
Seq.-9	0.31m, 7.65°, 1.10m/s	0.16m, 8.57°, 1.06m/s	0.38m, 2.70°, 1.35m/s	0.18m, 3.19°, **1.00**m/s
Seq.-10	0.37m, 6.55°, 0.90m/s	0.25m, 2.80°, 0.98m/s	0.28m, 2.50°, 1.00m/s	**0.14**m, **2.42**°, **0.78**m/s
Seq.-11	0.24m, 2.72°, 0.72m/s	0.24m, 2.03°, 0.57m/s	0.28m, 1.15°, 0.61m/s	**0.17**m, 1.80°, 0.64m/s
Seq.-12	0.28m, 4.96°, 1.14m/s	0.21m, 2.93°, 1.38m/s	0.35m, 1.94°, 1.38m/s	**0.19**m, 2.94°, 1.21m/s
Seq.-13	0.30m, 8.51°, 0.79m/s	0.32m, 7.75°, 1.85m/s	0.60m, 5.90°, 2.14m/s	**0.28**m, 6.11°, 1.64m/s
Seq.-14	0.32m, 6.00°, 1.13m/s	0.25m, 4.92°, 0.69m/s	0.35m, 4.01°, 0.72m/s	**0.17**m, 4.04°, **0.53**m/s
Seq.-15	0.39m, 3.31°, 0.79m/s	0.19m, 2.34°, 0.60m/s	0.22m, 2.09°, 0.64m/s	**0.14**m, **1.64**°, **0.48**m/s
Seq.-16	0.34m, 14.69°, 1.51m/s	0.13m, 15.84°, 1.32m/s	0.34m, 16.99°, 1.30m/s	**0.10**m, **10.33**°, **1.20**m/s
Seq.-17	0.36m, 33.42°, 0.83m/s	0.17m, 23.06°, 0.73m/s	0.44m, 26.86°, 0.86m/s	**0.14**m, **13.69**°, 0.83m/s
Seq.-18	0.32m, 23.77°, 0.94m/s	0.12m, 16.15°, 0.92m/s	0.32m, 17.44°, 0.91m/s	**0.12**m, **12.62**°, **0.75**m/s
Average	0.32m, 11.71°, 0.95m/s	0.24m, 8.73°, 0.92m/s	0.42m, 7.48°, 1.01m/s	**0.20**m, **6.71**°, **0.84**m/s

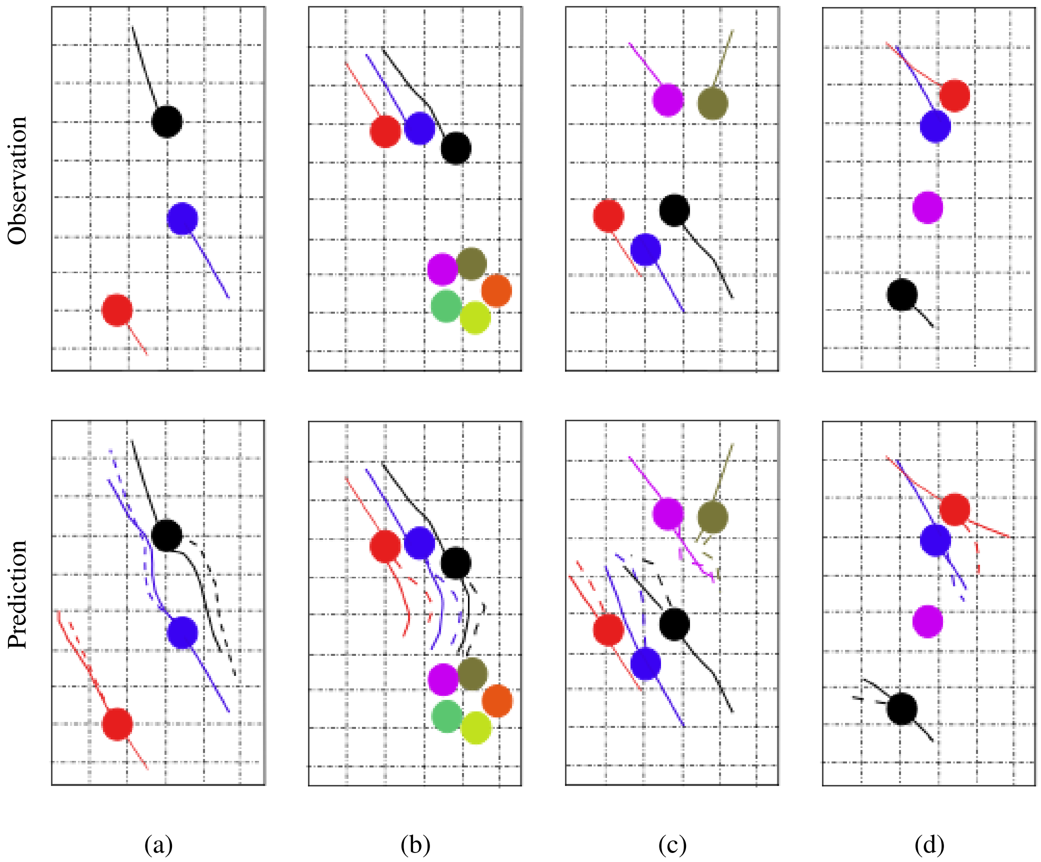


Figure 5.9: Qualitative analysis of our interaction-aware motion prediction IA-TCNN network on four example sequences from the UCY-UNI data. The top figures show the observed trajectories for all the pedestrians, along with the current pedestrian position marked by the colored circle. The bottom figures show the ground-truth (solid lines) and predicted (dashed lines) trajectories for each corresponding figure. Our approach is able to accurately model the pedestrian interactions and group behavior in each of the different scenarios presented.

reduction in the amount of significant information over time due to the short interaction times between the pedestrians and the low likelihood of abrupt changes in the behavior of one or more pedestrians.

We further evaluate the effect of the sequence length on the accuracy of our proposed IA-TCNN on the FSC dataset. Table 5.6 displays the average displacement error on Seq.-6 of the dataset, where we train our approach using sequence lengths between 10s and 35s. The results show that increasing the prediction length results in a decrease in the accuracy of the prediction which is consistent with the results in Table 5.5. The best accuracy is achieved using observation and prediction lengths of 5s as dynamic objects can be correlated better across the various sensors.

In order to evaluate the performance of our proposed model in various types of inter-

Table 5.5: Effect of the varying the observation and prediction lengths in frames on the average displacement error for our proposed IA-TCNN method on the task of motion prediction on the UCY-Uni dataset.

Pred. Length / Obs. Length	2	3	4	5	6	7	8	9	10	11	12
2	**0.42**m	0.48m	0.50m	0.53m	0.56m	0.61m	0.63m	0.62m	0.62m	0.61m	0.60m
3	**0.31**m	0.36m	0.41m	0.45m	0.49m	0.52m	0.52m	0.51m	0.51m	0.51m	0.51m
4	**0.23**m	0.30m	0.35m	0.39m	0.42m	0.43m	0.43m	0.43m	0.43m	0.43m	0.44m
5	**0.20**m	0.26m	0.31m	0.36m	0.36m	0.37m	0.37m	0.37m	0.38m	0.38m	0.38m
6	**0.18**m	0.23m	0.28m	0.29m	0.30m	0.32m	0.32m	0.33m	0.33m	0.33m	0.34m
7	**0.15**m	0.20m	0.22m	0.24m	0.26m	0.27m	0.28m	0.29m	0.29m	0.30m	0.31m
8	**0.14**m	0.17m	0.19m	0.21m	0.23m	0.24m	0.25m	0.26m	0.26m	0.27m	0.29m

Table 5.6: Evaluation of the effect of the sequence length on the average displacement error of the motion prediction on Seq.-6 of the Freiburg Street Crossing (FSC) dataset.

Pred. Length / Obs. Length	5	10	15	20
5	**0.31**m, **16.72**°, 2.08m/s	0.35m, 45.02°, 4.75m/s	0.56m, 44.21°, 4.04m/s	0.60m, 45.43°, 4.78m/s
10	-	0.32m, 20.53°, 1.35m/s	0.44m, 22.39°, 1.57m/s	0.58m, 24.54°, 2.01m/s
15	-	-	0.48m, 26.78°, 1.67m/s	0.47m, 30.63°, 1.80m/s

actions, we visualize four example scenes from the Uni sequence of the UCY dataset in Figure 5.9. The top row shows the observation sequence for each pedestrian represented by a solid line with the current position of the pedestrian depicted by a circle. In the bottom row, we plot the ground-truth trajectory (solid line) and the predicted trajectory of the network (dashed line). Figure 5.9(a) presents a scenario with collision avoidance for two individuals. In this scenario, our IA-TCNN method is able to predict the temporary change in direction for both individuals to avoid collision. Our proposed model is also able to represent group behavior as shown in Figure 5.9(b), where it predicts a common change of direction for all members of the group. Figure 5.9(c) shows a more complex scene with collision avoidance and overtaking maneuvers. The pedestrians depicted in red, blue and black display an example of the overtaking maneuver, where the red colored pedestrian is walking with a slightly lower velocity. Our model predicts that the blue colored pedestrian will adjust their trajectory to the right while increasing their velocity in order to overtake the red colored pedestrian. In order to avoid potential collision with the blue colored pedestrian, the model predicts a trajectory for the black colored pedestrian that is slightly deviated to the right. As for the purple and olive colored individuals, the model incorrectly predicts a trajectory where the olive colored pedestrian attempts to overtake the purple colored pedestrian. Whereas a more socially acceptable behavior, as shown by the ground-truth trajectory in this example would be to halt and wait for the purple colored pedestrian to pass.

Figure 5.9(d) shows another complex scenario with one static pedestrian in the middle, and a crossing maneuver between the red and blue colored pedestrians. In this example, our model predicts a trajectory where the red colored pedestrian follows the blue one. Note that although our approach incorporates the rotational information of the various dynamic objects into the prediction, we do not utilize the information about the heading of each individual on the ETH and UCY datasets, as this information is only available for one of the datasets, which further hinders it from being combined with the others. Nonetheless, we believe in such scenarios that information about the heading of each individual can significantly reduce the error in the predictions as shown by the results in Table 5.1, since sudden changes in direction are uncommon in the behavior of pedestrians.

We further compare the run-time and model size of our approach with various recurrent based approaches in Table 5.7. While the lowest average displacement error is achieved by the Social-LSTM approach [24], both the run-time and model size render it infeasible to be deployed in real-world scenarios. Similarly, while the SGAN method [77] achieves fast run-time, it has the lowest average and final displacement accuracies among all methods. The results show that using our proposed IA-TCNN, we improve upon the final displacement error by 40.3% while achieving analogous average displacement error in comparison with the best performing model with a competitive run-time of $0.06\mathrm{s}$ on a single NVIDIA Titan X GPU. Moreover, our model requires only $7.0\mathrm{MB}$ of storage space, thereby making it efficiently deployable in resource limited systems, while achieving

Table 5.7: Comparison of the performance of our interaction-aware motion prediction network with the state of the art on UCY-UNI.

Method	Avg. Disp. Error (m)	Final Disp. Error (m)	Run-time (s)	Size (MB)
Basic LSTM	0.30	1.25	0.29	6.1
Social-LSTM [24]	0.27	0.77	1.78	95.8
SGAN [77]	0.60	1.26	0.04	N/A
IA-TCNN (Ours)	0.29	$\mathbf{0.46}$	0.06	7.0

accurate predictions in an online manner.

5.4.4 Evaluation of the Traffic Light Recognition

In this section, we evaluate the efficacy of AtteNet for the task of traffic light recognition by benchmarking against convolutional neural network architectures designed for this task. Furthermore, we provide extensive ablation studies investigating the representations learned by the network as well as the various design choices and their subsequent effect on the recognition accuracy.

5.4.4.1 Comparison with the State of the art

We compare the performance of our AtteNet on the task of traffic light recognition with several network architectures tailored for the aforementioned task namely SqueezeNet [88], DenseNet [87] and ResNet [81]. We compare against the SqueezeNet architecture due to its relatively small size and high representational ability which enables it to be efficiently deployed in an online manner. This was demonstrated in the Nexar challenge where the first place winner used the SqueezeNet architecture achieving a recognition accuracy of 94.95%. Concurrently, we benchmark against DenseNet and ResNet architectures due to their top performance in various classification and regression tasks. We employ the ResNet-50 architecture with five residual blocks as a baseline. Similarly, we utilize the DenseNet-121 architecture with four dense blocks and a growth-rate of 16. We quantify the performance of each architecture by reporting the prediction accuracy, precision and recall rates. Table 5.8 shows the classification accuracy of AtteNet on all three datasets: Nexar, Bosch and FSC. AtteNet outperforms the best performing model (ResNet [81]) on all of the datasets by an average of 6.85% which in turn validates the suitability of our proposed architecture for the task of traffic light recognition. Furthermore, AtteNet is able to outperform the state of the art on the Nexar challenge dataset.

Analyzing the precision and recall rates for each class on the Nexar dataset in Table 5.9 shows that our proposed AtteNet is capable of accurately identifying the various traffic

Table 5.8: Comparison of classification accuracy of AtteNet with existing CNN traffic light recognition models.

Dataset	SqueezeNet [88]	DenseNet [87]	ResNet [81]	AtteNet (Ours)
Nexar	94.7%	91.5%	88.9%	**95.3%**
Bosch	62.9%	79.1%	80.9%	**82.9%**
FSC	76.1%	79.7%	86.5%	**91.8%**

Table 5.9: Comparison of precision and recall of AtteNet for traffic light recognition on the Nexar dataset.

Model	Precision			Recall		
	No Light	Red	Green	No Light	Red	Green
SqueezeNet [88]	91.8%	96.2%	94.4%	90.0%	96.0%	95.8%
DenseNet [87]	83.2%	93.2%	91.5%	88.7%	94.7%	87.9%
ResNet [81]	73.4%	92.6%	92.6%	90.0%	88.6%	84.2%
AtteNet (Ours)	**93.8%**	95.7%	**95.6%**	**90.3%**	**97.3%**	**95.8%**

light signals with the highest recall despite the challenging lighting conditions observed in this dataset. This is further corroborated in Figure 5.11(a) which plots the 3D t-Distributed Stochastic Neighbor Embedding (t-SNE) [196] of the features learned by our proposed AtteNet on the Nexar dataset in which data points belonging to the same traffic light category are distinctively clustered together. We discuss more about these plots in the ablation study presented in the following section.

Table 5.10 shows the precision and recall rates of our proposed AtteNet in comparison to the baseline approaches on the Bosch dataset. Unlike the Nexar dataset, the Bosch traffic lights dataset contains four categories for the traffic light signal by including a label for the yellow state. This in turn increases the difficulty of the task at hand as there only exists few labeled examples for the aforementioned class creating an imbalance in the distribution of the distinct classes. Nonetheless, our proposed approach is able to achieve comparable precision to the baseline variants and the highest recall rate. In Table 5.11, we present the precision and recall rates on the FSC dataset. Our proposed AtteNet architecture outperforms the baselines in terms of precision on each of the individual classes, while achieving high recall rates. This further corroborates the suitability of our proposed method for recognizing traffic lights in various conditions as shown in Figure 5.11(c) depicting the distribution of the learned features by our model in comparison to the baseline.

Table 5.10: Comparison of precision and recall of AtteNet for traffic light recognition on the Bosch dataset.

Model	Precision				Recall			
	No Light	Red	Green	Yellow	No Light	Red	Green	Yellow
SqueezeNet [88]	85.0%	77.4%	84.0%	56.0%	77.6%	80.5%	87.6%	15.9%
DenseNet [87]	76.1%	79.6%	80.2%	33.3%	79.4%	69.1%	88.9%	4.5%
ResNet [81]	80.3%	76.6%	87.5%	26.6%	77.0%	82.2%	86.9%	13.6%
AtteNet (Ours)	**85.2%**	**79.4%**	84.3%	55.2%	77.6%	**82.6%**	**89.2%**	**16.4%**

Table 5.11: Comparison of precision and recall of AtteNet for traffic light recognition on the Freiburg Street Crossing (FSC) dataset.

Model	Precision			Recall		
	No Light	Red	Green	No Light	Red	Green
SqueezeNet [88]	0.0%	98.5%	97.5%	0.0%	96.3%	97.0%
DenseNet [87]	55.2%	98.5%	91.1%	99.2%	69.5%	84.3%
ResNet [81]	71.3%	86.1%	86.3%	93.2%	84.6%	65.3%
AtteNet (Ours)	**75.4%**	**98.6%**	**99.7%**	**99.3%**	**96.8%**	71.6%

5.4.4.2 Ablation Study & Qualitative Analysis

In this section, we investigate the various architectural decisions made while designing AtteNet as well as present qualitative analysis of the obtained results on the benchmark datasets. In order to understand the design choices made in AtteNet, we compare the improvements gained by employing each of the following variants:

- ResNet: ResNet-50 base architecture
- AtteNet-M1: ResNet-50 base architecture with pre-activation residual units
- AtteNet-M2: ResNet-50 with pre-activation residual units and ELUs
- AtteNet-M3: ResNet-50 with SE blocks, pre-activation residual units and ELUs
- AtteNet-M4: ResNet-50 with 1×1 convolution SE blocks, pre-activation residual units and ELUs.

Table 5.12 reports the accuracy, precision and recall rates of the aforementioned variants on the Nexar dataset. We observe that the most notable improvement is achieved by replacing the traditional identity residual units with the pre-activation residual units, increasing the accuracy by 3.8%. This shows that utilizing the pre-activation residual

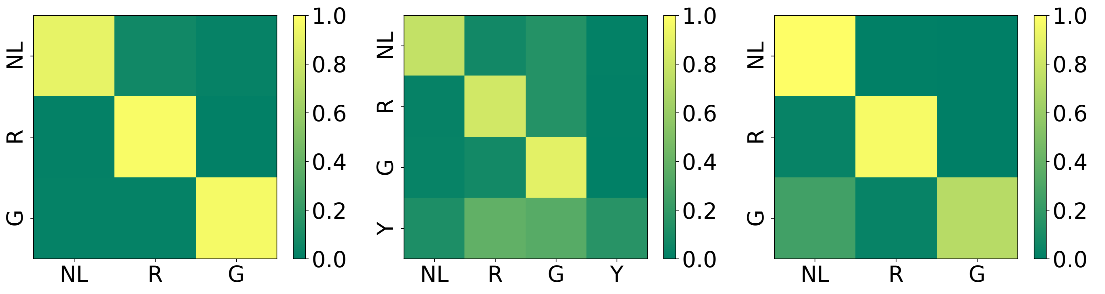
Table 5.12: Comparative analysis of AtteNet on the Nexar dataset for the task of traffic light recognition.

Model	Accuracy	Precision			Recall		
		No Light	Red	Green	No Light	Red	Green
ResNet	88.9%	73.4%	92.6%	92.6%	90.0%	88.6%	84.2%
AtteNet-M1	92.7%	86.3%	95.8%	92.7%	89.3%	94.3%	92.7%
AtteNet-M2	94.3%	92.0%	94.9%	95.2%	89.7%	96.1%	94.9%
AtteNet-M3	94.7%	90.7%	97.5%	94.2%	93.0%	94.9%	96.7%
AtteNet-M4	**95.3%**	**93.8%**	95.7%	**95.6%**	90.3%	**97.3%**	95.8%

units enables the network to better regularize the information flow which in turn leads to better representational learning. Replacing the traditional ReLU activation function for ELUs yields an additional 1.6% increase in the recognition accuracy which validates the importance of applying activation functions that are robust to noisy data. Incorporating SE blocks and replacing the fully connected layers with 1×1 convolutional layers further improves the recognition accuracy of the model. The results corroborate the significance of learning different weighting factors for the various channels of the feature maps. This in turn enables the network to learn the interdependencies between the channels, thereby improving the recognition capabilities as shown by the improved precision values.

Furthermore, we show the confusion matrix for AtteNet on the different datasets in Figure 5.10, where NL, R, G and Y stand for No Light, Red, Green and Yellow, respectively. On the Nexar dataset, our proposed architecture is able to accurately disambiguate the distinct classes as shown by the diagonal pattern of the confusion matrix. While on the bosch dataset shown in Figure 5.10(b), AtteNet is able to distinguish with high accuracy between three of the four classes with the yellow traffic light often misclassified as red or green. We believe this occurs as a result of the large imbalance in the distribution of the training data wherein the yellow traffic light occurs $6-10$-times less compared to the remaining classes. A potential remedy for this problem is to employ class balancing techniques, apply more augmentations to images belonging to this class, or by adding more images of the yellow class to the training set. Figure 5.10(c) shows the confusion matrix of AtteNet on the FSC dataset. The results indicate that our proposed AtteNet is able to accurately distinguish various classes further demonstrating the appropriateness of the architecture for the given task.

In order to gain a better understanding of the representations learned by the network, we employ the t-Distributed Stochastic Neighbor embedding (t-SNE) [196] on the learned features of the network. Through obtaining the set of principal components of the data, t-SNE is able to transform the data to a lower dimensional space, thereby revealing cluster and sub-cluster structures. In Figure 5.11, we display the down-projected features obtained after applying t-SNE on the features from the penultimate layer of AtteNet and



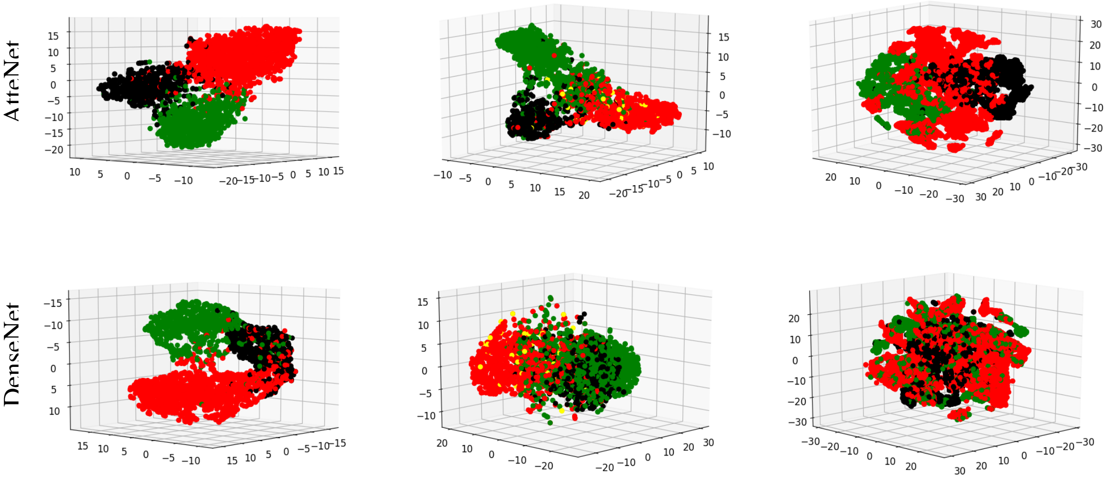
(a) Nexar (b) Bosch (c) FSC

Figure 5.10: Confusion matrix for our proposed AtteNet on the different datasets for traffic light recognition, where NL, R, G and Y stand for No Light, Red, Green and Yellow, respectively. Using our proposed architecture, we are able to accurately disambiguate the distinct traffic light states despite the small size of the light in the image and the various illumination conditions.

DenseNet on the various datasets. Unlike DenseNet, the features learned in AtteNet show clear cluster patterns separating the different classes, whereas in DenseNet there is no clear distinction between the features learned for the various classes especially in the Bosch and FSC dataset shown in Figure 5.11(b-c). Examining the t-SNE results of AtteNet on the Bosch dataset shows three distinct clusters for the no light, red and green classes, with the yellow class falling in between the red and green cluster. Nonetheless, the representations learned by AtteNet are able to better capture the distinct classes in the dataset in comparison to DenseNet where all four classes are merged together in one cluster.

Furthermore, we perform qualitative analysis of the recognition accuracy of our proposed AtteNet on the Nexar dataset in Figure 5.12. Figure 5.12(b, d, f) show incorrect classifications by our method, where in Figure 5.12(b), green light reflected off a glass structure is misidentified for a green traffic light signal due to both the shape and position of the light matching the shape and potential placement of a traffic light. Similarly, in Figure 5.12(f), the green sign of the shop is misidentified as the traffic light resulting in an incorrect classification. In Figure 5.12(d), the lack of information identifying the driving direction of the car results in a misclassification as the network incorrectly identifies the left-most traffic light to be the relevant one. However, despite the significant motion blur and low lighting conditions, our proposed model is able to accurately predict the state of the traffic light as shown in Figure 5.12(a, c, e).

In Figure 5.13, we show qualitative results on the Bosch traffic light dataset. Figure 5.13(d,f) show misclassification examples where AtteNet incorrectly predicts no traffic light in the driving direction. In both cases this occurs due to the small size of the traffic light and the presence of partial occlusions such as a pole hiding part of the traffic light. In Figure 5.13(h) a yellow traffic light signal is incorrectly classified as red. We

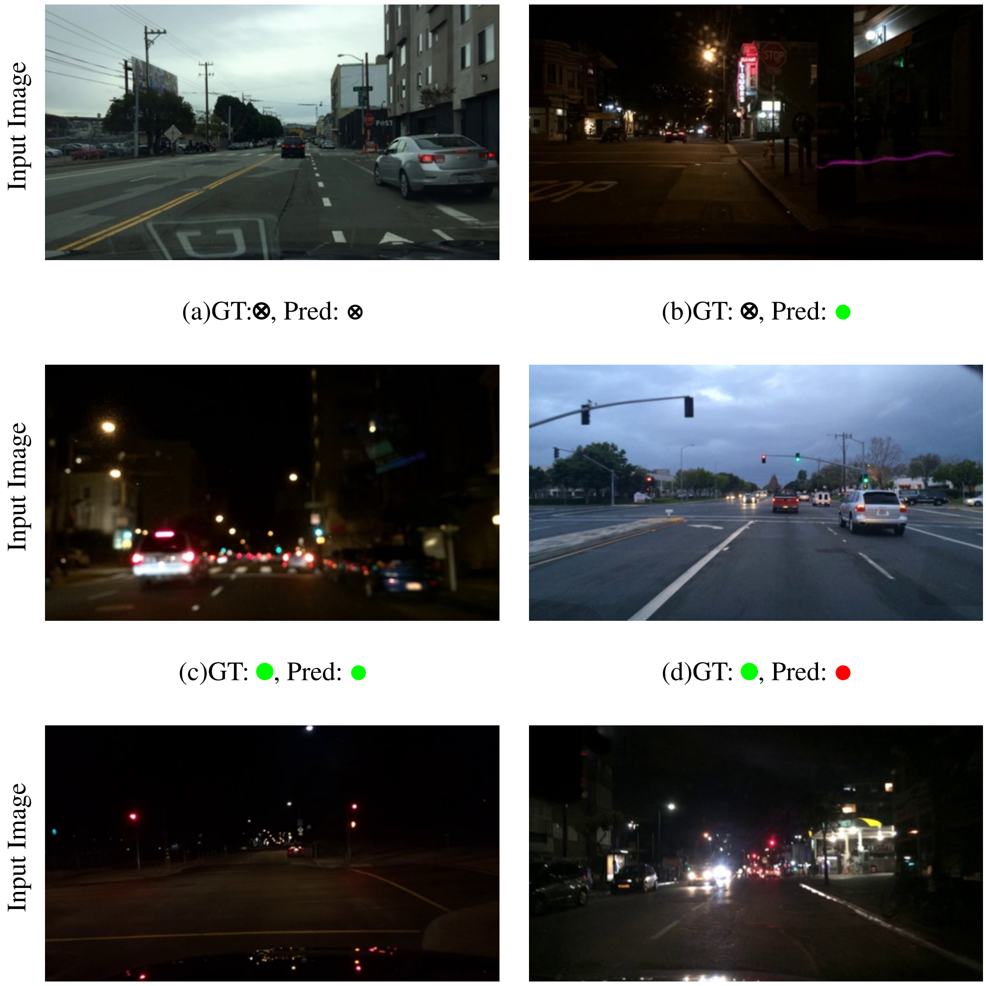


(a) Nexar (b) Bosch (c) FSC

Figure 5.11: Three dimensional t-Distributed Stochastic Neighbor Embedding (t-SNE) of features from the penultimate layer of our proposed AtteNet in comparison to DenseNet as a baseline trained on the various datasets for the task of traffic light recognition. The color of the points corresponds to the respective traffic light status, where black denotes no light. Features learned by AtteNet can better capture the distribution of the different classes in comparison to DenseNet where the clusters are not well separated.

attribute the cause of the misprediction to the close similarity of the red and yellow colors particularly in this image which can be verified by comparing the color of the brake lights of the cars to that of the traffic light signal. Figure 5.13(c) shows a correct classification of a green traffic light signal, where our proposed AtteNet is able to accurately recognize the traffic light signal despite the small size of the traffic light and the presence of partial occlusions. Similarly, in Figure 5.13(e, g), our network is able to accurately recognize the traffic light despite the presence of several surrounding traffic lights, the small size of the light and the presence of blur.

We present a qualitative evaluation of the performance of AtteNet on the FSC dataset in Figure 5.14. In Figure 5.14(b), the red and white pattern on the traffic light pole causes our network to incorrectly predict the presence of a red traffic light signal in spite of the absence of a traffic light within the image. Figure 5.14(a, c, e) show challenging scenarios in which AtteNet is able to accurately recognize the state of the traffic light while showing robustness towards the lighting conditions, presence of multiple light sources and the small size of the traffic light. Despite the achieved accuracy of the network, failing to recognize a traffic light as shown in Figure 5.14(d, f) will lead to unintended circumstances. While on the one hand, recognizing the traffic light in both images is quite challenging even



(a)GT:⊗, Pred: ⊗

(b)GT: ⊗, Pred: 🟢

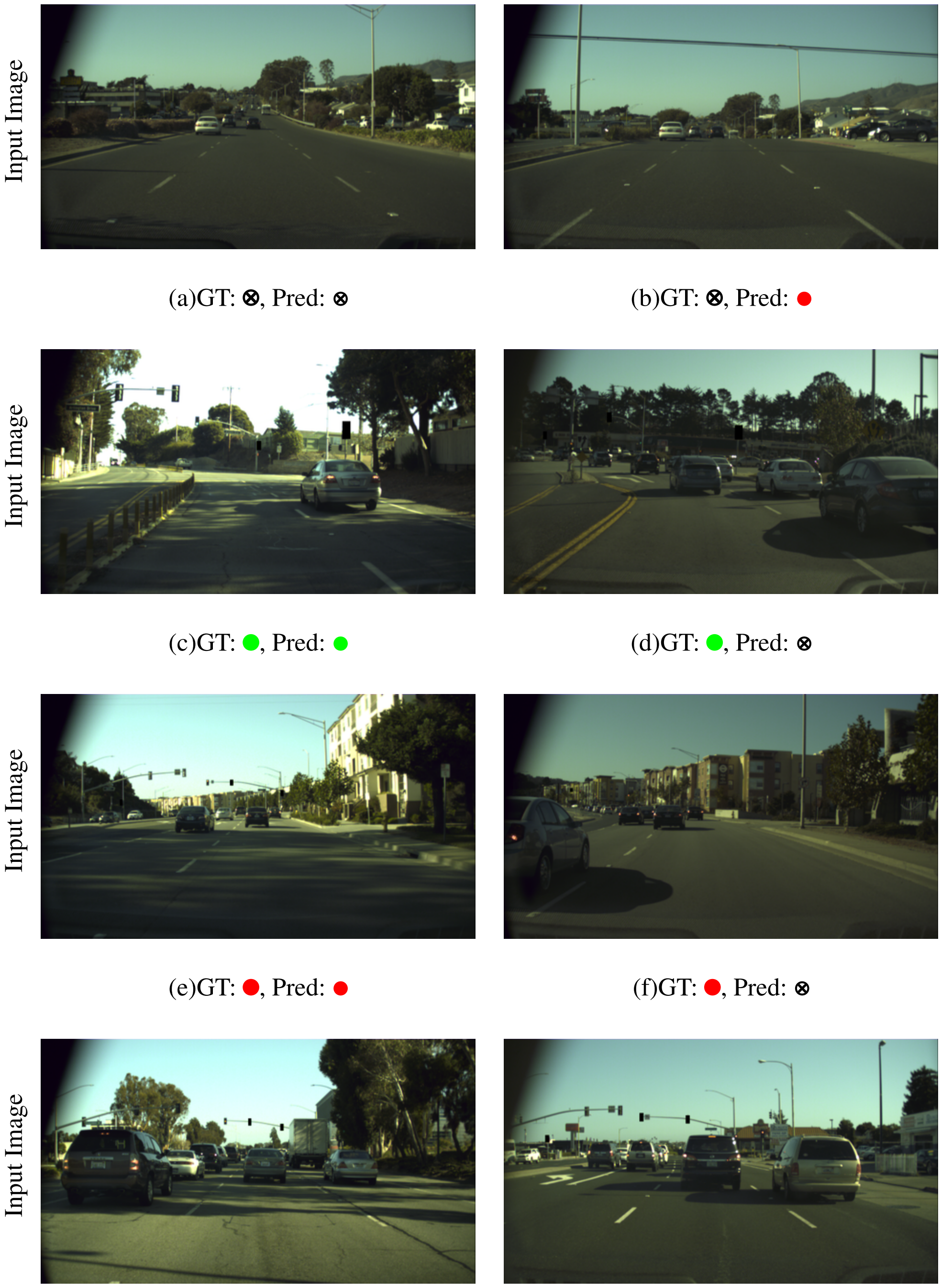
(c)GT: 🟢, Pred: 🟢

(d)GT: 🟢, Pred: 🔴

(e)GT: 🔴, Pred: 🔴

(f)GT: 🔴, Pred: 🟢

Figure 5.12: Qualitative evaluation of AtteNet predictions on the Nexar dataset for traffic light recognition. Below each image, we illustrate the ground-truth label (GT) and the network prediction (Pred). In images (c) and (e), the network is able to attend to the significant part of the image containing the traffic light and thus producing the correct prediction despite the small size of the light and the illumination noise from other sources of light. For the misclassified images, the attention of the network was placed on an incorrect area of the image (e.g. in image (f)), resulting in an incorrect prediction.



(a)GT: ⊗, Pred: ⊗

(b)GT: ⊗, Pred: 🔴

(c)GT: 🟢, Pred: 🟢

(d)GT: 🟢, Pred: ⊗

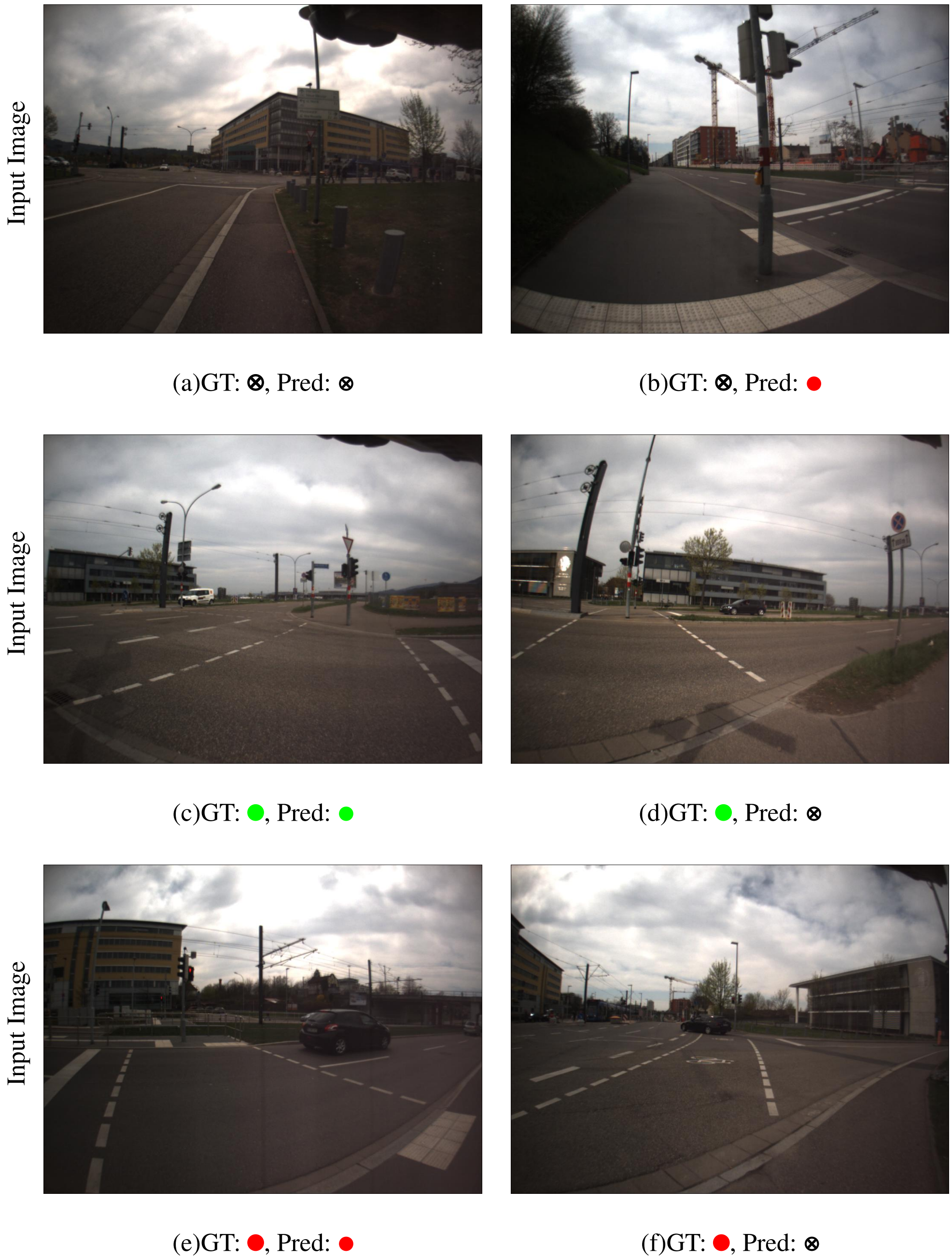
(e)GT: 🔴, Pred: 🔴

(f)GT: 🔴, Pred: ⊗

(g)GT: 🟡, Pred: 🟡

(h)GT: 🟡, Pred: 🔴

Figure 5.13: Qualitative evaluation of AtteNet on the Bosch Traffic Lights dataset. Images (a, c, e, g) illustrate correctly predicted examples, while images (b, d, f, h) illustrate misclassified predictions. Despite the small size of the traffic lights (fig. (c)), and the presence of multiple traffic lights in one image (fig. (e, g)), our proposed approach is able to accurately predict the state of the traffic light.



(a)GT: ⊗, Pred: ⊗

(b)GT: ⊗, Pred: 🔴

(c)GT: 🟢, Pred: 🟢

(d)GT: 🟢, Pred: ⊗

(e)GT: 🔴, Pred: 🔴

(f)GT: 🔴, Pred: ⊗

Figure 5.14: Qualitative analysis of the proposed traffic light recognition method on the Freiburg Street Crossing (FSC) dataset. Figures (a, c, e) illustrate correctly predicted images, while figures (b, d, f) illustrate mispredicted images. The small size of the traffic lights (fig. (c, f)), presence of noise sources (fig. (b, e)) and the varying illumination conditions (fig. (a, d)) under which the dataset was captured adds to the difficulty of the dataset for the given task. Despite these challenges, our approach is able to accurately recognize the state of the traffic light (fig. (a, c, e)).

for humans due to its small size in the image, one cannot rely solely on the traffic light recognition system to decide the safety of the intersection for crossing. Our proposed approach for autonomous street crossing prediction rather combines the information from both the traffic light recognition and motion prediction modules to accurately predict the intersection safety for crossing as discussed in the following section.

In Figure 5.15, we utilize Grad-CAM [173] to visualize the activation masks of AtteNet on the FSC dataset. Visualizing the output of the penultimate layer of our network using Grad-CAM produces a gradient-weighted class activation mask highlighting regions of the relevant regions in the image for predicting the output. For each image we show the activation mask, the ground-truth label and the network prediction. In Figure 5.15(a, c, e), the attention of the network is placed on areas of the image that contain the traffic light hence leading to correct predictions. Figure 5.15(b) shows an example image where a car crossing the intersection is occluding the traffic light. The activation mask shows that the attention of the network is incorrectly placed on the brake lights of the car which in turn lead to the incorrect prediction of a red traffic light signal. In Figure 5.15(d, f), the small size of the traffic light in the image increase the difficulty of locating and recognizing it as can be seen from the activation masks.

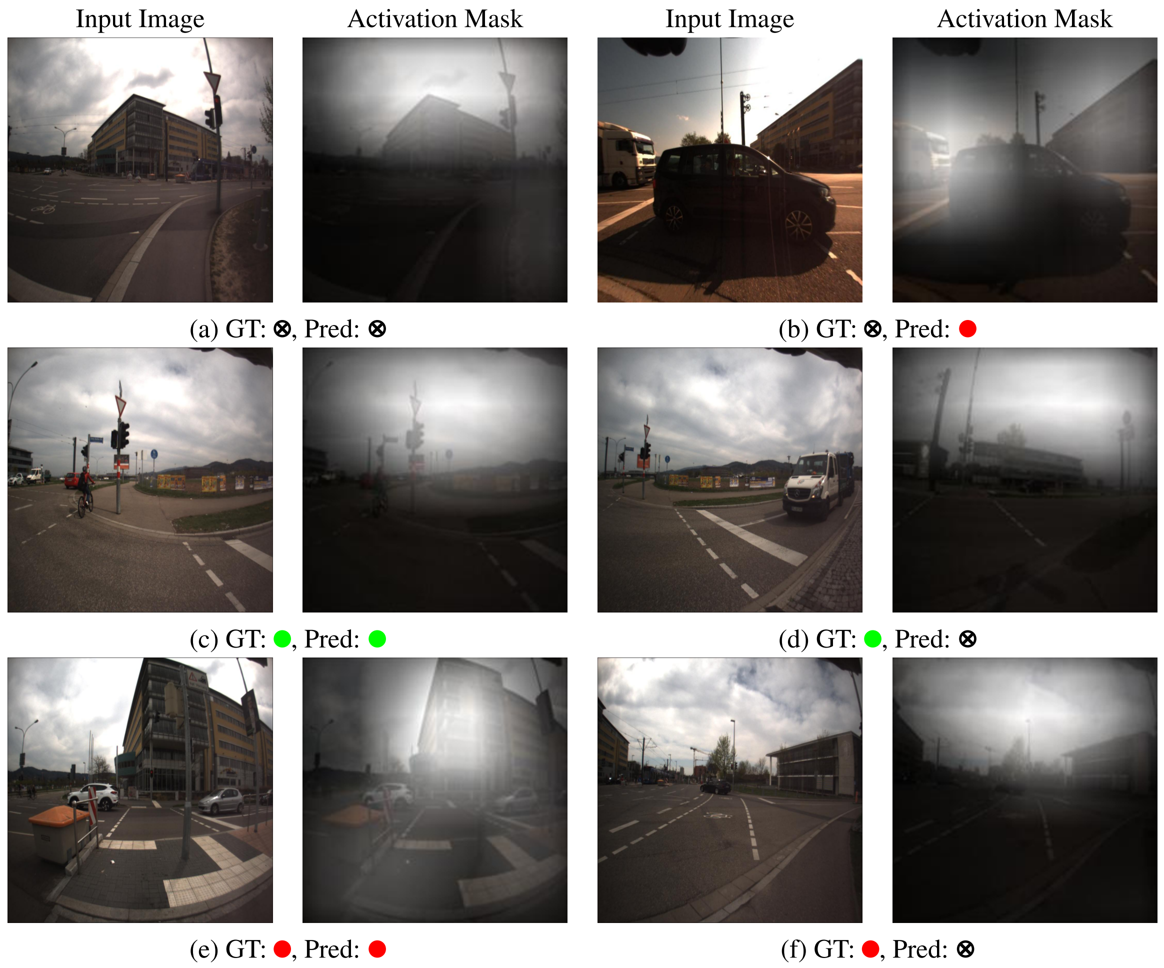
5.4.5 Evaluation of the Crossing Decision

In this section, we evaluate the efficacy of the proposed method for the task of predicting the safety of the street intersection for crossing. We first evaluate the baseline approaches addressing the problem as a binary classification task, followed by an extended evaluation of the proposed method of incorporating information from both traffic light recognition and motion prediction streams to learn the crossing decision.

5.4.5.1 Baseline

We evaluate the performance of the proposed Random Forest (RF) baseline classifier by comparing with Support Vector Machine (SVM) [83], k-Nearest Neighbor (kNN) [60] and the Naive Crossing Predictor (NCP). The parameters of both the SVM and kNN classifiers were chosen using the same procedure as the Random Forest classifier, namely leave-one-out cross validation on the training data. The best performance was obtained using a Sigmoid kernel with a C-value of 2.0 and a γ-value of 0.1 for the SVM classifier and a k-value of 8 for the kNN classifier proved to provide the best compromise between precision and recall. We train the RF classifier using the parameters stated in Section 5.4.2.

Figure 5.16 shows the confusion matrix for the different classifiers on the FSC dataset. The Random Forest classifier shows the best accuracy with the lowest number of false positives in comparison to all other classifiers. The confusion matrix of the SVM classifier shows that it favors labeling examples as not safe to cross over safe, which indicates that



(a) GT: ⊗, Pred: ⊗

(b) GT: ⊗, Pred: 🔴

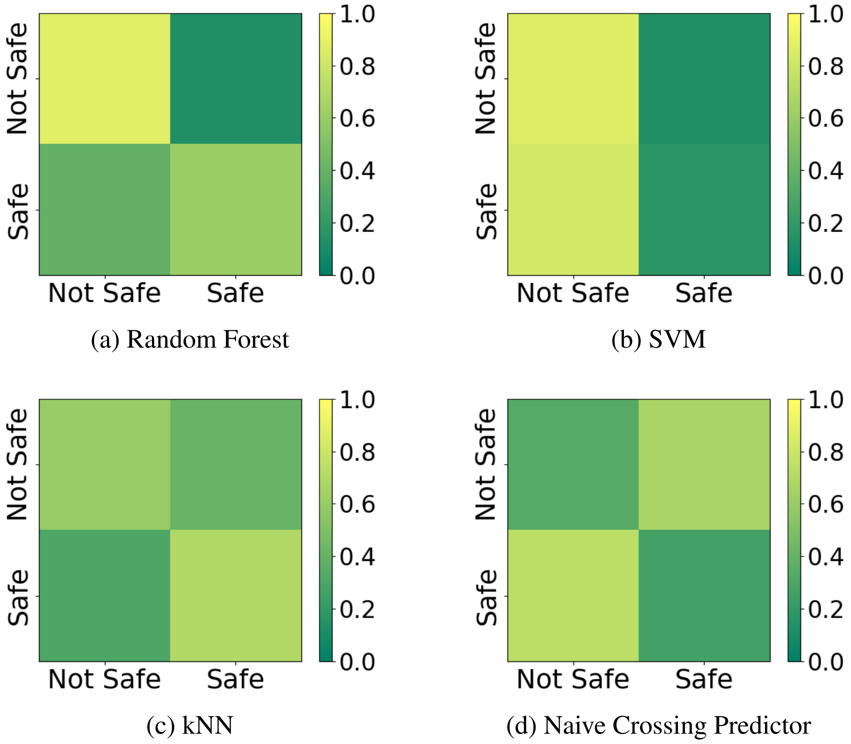
(c) GT: 🟢, Pred: 🟢

(d) GT: 🟢, Pred: ⊗

(e) GT: 🔴, Pred: 🔴

(f) GT: 🔴, Pred: ⊗

Figure 5.15: Visualization of the classification activation maps using Grad-CAM [173] of our AtteNet on the Freiburg Street Crossing (FSC) dataset. We overlay the activation masks on the image for ease of visualization. Below each image, we depict the ground-truth label (GT) and the network prediction (Pred).



(a) Random Forest

(b) SVM

(c) kNN

(d) Naive Crossing Predictor

Figure 5.16: Confusion matrix for the various binary classifiers on the Freiburg Street Crossing (FSC) dataset. The Random Forest (RF) classifier has the highest accuracy followed by the kNN classifier.

the learned classifier is more likely to wait for longer periods of time. The kNN classifier shows slightly better performance compared to the SVM, but with a higher number of mispredictions in comparison to the Random Forest. The Naive Crossing Predictor on the other hand shows the worst accuracy, consistently confusing both classes.

In order to measure the robustness of the Random Forest classifier to the type of intersection, we re-evaluate all classifiers by training on a subset of the dataset in which all encountered intersections are traffic light regulated, and test on a subset containing zebra crossings. This amounted to five logs for training and three for testing. It is worth noting that after the splitting was performed, the training data was observed to have the same class distribution as the entire dataset, while the test data has a class distribution with more negative examples. Figure 5.17 displays a bar plot of the precision and recall values for the trained classifiers. The Random Forest classifier shows the best generalization capabilities with a precision value of 98.6% and recall of 82.8%. The SVM classifier has a high recall value which we attribute to the unbalanced class distribution of the test set. The kNN classifier is able to generalize well to the current setup with a precision of 89.2% and a recall of 79.5%, while the Naive Crossing Predictor on the other hand, shows the worst performance with a precision value slightly better than random guessing. The

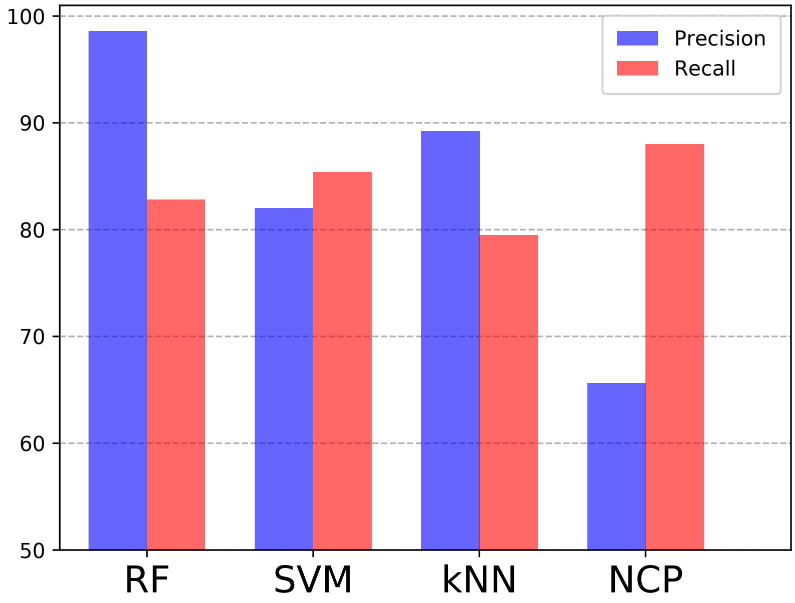


Figure 5.17: Bar plot showing the precision and recall values of the evaluated classifiers trained on a subset of the Freiburg Street Crossing (FSC) dataset with only traffic light regulated intersections and tested on zebra crossings. Note that the y-axis of the plot starts from 50 to better highlight the differences between the classifiers. The Random Forest (RF) classifier shows the best generalization capabilities with the highest combined precision and recall values, 98.6% and 82.8% respectively. The lowest precision is achieved by the Naive Crossing Predictor (NCP) with a value of 65.6%.

inferior performance of the Naive Crossing Predictor can be attributed to the field-of-view of the radars. Since there is no full overlap between the field-of-view of the radars and the Velodyne, blind spots exist. The learning-based classifier approaches are able to compensate for the presence of intermediate blind spots and hence can correctly predict the intersection safety in cases where the cars are temporarily in the blind spots. The Naive Crossing Predictor approach on the other hand, is unable to learn such a behavior. In the coming section we compare the performance of our proposed method for predicting the safety of the intersection for crossing with the Random Forest classifier.

5.4.5.2 Evaluation of the Autonomous Street Crossing Predictor

We evaluate the performance of our proposed Autonomous Road Crossing Predictor (ARCP) by reporting the accuracy, precision and recall rates on the FSC dataset. We compare the performance of our approach with the Random Forest classifier. Furthermore, in order to evaluate the tolerance of the learned predictor to mispredictions and noise from the information source, we also report the individual performance of our proposed ARCP by utilizing data from either the traffic light recognition module ARCP(TLR) or the

Table 5.13: Comparative analysis of the learned crossing decision on the Freiburg Street Crossing (FSC) dataset.

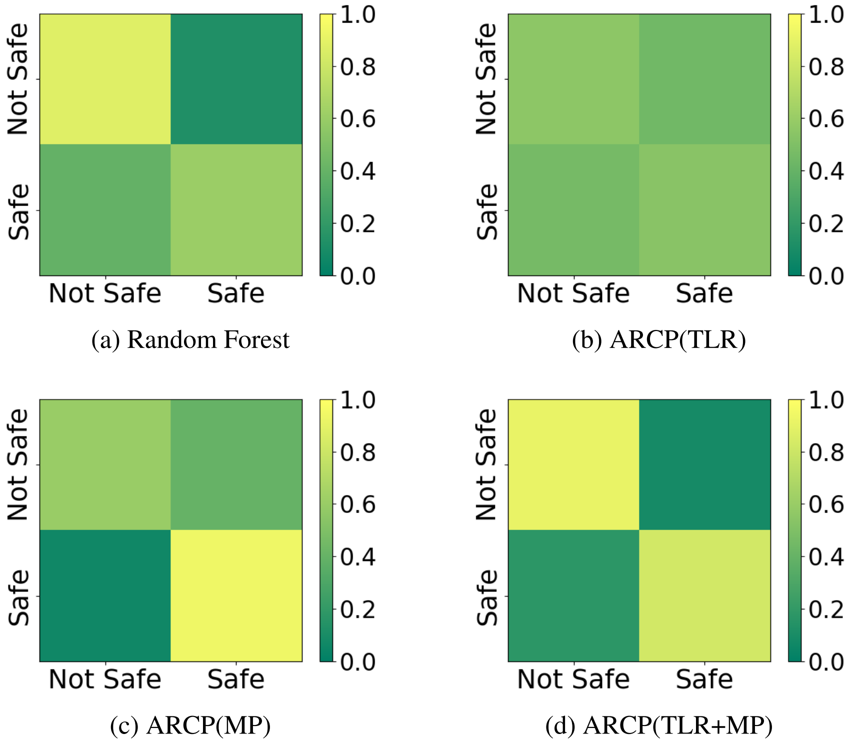
Method	Precision	Recall	Accuracy
Random Forest	78.8%	60.7%	75.4%
ARCP(TLR)	53.1%	61.3%	54.6%
ARCP(MP)	87.8%	93.5%	85.4%
ARCP(TLR+MP)	**91.9**%	82.3%	**86.2**%

motion prediction module ARCP(MP) or their combination ARCP(TLR+MP). Table 5.13 demonstrates the precision, recall and accuracy for each of the aforementioned methods. While the Random Forest classifier outperforms the binary classification baselines as shown in the previous section and as shown by the diagonal pattern of the confusion matrix in Figure 5.18(a), the precision, recall and accuracy of the classifier are critically low which would hinder its deployment in real-world environments.

Furthermore, despite the high accuracy of the proposed AtteNet for traffic light recognition, we observe that utilizing only information from this module, as in ARCP(TLR), results in minor improvement in the accuracy over random guessing. We attribute this to the difficulty of accurately predicting the intersection safety for crossing in the absence of a traffic light or in cases where the classifier fails to detect the presence of one, which is further demonstrated in Figure 5.18(b), where the confusion matrix does not show strong distinction between the various classes.

On the other hand, by employing information from only the motion prediction module, the overall accuracy of the crossing decision as well as the precision and recall are improved by $10.0\%, 9.0\%$ and 32.8% respectively. Unlike ARCP(TLR), the confusion matrix shown in Figure 5.18(c) shows that the learned classifier is able to better differentiate between safe and unsafe crossing intervals. However, comparing the top row of the confusion matrix of ARCP(MP) with that of the Random Forest baseline shows that in unsafe intervals, there is no clear distinction from the learned classifier between the selected decision, which can lead to suboptimal circumstances in deployment scenarios. This problem is, however, rectified in the ARCP(TLR+MP) classifier as shown in Figure 5.18(d). By combining both information from the traffic light recognition and the motion prediction modules, the classifier is agnostic to the type of intersection encountered. Furthermore, by incorporating feature maps from the last downsampling stage in AtteNet and the Gaussian distribution parameters of IA-TCNN, the learned classifier can better generalize to unseen environments as shown by the improvement in the accuracy, precision and recall rates over the Random Forest baseline classifier in Table 5.13.

We conducted experiments to determine the optimal number of units in the last fully



(a) Random Forest

(b) ARCP(TLR)

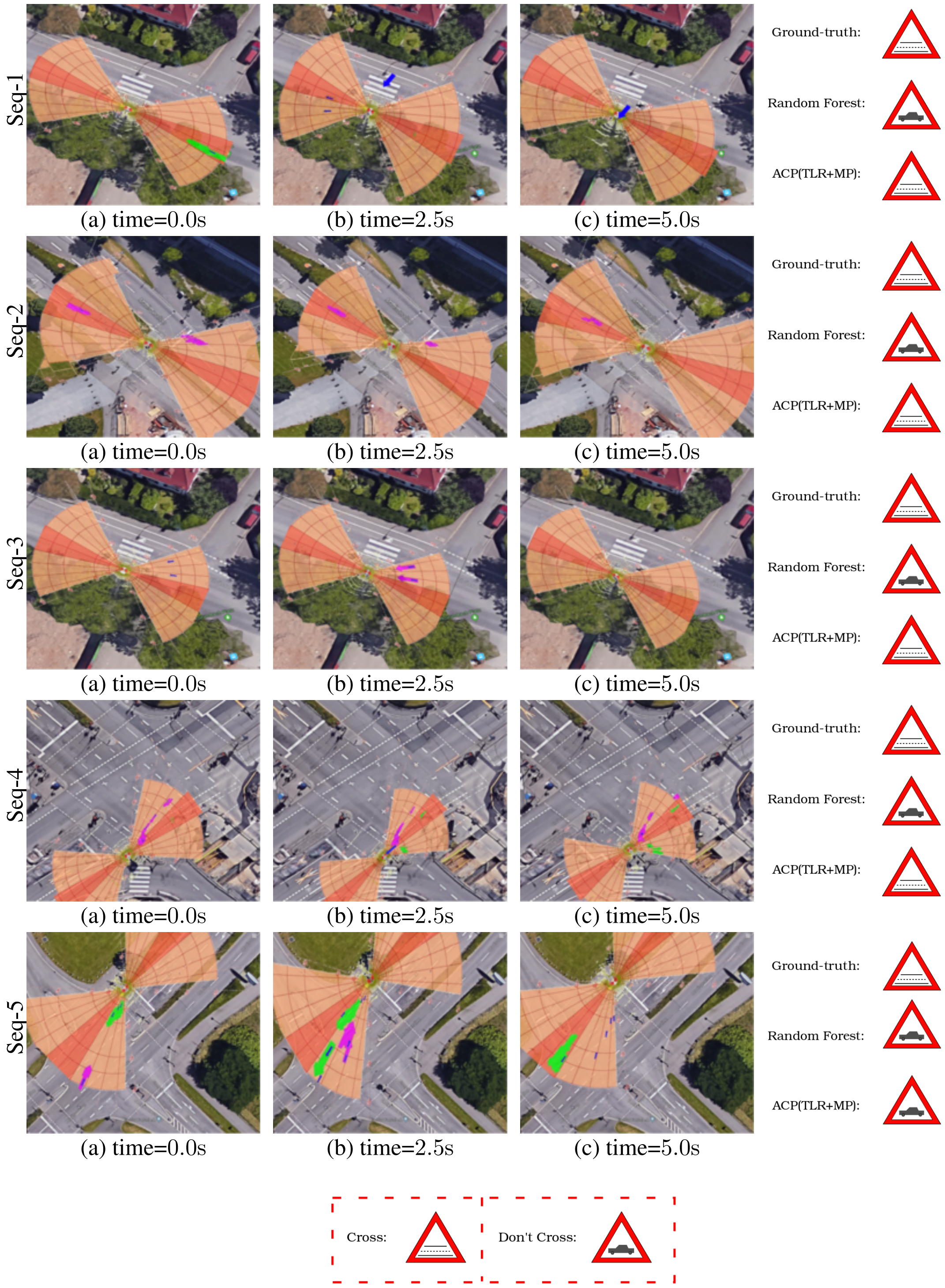
(c) ARCP(MP)

(d) ARCP(TLR+MP)

Figure 5.18: Confusion matrix of the various Autonomous Road Crossing Predictors (ARCP) in comparison to the baseline Random Forest Classifier. The best performance is achieved by utilizing both information from the Traffic Light Recognition (TLR) module and the Motion Prediction (MP) module to learn the crossing safety.

connected layer *fc1* of our ARCP method. Table 5.14 shows the precision, recall and accuracy using different values for the aforementioned parameter. We hypothesize that using a smaller number of units restricts the feature discrimination abilities of the network which is corroborated by the results, where using 128 units results in the lowest accuracy. However, we observe that by increasing the number of units to $1{,}024$, the recall increases by 0.2% over using 512 units, while the precision decreases by 17.7% and the accuracy decreases by 4.0%. The best compromise between precision, recall and accuracy is achieved by setting the number of units to 512.

In Figure 5.19, we perform qualitative analysis of the learned decision of our proposed ARCP(TLR+MP) classifier in comparison to the Random Forest classifier as a baseline on the FSC dataset. Each sequence is represented by three images corresponding to the beginning, middle and end of the interval. Furthermore, we overlay the sensor detections on birds-eye-view images of the intersections for ease of visualization. Detected dynamic objects are represented by arrows, where magenta arrows signify objects moving towards the robot, and green arrows signify objects moving away from the robot. Furthermore, the size of the arrow grows proportionally with the detected velocity. Seq-1 depicts a situation



(a) time=0.0s (b) time=2.5s (c) time=5.0s

(a) time=0.0s (b) time=2.5s (c) time=5.0s

(a) time=0.0s (b) time=2.5s (c) time=5.0s

(a) time=0.0s (b) time=2.5s (c) time=5.0s

(a) time=0.0s (b) time=2.5s (c) time=5.0s

Figure 5.19: Depiction of various crossing scenarios from the Freiburg Street Crossing (FSC) dataset. For each sequence, we depict three timesteps corresponding to the beginning, middle and end of the interval. We overlay the sensor visualization on birds-eye-view images from the corresponding intersections. For each sequence, on the right-most column, we depict the ground-truth label versus the predictions of the Random Forest classifier as a baseline and our proposed ARCP(TLR+MP) classifier. The legend is shown enclosed in a red box.

Table 5.14: Effect of the number of units in the penultimate layer of our Autonomous Road Crossing Predictor (ARCP) on the accuracy of the crossing decision.

Number of Units	Precision	Recall	Accuracy
128	76.9%	63.5%	72.7%
512	**87.8%**	**93.5%**	**85.4%**
1024	70.1%	93.7%	81.4%

where the robot is located at the side of a zebra crossing that is clear, with the exception of a cyclist (represented by the blue arrow) that is moving towards the robot. Despite the intersection being safe for crossing, the Random Forest classifier incorrectly labels the interval as unsafe for crossing. On the other hand, the ARCP(TLR+MP) classifier correctly identifies the intersection state by utilizing the information from the motion prediction module to infer the driving direction of the cyclist.

In Seq-2, the robot is located in the middle island at a signalized intersection, with traffic coming from the left-hand side. As the pedestrian traffic light is green, the approaching vehicle slows down throughout the observed interval rendering the intersection safe for crossing. By utilizing information from the traffic light recognition module to detect the state of the traffic light, in combination with the motion prediction module to identify that the approaching vehicle is reducing its velocity, our ARCP(TLR+MP) classifier is able to correctly label the interval as safe for crossing.

Seq-3 demonstrates a situation where a false positive detection by the tracker causes an incorrect classification of the intersection as unsafe by the Random Forest classifier. Our proposed classifier is, however, able to correctly predict the safety of the intersection for crossing as it is able to identify the spurious detection as a false positive or a ghost detection by the tracker. Another scenario is depicted in Seq-4, where the robot is located at a grid-type signalized intersection, with vehicles approaching from the upper right corner and heading towards the street parallel to the robot. By utilizing only information from the tracker, the Random Forest classifier labels the crossing unsafe as it appears that the cars are approaching perpendicularly to the robot. However, by predicting the behavior of the vehicles for the remainder of the interval, our proposed ARCP(TLR+MP) classifier is able to correctly classify the safety of the interval for crossing.

Finally, Seq-5 depicts an interval for which both classifiers incorrectly label the intersection as unsafe. The robot is placed at a signalized intersection with a green pedestrian light and a vehicle approaching from the lower left corner of the image. As the intersection contains a middle island, and since there is no traffic approaching from the significant direction (top left corner of the image), the crossing is labeled as safe. However, as neither classifier has a representation of the structure of the intersection showing the presence of the middle island, the interval is in turn misclassified as unsafe for crossing.

By incorporating semantic knowledge of the scene or learning an obstacle map of the environment, the aforementioned problem can be rectified as the classifier can learn about the various road topologies and their effect on the crossing decision.

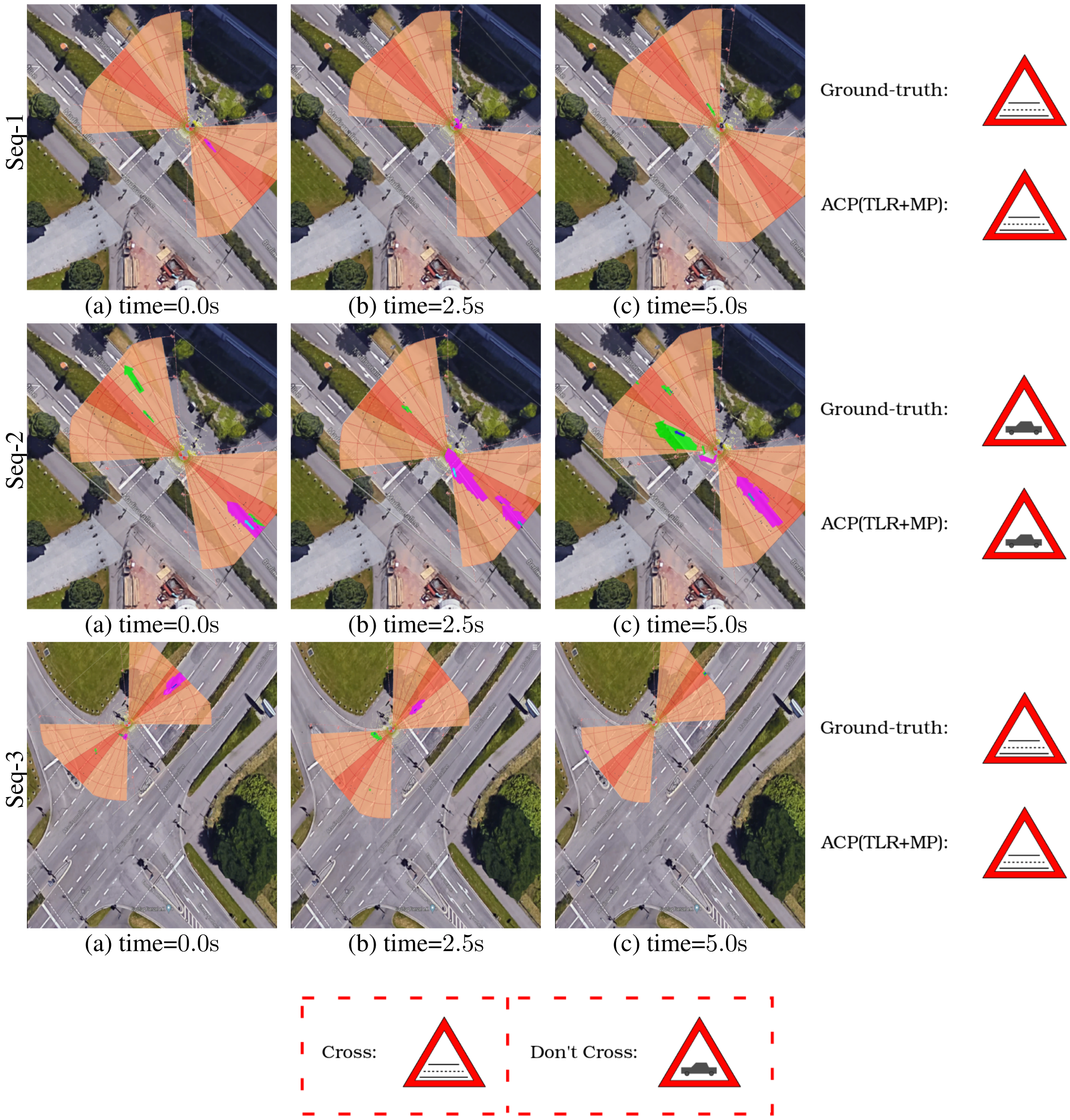
5.4.6 Generalization Analysis on the DeepLocCross dataset

In the following, we perform an extended evaluation of our proposed pipeline for predicting the safety of the intersection for crossing by analyzing the performance of each module as well as the entire system on the test sequences of the DeepLocCross dataset (Section 4.3.3). Note that we do not utilize the training sequences of the DeepLocCross dataset to pre-train the individual modules, but rather directly evaluate the generalization capabilities of our framework.

Employing our IA-TCNN on the dataset to predict the future trajectories of all observable traffic participants, we achieve an average displacement error of $0.14\mathrm{m}$, $11.41°$, $0.13\mathrm{m/s}$ in terms of translation, rotation and velocity respectively. Furthermore, using AtteNet we achieve an accuracy of 78.5% for predicting the state of the traffic light. Overall, predicting the safety of the intersection for crossing, our ACP classifier achieves a precision of 85.7% and a recall of 78.2% on the DeepLocCross dataset. The low prediction errors achieved by the overall network as well as the individual modules demonstrate the generalization capabilities and efficacy of our proposed framework.

Additionally, we perform qualitative analysis of the crossing decision predicted by our proposed ACP classifier on the DeepLocCross dataset in Figure 5.20. We depict three example scenarios from the dataset, where each scenarios is represented by three sequence images from the beginning, middle and end of the prediction interval. As in Figure 5.19, we overlay the sensor detections on birds-eye-view images of the intersection to provide a more comprehensive image of the scene. Seq-1 depicts a scenario wherein a cyclist is approaching the robot on the sidewalk from the direction of oncoming traffic and continues to cycle past the robot. Utilizing and predicting the orientation information for the observed traffic participants, our IA-TCNN network is able to predict an accurate trajectory for the cyclist continuing on the sidewalk and hence passing behind the robot as opposed to in front of it. This information is in turn used by our ACP classifier to correctly predict the safety of the intersection for crossing.

Seq-2 depicts a situation with heavy oncoming traffic, in which our ACP classifier accurately predicts the intersection at the observed interval to be not safe for crossing. In Seq-3, in the first half of the interval a car is approaching the intersection, however, at the remaining half it slows down as the traffic light signal changes. By utilizing both the traffic light predictions from AtteNet showing the pedestrian traffic light to be green, and the trajectory information from IA-TCNN predicting the continued deceleration of the car until it comes to a halt at the end of the interval, our ACP classifier is able to accurately predict the safety of the intersection at the given interval for crossing.



(a) time=0.0s (b) time=2.5s (c) time=5.0s

(a) time=0.0s (b) time=2.5s (c) time=5.0s

(a) time=0.0s (b) time=2.5s (c) time=5.0s

Figure 5.20: Depiction of various crossing scenarios from the DeepLocCross dataset. For each sequence, we depict three timesteps corresponding to the beginning, middle and end of the interval. We overlay the sensor visualization on birds-eye-view images from the corresponding intersections. For each sequence, on the right-most column, we depict the ground-truth label versus the predictions of our proposed ARCP(TLR+MP) classifier. The legend is shown enclosed in a red rectangle.

5.5 Related Work

In this chapter, we presented a novel approach to predict the safety of street intersections for crossing by incorporating predictions from a motion prediction sub-network and a traffic light recognition sub-network. In the following, we review recent related works in the areas of motion prediction, traffic light recognition and intersection handling.

Motion Prediction approaches can be divided into two categories: methods modeling interactions among pedestrians and approaches modeling the behavior of vehicles. Among the first methods to model pedestrian interactions is the *Social Forces (SF)* method of Helbing and Molnar [84] in which they applied a potential field based approach with attractive and repulsive forces to model the interactions among various pedestrians in the surrounding environment. A subsequent variant of the *SF* method was later proposed by Yamaguchi *et al.* [203], in which the authors employed a data-driven approach to estimate the hidden variables affecting the behavior of the agents such as group affinity and destinations. Lerner *et al.* [115] used an example-based reactive approach to model pedestrian behavior by creating a database of local spatio-temporal scenarios. During an interaction, the autonomous agent samples its trajectory incrementally by considering similar spatio-temporal scenarios from the database. Subsequently, Pellegrini *et al.* [148] introduced the *Linear Trajectory Avoidance (LTA)* method which uses similar concepts from crowd simulation to model the behavior of pedestrians in crowded environments using linear extrapolation over short intervals. Kuderer *et al.* [106] employed a maximum entropy reinforcement learning approach to model human navigation behavior. In order to approximate the feature expectations, the proposed method employs Dirac delta functions at the modes of the distributions. However, while this approach was able to accurately model the behavior of pedestrians, suboptimal behavior often emerged due to the large amount of data required to capture the stochasticity of human behavior. In order to address this problem and enable accurate modeling of the pedestrian behavior, Kretzschmar *et al.* [103] proposed computing feature expectations using Hamiltonian Markov chain Monte Carlo sampling.

While the aforementioned approaches were able to capture the pedestrian behavior in specific situations, the need for defining hand-crafted features made them undesirable for deployment in dynamic environments. Inspired by the success of deep learning based approaches in the various areas of computer vision and robotics, Alahi *et al.* [24] proposed an approach dubbed *Social LSTM*. Using a Long-Short Term Memory (LSTM) network architecture and a *Social Pooling* layer that leverages spatial information of nearby pedestrians thereby implicitly modeling interactions among them. Similarly, Sun *et al.* [179] used a sequence-to-sequence LSTM encoder-decoder architecture to predict the pedestrian position and the angle of direction. The authors showed that incorporating the angular information in addition to the temporal information led to a significant improvement in the accuracy of the prediction. Vemula *et al.* [197] proposed an alternative *Social Attention* method to predict future trajectories based on capturing the relative importance of pedestrians regardless of their proximity. The authors formulated the problem as a spatio-temporal graph with nodes representing the pedestrians and edges capturing the dynamics of the interactions between two pedestrians such as orientation and distance. Concurrently, Pfeiffer *et al.* [149] proposed an LSTM based data driven model for motion prediction by incorporating the obstacle map of the environment and encoding the surrounding pedestrians in polar angular space, thereby enabling fast inference times in crowded environments. More recently, Gupta *et al.* [77] proposed the use of recurrent based Generative Adversarial Network (GAN) to generate and predict socially acceptable paths. Their proposed *SGAN* approach is comprised of an LSTM-based encoder-decoder generator to predict the future trajectories, followed by an LSTM-based discriminator to predict whether each generated trajectory follows the social norms. Similarly, Sadeghain *et al.* [166] presented a framework for predicting trajectories based on GAN dubbed *SoPhie*. By utilizing an RGB image from the scene and the trajectory information of the pedestrians, the method computes both the physical and social context vectors by focusing on only the relevant information for each observed pedestrian. The computed vectors are then utilized by an LSTM-based GAN module to generate physically and socially acceptable trajectories.

Over the years, several methods have been proposed for trajectory estimation of vehicles [114]. Lefèvre *et al.* [113] proposed a Bayesian network to infer the driver's intention by utilizing the digital map of the road network. Kim *et al.* [97] proposed a trajectory prediction method that employs a recurrent approach to predict the future coordinates of all surrounding vehicles using an occupancy grid map representation with probability values to reflect the uncertainty of the predictions. Similarly, Baumann *et al.* [34] proposed an encoder-decoder architecture to predict the ego-motion of the vehicle using previous path information. In order to minimize the potential collision risk, Park *et al.* [147] proposed an encoder-decoder LSTM architecture accompanied with beam search to produce the most likely K trajectories.

Despite the varying application areas of the motion prediction task, there is a growing consensus that recurrent units in combination with trajectory information of the most relevant pedestrians/vehicles can provide accurate predictions. While this is true, it comes at the cost of the representational and run-time capabilities of these methods. As the majority of the aforementioned approaches model each pedestrian/vehicle separately by predicting only their local neighborhood, suboptimal behavior often occurs in complex densely populated environments. In this chapter, we proposed a novel scalable neural network architecture to address the problem of learning trajectories in populated environments. Instead of the widely employed recurrent units such as LSTMs, our proposed network utilizes causal convolutions to model the sequential behavior of the various agents in the scene. Furthermore, by jointly learning the trajectories for all agents in the scene, our network is able to better leverage the interdependencies in the motion without the need for explicitly defining the relative importance of each agent. Finally, our approach is not restricted to modeling the behavior of either pedestrians or vehicles, but is rather able to learn and infer the complex interactions among the various types of agents in the scene.

Traffic Light Recognition is one of the vital tasks for autonomous agents operating in urban environments whether pedestrian assistant robots or autonomous vehicles. Although traffic lights are designed to be relatively easily perceived by humans, they are not always easily identified in camera images due to their small size, presence of other sources of similar lights e.g. brake lights, billboards and other traffic lights in different directions, and partial occlusions caused by different objects in the scene [90]. Furthermore, due to the highly dynamic nature of the environment, traffic light recognition approaches need to have fast inference times to enable safe deployment. In order to accurately recognize traffic lights in varying illumination conditions, John *et al.* [91] employed a Convolutional Neural Network (CNN) based approach to extract features from the image. Accompanied by a GPS sensor to identify the regions of interest within the image, the approach produces a saliency map containing the traffic light location to enable recognition in low lighting conditions. Behrendt and Novak [35] proposed a system for detecting, tracking and recognizing traffic lights for autonomous vehicles. Their approach utilizes the YOLO architecture [160] to detect the location of the traffic lights within the image. The traffic lights are then tracked using the ego-motion information and stereo imagery to triangulate their relative position. Finally, the state of the light is identified using a small neural network trained on the extracted regions.

Similarly, in order to enable accurate traffic light recognition in complex scenes, Li *et al.* [117] utilized prior information from the image regarding the position and size of the traffic light in order to reduce the computational complexity of locating it within the image. Additionally, they proposed an aggregate channel feature method accompanied with inter-frame information analysis to facilitate accurate and consistent recognition across the different frames. With the goal of improving the run-time capabilities and reducing the computational resources, Liu *et al.* [120] proposed a traffic light recognition system operating in an online manner on smartphones. Using ellipsoid geometry in the HSV colorspace, their approach is able to extract region proposals which are in turn passed through a kernel function to recognize the phase and type of the traffic light.

In contrast to the aforementioned methods for traffic light recognition, we do not perform any pre-processing or utilize any structural prior from the scene, rather our proposed network is able to attend to areas in the image containing the traffic light, thereby increasing ease of deployment and robustness to new environments.

Intersection Crossing Safety Prediction: Among the early works on enabling autonomous street crossing for pedestrian assistant robots are the works of Baker and Yanco [31, 32] in which the authors proposed a system to detect and track vehicles using cameras mounted on both sides of the robot. Using image differencing and edge extraction techniques, the method is able to identify and track vehicles in a two lane street. Subsequently, Bauer *et al.* [33] proposed an autonomous city explorer robot to navigate in urban environments. In their approach, the robot is able to handle street crossings by identifying and recognizing the state of the traffic light. In order to identify the safety of intersections for autonomous vehicles, Campos *et al.* [50] proposed a negotiation approach by solving local optimization problems for each of the vehicles approaching the intersection. Similarly, Medina *et al.* [131] proposed a decentralized *Cooperative Intersection Control (CIC)* system to enable safe navigation of a T-intersection for a platoon of vehicles. An alternate approach to cooperative intersection crossing is proposed in [26], in which the authors proposed a vehicle-to-vehicle intersection protocol guided by a GPS model, where each vehicle periodically broadcasts its pose and intent to nearby vehicles and the crossing priority is then decided by the vehicles among themselves.

Inspired by learning from demonstration approaches, Diaz *et al.* [65] proposed an approach to aid visually impaired users to remain within the crosswalk bounds while crossing a road. Their proposed method processes images from the scene to extract the relative destination of the user and in turn produces an audio signal as a beacon for the user to follow to reach the goal. More recently, Habibi *et al.* [78] and Jaipuria *et al.* [89] presented techniques for pedestrian intent prediction at intersections by utilizing the contextual information of the scene and *Augmented Semi Non-negative Sparse Coding (ASNSC)* for learning the motion primitives to enable more accurate predictions of the trajectories at street crossings. Fang and López [67] developed an approach for predicting the crossing intention of pedestrians. Their proposed method first detects and tracks pedestrians approaching the sidewalks and then utilizes this information to estimate the pose of the pedestrians by fitting skeletal features which are in turn utilized by a Random Forest classifier to predict the crossing intent. In this chapter, we proposed a novel method for predicting the safety of street intersections for crossing by utilizing information from both the interaction-aware motion prediction and traffic light recognition approaches. By leveraging the predicted trajectories of all observable vehicles and pedestrians in the vicinity of the robot in addition to the state of the traffic light if present, our approach is able to accurately estimate the safety of the intersection for crossing. Furthermore, as we do not rely on any prior knowledge of the environment or any form of communication technique with the surrounding traffic participants, our approach can be easily deployed in various environments without any additional pre-processing steps.

5.6 Conclusion

In this chapter, we presented a system for autonomous street crossing using multimodal data. Our system consists of two main network streams: a traffic light recognition stream and an interaction-aware motion prediction stream. Information from both streams is fused as input to a convolutional neural network to predict the safety of the intersection for crossing. We proposed AtteNet, a convolutional neural network architecture for traffic light recognition that utilizes the global information in the images to selectively emphasize informative features suppressing irrelevant features, while being robust to noisy data. We performed extensive experimental evaluations on various traffic light recognition benchmarks and show that the proposed architecture outperforms the existing methods. Furthermore, we proposed an interaction-aware temporal convolutional neural network architecture that utilizes causal convolutions to accurately predict the trajectories of all observable traffic participants. We demonstrated that our approach is scalable to complex urban environments while simultaneously being able to predict accurate trajectories of all the observable traffic participants in the scene. Experimental evaluations on several real-world datasets demonstrate that our architecture achieves state-of-the-art performance on both indoor and outdoor datasets, while achieving faster inference times and requiring less storage space in comparison to recurrent approaches.

In order to learn a classifier that is robust to the type of intersection, the learned representations from the traffic light recognition network and the interaction-aware motion prediction network are fused to infer the final crossing decision. By incorporating the uncertainty information from the motion prediction stream and the learned representations from the traffic light recognition stream, the classifier is robust to incorrect predictions by either task-specific sub-network. Moreover, we introduced the Freiburg Street Crossing dataset for motion prediction and intersection safety prediction for crossing which we make publicly available to facilitate future work on the topics of motion prediction and intersection safety prediction. Comprehensive experimental evaluations demonstrate the efficacy of the proposed system for determining the safety of the intersection for crossing. Furthermore, the results demonstrate the tolerance of the system to noise and inaccuracies in the data, while accurately generalizing to new unseen scenarios.

Chapter 6

Conclusion and Future Work

In this thesis, we proposed several contributions to the field of state estimation for mobile robots. Our contributions are focused on enabling mobile robot deployment in urban environments by leveraging the rich features from the scene to enable robust state estimation in an efficient manner. We presented frameworks for *i)* learning to reliably localize in urban environments using the abundant textual information in the scene, *ii)* learning to leverage the structural and temporal information to simultaneously predict the pose, ego-motion and semantics of the scene in a multitask network architecture, and finally, *iii)* learning to estimate the future trajectories of all traffic participants in the vicinity of the robot concurrently with recognizing the traffic light signal and utilizing information from both sources to predict the safety of a street intersection for crossing. We extensively evaluated our proposed frameworks on several real-world indoor and outdoor datasets. The results demonstrate that our proposed methods exceed the state of the art while enabling efficient online deployment.

We first tackled the problem of localizing in urban environments. Although GPS is frequently used to provide position estimates, its accuracy deteriorates in or near buildings due to outages. Furthermore, frequent structural changes in the environment require continuous map updates to enable successful localization. We proposed two novel probabilistic localization frameworks to address the aforementioned challenges by leveraging the textual information in the environment. Our first localization approach enables pose estimation in a single step, while the second aggregates information across multiple timesteps to produce a localization estimate that is tolerant to the amount of textual information available. We extract stable textual features from the environment and employ a probabilistic data association method that combines both distance and linguistic metrics to match the extracted text with landmarks in a publicly available map. Finally, we employ a particle filter-based method with dedicated sensor models in order to estimate the pose of the robot in the environment. Our contribution is the first method to utilize textual features from images of the scene to produce reliable localization estimates that are more accurate than the estimates obtained from GPS. As publicly available maps are frequently updated by the map provider, leveraging information from such maps renders our method robust to changes in the environment, and alleviates the need of frequently revisiting the environment to perform map updates. It further facilitates the deployment of robots in new environments without the need for any pre-processing steps. We evaluated our methods on data captured from three different cities, and the results demonstrate that both our proposed frameworks outperform GPS in terms of the localization accuracy, thereby demonstrating the robustness of utilizing textual features for localization in urban environments.

In the absence of textual information, enabling successful robot localization entails finding stable features while being robust to repetitive and reflective surfaces which are often present in the scene. Although local feature-based localization methods are able to accurately estimate the robot pose, expert knowledge is required for selecting the set of features that are representative of the environment. On the other hand, although deep learning-based localization methods are robust to weather and illumination variations, they are unable to match the performance of state-of-the-art local feature-based localization methods. With the goal of enabling accurate and robust pose estimation, we proposed two multitask learning convolutional neural network architectures for jointly estimating the global pose, ego-motion and semantics of the scene which enable the encoding of geometrical and structural features into deep learning-based localization methods. In order to enable the network to exploit the geometric information regarding the environment, we proposed a novel loss function which utilizes the relative motion information to constrict the search space of the global pose regression stream, thus enabling the network to learn a motion model that is globally consistent.

Our first architecture consists of a global pose regression stream and a Siamese-type visual odometry stream. We employ a parameter sharing scheme to enable each subnetwork to exploit the interdependencies among the tasks. Moreover, we proposed an improved architecture that further incorporates the semantic features from the scene into the global pose regression network to encode the structural information and semantic relations in the environment. We introduced an adaptive weighted fusion layer that enables the learning of favorable weights for the fusion of feature maps based on region activations, and utilized the layer to facilitate the encoding of the structural and geometrical constraints into the pose regression network. Additionally, we presented a self-supervised warping method to enable the segmentation network to incorporate temporal consistency in the prediction and accelerate training by exploiting the estimated ego-motion from the odometry stream to warp the semantic features from the previous timestep with the current timestep. We performed extensive experimental evaluations complemented with ablation studies on several indoor and outdoor benchmark datasets. The results demonstrate that both our single-task and multitask architectures outperform state-of-the-art methods, while achieving online run-time thus facilitating deployment in real-world scenarios. Our contributed architectures are the first deep learning-based methods to outperform local feature-based localization techniques in terms of localization accuracy, while being robust to various perceptual challenges.

As robots navigate in urban cities often encounter street intersections, safely navigating across street intersections is an essential task for mobile robots. Although a majority of the existing approaches utilize the traffic light signal to make an informed crossing decision, relying solely on this signal limits the navigation capabilities of the robot to streets that contain only signalized intersections. In order to overcome this limitation, we proposed a multimodal convolutional neural network framework for predicting the safety of street intersections for crossing. Our architecture consists of a motion prediction stream and a traffic light recognition stream. We fuse representations from both sub-networks to enable the prediction of a safe crossing strategy that is agnostic to the type of intersection. In order to enable accurate and reliable prediction of the intersection safety, our motion prediction network learns to estimate the future trajectories of all observable traffic participants concurrently. Our network accomplishes this by utilizing causal convolutions coupled with a binary masking mechanism to enable the prediction of trajectories with dynamic lengths for a varying number of traffic participants. This eliminates the need for creating handcrafted definitions for modeling the interdependencies between the traffic participants and enables the network to learn a more realistic model of the trajectories. Furthermore, it expedites the flow of information throughout the network, thus facilitating the prediction of future trajectories for all observable traffic participants in an online manner.

Our proposed traffic light recognition stream employs a global attention mechanism which enables the network to selectively emphasize informative features belonging to the traffic light, while being robust to noisy data. We extensively evaluated our intersection safety prediction framework on a real-world dataset captured at various street intersections, demonstrating the benefit of leveraging both the motion information of traffic participants in the vicinity of the robot and the traffic light signal in accurately predicting the safety of an intersection for crossing. The results further demonstrate the robustness of our proposed framework to the type of intersection and incorrect predictions by either sub-network. Furthermore, extensive experiments on the traffic light recognition and the motion prediction modules demonstrate that both our networks exceed the state of the art, while enabling online deployment in an efficient manner.

In summary, in this thesis we proposed several contributions enabling robots to reliably estimate their state and the state of all agents in their vicinity by utilizing semantic information and leveraging rich features from the environment. With the goal of facilitating the deployment of robots in urban environments, our proposed methods enable the robot to learn a robust model of its surroundings. Our frameworks achieve state-of-the-art performance in each of the addressed tasks, while enabling efficient online deployment. We believe that the proposed techniques have brought us a step closer towards the goal of life-long robot deployment in urban environments.

Future Work

There are several directions in which future research can extend the scope and capabilities of the methods proposed in this thesis. We presented an approach for localization in urban environments that leverages the textual information in the scene utilizing publicly available maps. As the area traversed by the robot increases, the probability of encountering signs belonging to the same shop increases. As an example, there are multiple branches for a certain chain restaurant within a city. In order to reduce the search space for the first timestep, we can complement our approach with a topological localization method that runs as an initial step. This would have the advantage of eliminating potential incorrect initializations for the particle filter and thus speed up the overall localization procedure. Another potential research direction to reduce incorrect data associations would be to incorporate the uncertainty of the extracted text from the text spotting phase into the landmark selection procedure. Currently, we assume that all selected landmarks are equally likely, which adds tolerance to text spotting failures. However, it could also potentially introduce noise to the localization method. Incorporating the text spotting uncertainty to the extracted landmarks could reduce this noise, as particles sampled in the vicinity of landmarks with low probabilities would get penalized. Furthermore, text signs within a city can contain words from different languages, for instance signs at or near train stations are often both in English and the official language of the country/city. Instead of using the WordNet database in the official language of the country/city, we can combine the databases from multiple languages, thus improving the accuracy of our lexical data association metric.

In this thesis, we proposed a multitask learning architecture for jointly predicting the global pose, visual odometry and semantic segmentation for a given scene image. Currently, our network predictions for the ego-motion are scale dependent, which requires the visual odometry network to be trained for every new scene. In order to mitigate this, we can adapt our odometry prediction by utilizing depth maps predicted from a pre-trained network in order to recover the scale in a manner similar to the approach proposed by Yin *et al.* [205]. As the robot traverses in urban environments, it is often surrounded by other pedestrians and cars with different traversal directions and velocities than its own. Learning to estimate the motion of the surrounding dynamic and movable objects in the scene through scene flow can be beneficial towards improving the visual odometry estimated by the network, as the network can learn to discard feature correspondences belonging to dynamic objects. Furthermore, the presence of dynamic objects in the scene can impair the quality of the predicted global poses. Learning to segment out the dynamic objects in the scene and inpaint the occluded parts of the image would enable our method to produce more accurate pose estimates that are invariant to occlusions.

We addressed the problem of predicting the safety of street intersections for crossing in this thesis by employing a multimodal framework that simultaneously predicts the future trajectories for all observable traffic participants and recognizes the traffic light signal. One future direction would be to incorporate obstacle map predictions of the environment into the motion prediction sub-network. We believe that knowledge about the vicinity can improve the accuracy of the predicted trajectories by avoiding paths that intersect with obstacles. Furthermore, this could also potentially improve the accuracy of predicting the street intersection safety for crossing, as it eliminates false negative predictions by leveraging the road structure. Learning to semantically classify the traffic participants can also aid in understanding the potential interactions among them, thereby increasing the accuracy of the predicted trajectories, as well as improving the prediction of the street intersection safety for crossing. Another interesting direction would be learning to predict the direction of the traffic flow and incorporating the information into the intersection safety prediction sub-network. For instance, crossing a T-junction, the robot needs to only observe traffic flowing in a perpendicular direction to it and safely discard parallel traffic flow. Utilizing the information about the direction of the traffic flow can aid in reducing false negative predictions from the classifier, and thus improve the overall prediction accuracy.

In summary, the aforementioned directions are a subset of the potential extensions in the scope of this thesis. There are several challenges facing the deployment of mobile robots in urban environments due to the inherent complexity and stochastic nature of the environment. We hope that the insights gained from the work presented in this thesis will inspire future work, and bring us a step closer towards long-term deployment of mobile robots in urban environments.

Appendices

Appendix A

Additional Ablation Studies on Semantics-Aware Pose Regression

In this appendix, we present additional ablation studies investigating the various design choices for the problem of multitask learning for geometry and semantics-aware pose regression. Our goal is to provide a complete overview on the proposed framework through an in-depth analysis of the architectural choices. We begin with an evaluation of the base network topology, followed by an evaluation of the choice of the downsampling stage to carry out the previous pose fusion. We conclude this appendix with quantitative evaluations of the multitask learning design choices. The following results are the outcome of joint work with Abhinav Valada [157, 193].

A.1 Base Architecture Topology

In the following, we investigate the effect of the different base architectures on the median localization error for the task of global pose regression. We evaluate the performance of employing shallow to deeper residual architectures with ReLU activation functions on the DeepLoc dataset in Table A.1. We observe that increasing the depth of the model improves the localization accuracy, as shown by the improvement between M1, M2 and M3 models. While increasing the number of layers improves the representational capabilities of the network, the risk of over-fitting to the training data increases. This can be observed by inspecting the localization accuracy of the M2 and M3 models, in which the rotational error is reduced by approximately 30.0% at the cost of reduced translational accuracy. Utilizing the pre-activation ResNet-50 architecture [82] in the M4 model improves both the translational and rotational accuracy as it reduces over-fitting and enables faster convergence of the network during training. Therefore, we utilize this model as the backbone for our VLocNet++$_{\mathrm{STL}}$ network.

Table A.1: Comparison of the VLocNet++$_{\text{STL}}$ base architecture topology on the visual localization error on the DeepLoc dataset [158].

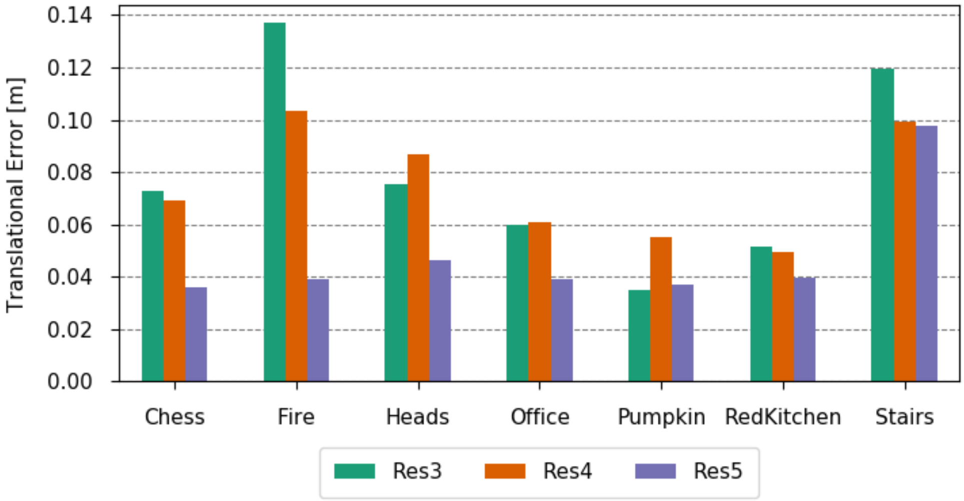
Method	Base Model	Activation	Median Error
M1	ResNet-18	ReLU	$0.83\text{m}, 5.96°$
M2	ResNet-34	ReLU	$0.57\text{m}, 4.04°$
M3	ResNet-50	ReLU	$0.65\text{m}, 2.87°$
M4	PA ResNet-50	ReLU	$0.57\text{m}, 2.44°$

A.2 Previous Pose Fusion

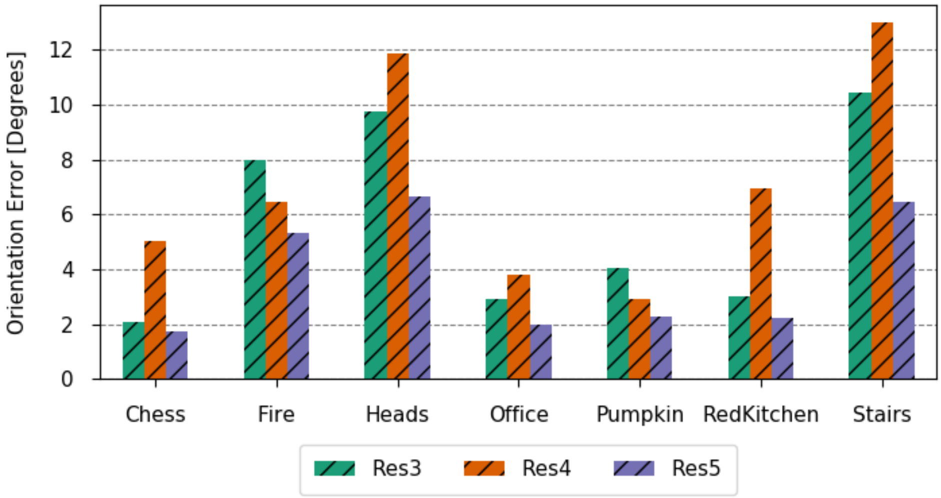
In order to determine the downsampling stage to fuse the previous pose information into our architecture, we evaluate the median localization error achieved by fusing the previous pose at various stages of our VLocNet$_{\text{STL}}$ architecture, namely at *Res3*, *Res4* and *Res5*. In Figure A.1, we plot the localization error at each of the aforementioned stages on the Microsoft 7-Scenes dataset. The results show that performing the pose fusion at earlier stages results in an imbalance in the pose error by either reducing the translational error at the cost of the rotational error or vice versa. However, by fusing the previous predicted pose at *Res5*, the localization accuracy achieved is consistently higher than at the remaining stages. We hypothesize this occurs due to the maturity of the features at this stage which enable the network to take full advantage of the pose information.

A.3 Parameter Sharing Evaluation

In this section, we study the impact of sharing features across the global pose regression and visual odometry streams by experimenting with varying amounts of feature sharing. Table A.2 shows the median global localization error achieved by VLocNet$_{\text{MTL}}$ with various amounts of sharing between the pose regression and the visual odometry streams on the Microsoft 7-Scenes dataset. We experiment with maintaining a shared stream up to the end of *Res2*, *Res3* and *Res4* blocks (Figure 2.7). We only consider the error achieved by the global localization stream as it is considered the primary task which we aim to improve, while the visual odometry estimation is an auxiliary task. The results show that maintaining a shared stream up to the end of *Res4* block results in lower localization accuracy. We believe this occurs as the representations learned at the *Res4* block are more task-specific and as such maintaining a shared stream negatively impacts both tasks. Similarly, maintaining a shared stream until the end of the *Res2* block results in a larger pose error as the representations learned by the end of the second residual block are too generic to provide benefit to either task. We achieve the best performance by maintaining



(a) Translational error



(b) Rotational error

Figure A.1: Comparison of the median localization error from fusing the previous pose information at various stages in the VLocNet$_{\text{STL}}$ architecture on the Microsoft 7-Scenes dataset [193].

Table A.2: Summary of the localization performance achieved by VLocNet$_{\text{MTL}}$ with varying amounts of sharing on the Microsoft 7-Scenes dataset [193].

Scene	Res2	Res3	Res4
Chess	0.04m, 1.60°	**0.03**m, 1.69°	0.05m, 1.76
Fire	0.05m, 4.40°	**0.04**m, 4.86°	0.05m, 4.59
Heads	0.05m, 4.44°	**0.05**m, 4.99°	0.06m, 5.99°
Office	0.04m, 1.68°	**0.03**m, **1.51**°	0.04m, 1.82°
Pumpkin	0.05m, 1.83°	**0.04**m, 1.92°	0.04m, 1.64°
RedKitchen	0.04m, 1.89°	**0.03**m, **1.72**°	0.04m, 1.75°
Stairs	0.10m, 5.08°	**0.07**m, 4.96°	0.09m, 4.67°
Average	0.06m, 2.99°	**0.04**m, 3.09°	0.05m, 3.17°

a shared stream until the end of the *Res3* block resulting in an improvement of 12.5% and 18.49% in the translational and rotational components of the pose over the single-task VLocNet$_{\text{STL}}$ architecture. We believe these results further demonstrate the utility of jointly learning visual localization and odometry estimation.

A.4 Evaluation of the Semantic Feature Fusion

We experiment with fusing the semantic feature maps from *Res5c* into the localization stream at various stages of the *Res4* block in order to study the impact of the feature fusion on the localization accuracy. Although the spatial dimensions of the semantic feature maps match the feature maps at the *Res5* block in the localization stream, the *Res5* block has a substanially larger number of feature channels which would outweigh the rich semantic features from the segmentation stream, thereby diluting their impact. We do not attempt to perform the semantic feature map fusion at earlier stages of the network before the *Res4* block as we believe incorporating high level semantic features at early/intermediate stages of the network would be of no benefit since the features learned at those stages are of lower level.

In Table A.3, we show the median localization pose error achieved by fusing the feature maps at different stages of VLocNet++$_{\text{MTL}}$ on the DeepLoc dataset. In an attempt to avoid having a cyclic dependency between the localization and segmentation sub-networks, we do not fuse the semantic feature maps at *Res4b* whose output is forwarded to the segmentation stream. Fusing the feature maps at the middle of the *Res4* block at *Res4c* results in the lowest localization error of $0.32\text{m}, 1.48^\circ$ as it is able to achieve the right balance between task-specificity and feature maturity.

Table A.3: Evaluation of the impact of the semantic feature fusion on the localization performance. The fusion layer denotes where the semantic feature maps are fused into the localization stream. The results are shown for the DeepLoc dataset [158].

Fusion Layer	Median Translational Error	Median Rotational Error
No fusion	0.37m	$1.93°$
Res4a	0.49m	$3.10°$
Res4c	$\mathbf{0.32}\text{m}$	$1.48°$
Res4d	0.54m	$1.30°$
Res4e	0.46m	$1.95°$
Res4f	0.61m	$1.45°$

A.5 Evaluation of the Adaptive Weighted Fusion Layer

In order to evaluate the efficacy of the proposed fusion layer at favorably combining feature maps from various timesteps and sub-networks, in this section, we compare the localization accuracy of VLocNet++$_{\text{MTL}}$ that incorporates the adaptive weighted fusion layer, with three of the most commonly employed feature combination methods. Furthermore, in order to gain more insight on the effect of the feature fusion method on the overall accuracy of the approach, we compare the results with the single-task VLocNet++$_{\text{STL}}$ model. We compare the following fusion schemes:

- **MTL-input-concat**: A simple approach to incorporate the semantic features, learned by the segmentation stream, into the visual localization stream is by concatenating the predicted segmentation mask M_t with the input image I_t as a fourth image channel. The input to the localization sub-network would be the resulting four-stream tensor.

- **MTL-mid-concat**: As a second baseline, we concatenate the semantic feature maps with intermediate representations of the pose regression stream. As our earlier experiments in Appendix A.4 demonstrate that fusing the semantic feature maps at *Res4c* is most beneficial, we similarly concatenate the semantic representations at *Res4c*. We additionally compare with concatenating the semantic feature maps at the end of the *Res4* block.

- **MTL-shared**: For the final baseline, we investigate the effect of sharing the latent space of both networks as a variant of the approach proposed in [22]. Latent space sharing can be represented by sharing a layer among the various networks. In our implementation, we share the *Res4c* layer between both the segmentation and localization streams.

Table A.4: Comparison of VLocNet++$_{\text{MTL}}$ with baseline models for fusing semantic features into the localization stream. Results are shown for the entire DeepLoc dataset [158].

Method	Median Translational Error	Median Rotational Error
VLocNet++$_{\text{STL}}$	0.37m	$1.93°$
MTL-input-concat	0.56m	$3.63°$
MTL-mid-concat *Res4c*	0.55m	$3.38°$
MTL-mid-concat *Res4f*	0.50m	$3.10°$
MTL-shared	1.17m	$4.20°$
VLocNet++$_{\text{MTL}}$	$\mathbf{0.32}$m	$\mathbf{1.48°}$

Table A.4 shows the median localization accuracy for each of the aforementioned baselines, in addition to our single-task VLocNet++$_{\text{STL}}$ and multitask VLocNet++$_{\text{MTL}}$ on the DeepLoc dataset. Naively concatenating the semantic feature maps as in the MTL-input-concat approach results in significantly worse accuracy in comparison to the single-task architecture. We observe that concatenating the semantic features at the end of the *Res4* block is more beneficial than at *Res4c* which can be attributed to the complexity of the learned representations. Nonetheless, this variant still achieves a lower localization accuracy in comparison to VLocNet++$_{\text{STL}}$. MTL-shared achieves the lowest accuracy in comparison to the remaining methods. We believe this occurs due to the diverse nature of the tasks learned and as such the representations learned by each stream differ significantly. Thus, sharing weights across both network streams subsequently lowers the performance of each individual task. VLocNet++$_{\text{MTL}}$ achieves the highest performance with an improvement of 36.0% in the translational and 53.9% in the rotational components of the pose compared to the best performing baseline (MTL-input-concat). Moreover, the results demonstrate the effectiveness of the proposed fusion scheme in aggregating both inter- and intra-dependent feature maps. Utilizing the proposed layer achieves an improvement of 13.5% and 23.3% in the translation and orientation pose accuracy over VLocNet++$_{\text{STL}}$, which further demonstrates the utility of incorporating semantic features into the localization network.

List of Figures

List of Tables

Bibliography

[1] Unimation. https://goo.gl/sE6y3a, 1962. Accessed:2018-12-06.

[2] KUKA. https://www.kuka.com/en-de, 1973. Accessed:2018-12-07.

[3] ABB robotics. https://new.abb.com/products/robotics, 1974. Accessed:2018-12-07.

[4] iRobot. https://www.irobot.de/uber-irobot/uber-irobot/History, 2002. Accessed:2018-12-07.

[5] Spidertracks. https://www.spidertracks.com/, 2005. Accessed:2018-12-04.

[6] Waymo. https://waymo.com/journey/, 2009. Accessed:2018-12-07.

[7] MIIMO Robotic Lawnmower. https://www.honda.co.uk/lawn-and-garden/products/miimo/overview.html, 2013. Accessed:2018-12-17.

[8] Camera calibration and 3D reconstruction. https://goo.gl/8ii7wP, 2014. Accessed:2018-12-07.

[9] Starship. https://www.starship.xyz/company/, 2014. Accessed:2018-12-07.

[10] Tesla autopilot. https://www.tesla.com/autopilot?redirect=no, 2014. Accessed:2018-12-07.

[11] Marble. https://www.marble.io/, 2015. Accessed:2018-12-07.

[12] Parcel delivery: The future of last mile. https://www.mckinsey.com/~/media/mckinsey/industries/travel%20transport%20and%20logistics/our%20insights/how%20customer%20demands%20are%20reshaping%20last%20mile%20delivery/parcel_delivery_the_future_of_last_mile.ashx, 2016. Accessed:2018-12-07.

[13] Nuro. https://nuro.ai/product/, 2016. Accessed:2018-12-07.

[14] BMW: Autonomous driving: Digital measuring of the world. https://www.bmw.com/en/innovation/mapping.html, 2017. Accessed:2018-12-07.

[15] San francisco bans delivery robots. https://goo.gl/RhfysH, 2017. Accessed:2018-12-07.

[16] California DMV: Report of traffic collision involving an autonomous vehicle. https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/autonomousveh_ol316+, 2018. Accessed:2018-12-07.

[17] Deepmap. https://www.deepmap.ai/, 2018. Accessed:2018-12-07.

[18] HERE HD live map. https://goo.gl/AF36TD, 2018. Accessed:2018-12-07.

[19] Delivery robots. https://goo.gl/6fko2i, 2018. Accessed:2018-12-07.

[20] Mapper.ai. https://mapper.ai/product/, 2018. Accessed:2018-12-07.

[21] M. Abadi, A. Agarwal, P. Barham, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

[22] A. Abdulnabi, G. Wang, J. Lu, and K. Jia. Multi-task CNN model for attribute prediction. *IEEE Transactions on Multimedia*, 17(11):1949–1959, 2015.

[23] P. Agarwal, W. Burgard, and L. Spinello. Metric localization using Google Street View. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[24] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[25] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[26] R. Azimi, G. Bhatia, R. R. Rajkumar, and P. Mudalige. Stip: Spatio-temporal intersection protocols for autonomous vehicles. In *International Conference on Cyber-Physical Systems (ICCPS)*, 2014.

[27] D. J. Backman, G. V. Roe, F. D. Defalco, and W. R. Michalson. Hand-held gps-mapping device, 1999. US Patent 5,902,347.

[28] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture. *arXiv: 1511.00561*, 2015.

[29] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[30] J. E. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the second international conference on genetic algorithms*, volume 206, pages 14–21, 1987.

[31] M. Baker and H. A. Yanco. A vision-based tracking system for a street-crossing robot. *University of Massachusetts Lowell Technical Report*, 2003.

[32] M. Baker and H. A. Yanco. Automated street crossing for assistive robots. In *International Conference on Rehabilitation Robotics*, 2005.

[33] A. Bauer, K. Klasing, G. Lidoris, Q. Mühlbauer, F. Rohrmüller, S. Sosnowski, T. Xu, K. Kühnlenz, D. Wollherr, and M. Buss. The autonomous city explorer: Towards natural human-robot interaction in urban environments. *International Journal of Social Robotics*, 1(2):127–140, 2009.

[34] U. Baumann, C. Glaeser, M. Herman, and J. M. Zöllner. Predicting ego-vehicle paths from environmental observations with a deep neural network. In *IEEE International Conference on Robotics & Automation (ICRA)*, 2018.

[35] K. Behrendt and L. Novak. A deep learning approach to traffic lights: Detection, tracking, and classification. In *IEEE International Conference on Robotics & Automation (ICRA)*, 2017.

[36] H. Bilen and A. Vedaldi. Universal representations: The missing link between faces, text, planktons, and cat breeds. *arXiv:1701.07275*, 2017.

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

[38] M. Blomberg and G. Henriksson. Evidence for the minoan origins of stellar navigation in the aegean. *Actes de la Vème conférence de la SEAC*, 1999.

[39] F. Boniardi, T. Caselitz, R. Kümmerle, and W. Burgard. Robust lidar-based localization in architectural floor plans. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[40] N. Bowditch. National imagery and mapping agency. *American practical navigator:" Bowditch". An epitome of navigation. Originally by Nathaniel Bowditch (1773-1838). Arcata (CA, US): Paradise Cay Publications*, 2002.

[41] E. Brachmann and C. Rother. Learning less is more-6d camera localization via 3d surface regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[42] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother. DSAC - differentiable RANSAC for camera localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[43] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[44] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transaction on Intelligent Vehicles*, 20:1–1, 2017.

[45] M. A. Brubaker, A. Geiger, and R. Urtasun. Lost! leveraging the crowd for probabilistic visual self-localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[46] A. Budanitsky and G. Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[47] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial intelligence*, 114(1-2):3–55, 1999.

[48] W. Burgard, A. Valada, N. Radwan, T. Naseer, J. Zhang, J. Vertens, O. Mees, A. Eitel, and G. Oliveira. Perspectives on deep multimodal robot learning. In *Proceedings of the International Symposium of Robotics Research*, 2017.

[49] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32 (6):1309–1332, 2016.

[50] G. D. D. Campos, P. Falcone, and J. Sjoberg. Autonomous cooperative driving: a velocity-based negotiation approach for intersection crossing. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.

[51] R. Caruana. Multitask learning. *Machine Learning*, 1997.

[52] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. *arXiv preprint arXiv:1710.11063*, 2017.

[53] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. 2015.

[54] L. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[55] H. Chisholm. Encyclopædia britannica. 24:749–751, 1910.

[56] D. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[57] J. Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[58] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[59] N. R. Council. *The global positioning system: A shared national asset*. National Academies Press, 1995.

[60] T. Cover and P. Hart. Nearest neighbor pattern classification. *Transactions on Information Theory*, 13(1):21–27, 1967.

[61] D. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the world's photos. In *International Conference on World Wide Web*, 2009.

[62] M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *International Journal of Robotics Research (IJRR)*, 27(6), 2008.

[63] M. Cummins and P. Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *International Journal of Robotics Research (IJRR)*, 30(9):1100–1123, 2011.

[64] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[65] M. Diaz, R. Girgis, T. Fevens, and J. Cooperstock. To veer or not to veer: Learning from experts how to stay within the crosswalk. In *International Conference on Computer Vision Workshops (ICCV Workshops)*, 2017.

[66] M. Donoser and D. Schmalstieg. Discriminative feature-to-point matching in image-based localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[67] Z. Fang and A. M. López. Is the pedestrian going to cross? answering by 2d pose estimation. *arXiv preprint arXiv:1807.10580*, 2018.

[68] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[69] G. Floros, B. van der Zander, and B. Leibe. Openstreetslam: Global vehicle localization using openstreetmaps. In *IEEE International Conference on Robotics & Automation (ICRA)*, 2013.

[70] D. Fox, S. Thrun, F. Dellaert, and W. Burgard. Particle filters for mobile robot localization. In *Sequential Monte Carlo methods in practice*, pages 401–428. 2001.

[71] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha. Visual simultaneous localization and mapping: A survey. *Artificial Intelligence Review*, 43(1): 55–81, 2015.

[72] E. Garcia-Fidalgo and A. Ortiz. Vision-based topological mapping and localization methods: A survey. *Robotics & Autonomous Systems*, 64:1–20, 2015.

[73] J. Geweke. Bayesian inference in econometric models using monte carlo integration. *Econometrica: Journal of the Econometric Society*, pages 1317–1339, 1989.

[74] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proc. of the Int. Conf. on Artificial Intelligence and Statistics*, 2010.

[75] C. Gómez, M. Mattamala, T. Resink, and J. R. del Solar. Visual slam-based localization and navigation for service robots: The pepper case. *arXiv preprint arXiv:1811.08414*, 2018.

[76] E. Griffith, C. Hudson, and T. L. Mosher. Uninterruptable ads-b system for aircraft tracking, 2005. US Patent 6,952,631.

[77] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[78] G. Habibi, N. Jaipuria, and J. P. How. Context-aware pedestrian motion prediction in urban intersections. *arXiv preprint arXiv:1806.09453*, 2018.

[79] Q. Hao, R. Cai, Z. Li, L. Zhang, Y. Pang, and F. Wu. 3d visual phrases for landmark recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[80] J. Hayes and A. A. Efros. IM2GPS: Estimating geographic information from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[81] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[82] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, 2016.

[83] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *Intelligent Systems and their Applications*, 13(4):18–28, 1998.

[84] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

[85] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[86] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2017.

[87] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

[88] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[89] N. Jaipuria, G. Habibi, and J. P. How. A transferable pedestrian motion prediction model for intersections with different geometries. *arXiv preprint arXiv:1806.09444*, 2018.

[90] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi. Vision for looking at traffic lights: Issues, survey, and perspectives. *IEEE Transactions on Intelligent Transportation Systems (ITS)*, 17(7):1800–1815, 2016.

[91] V. John, K. Yoneda, B. Qi, Z. Liu, and S. Mita. Traffic light recognition in varying illumination using deep learning and saliency map. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2014.

[92] B. Jou and S. Chang. Deep cross residual learning for multitask visual recognition. In *ACM MM*, pages 998–1007, 2016.

[93] A. Kendall and R. Cipolla. Modelling uncertainty in deep learning for camera relocalization. *IEEE International Conference on Robotics & Automation (ICRA)*, 2016.

[94] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[95] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[96] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *arXiv:1705.07115*, 2017.

[97] B. Kim, C. M. Kang, S. H. Lee, H. Chae, J. Kim, C. C. Chung, and J. W. Choi. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. *arXiv preprint arXiv:1704.07049*, 2017.

[98] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[99] N. Kobyshev, H. Riemenschneider, and L. V. Gool. Matching features correctly through semantic understanding. In *International Conference on 3D Vision*, 2014.

[100] S. Kohli and S. Chen. Gps car navigation system, 2000. US Patent 6,041,280.

[101] K. R. Konda and R. Memisevic. Learning visual odometry with a convolutional network. In *VISAPP*, 2015.

[102] K. Konolige and M. Agrawal. FrameSLAM: from bundle adjustment to realtime visual mapping. *IEEE Transactions on Robotics*, 24(5):1066–1077, 2008.

[103] H. Kretzschmar, M. Kuderer, and W. Burgard. Learning to predict trajectories of cooperatively navigating agents. In *IEEE International Conference on Robotics & Automation (ICRA)*, 2014.

[104] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[105] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a non-metric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[106] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard. Feature-based prediction of trajectories for socially compliant navigation. In *Proceedings of Robotics: Science and Systems*, 2012.

[107] R. Kümmerle, R. Triebel, P. Pfaff, and W. Burgard. Monte carlo localization in outdoor terrains using multilevel surface maps. *Journal on Field Robotics*, 25(6-7): 346–359, 2008.

[108] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard. Autonomous robot navigation in highly populated pedestrian zones. *Journal on Field Robotics*, 32(4):565–589, 2015.

[109] T. R. Kurfess. *Robotics and automation handbook*. 2004.

[110] A. LaMarca and E. D. Lara. Location systems: An introduction to the technology behind location awareness. *Synthesis Lectures on Mobile and Pervasive Computing*, 3(1):1–122, 2008.

[111] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala. Camera relocalization by computing pairwise relative poses. *arXiv:1707.09733*, 2017.

[112] C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283, 1998.

[113] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán. Exploiting map information for driver intention estimation at road intersections. In *Intelligent Vehicles Symposium (IV)*, 2011.

[114] S. Lefèvre, D. Vasquez, and C. Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *Robomech Journal*, 1(1):1, 2014.

[115] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. In *Computer Graphics Forum*, volume 26, pages 655–664. Wiley Online Library, 2007.

[116] V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, 1965.

[117] X. Li, H. Ma, X. Wang, and X. Zhang. Traffic light recognition for complex scene with fusion detections. *IEEE Transactions on Intelligent Transportation Systems (ITS)*, 19(1):199–208, 2018.

[118] D. Lin. An information-theoretic definition of similarity. In *International Conference on Machine Learning (ICML)*, 1998.

[119] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv: 1506.04579*, 2015.

[120] W. Liu, S. Li, J. Lv, B. Yu, T. Zhou, H. Yuan, and H. Zhao. Real-time traffic light recognition based on smartphone platforms. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(5):1118–1131, 2017.

[121] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[122] P. Lothe, S. Bourgeois, E. Royer, M. Dhome, and S. Naudet-Collette. Real-time vehicle global localisation with a single camera in dense urban areas: Explotitation of coarse 3D city models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 863–870, 2010.

[123] L. Luft, A. Schaefer, T. Schubert, and W. Burgard. Closed-form full map posteriors for robot localization with lidar sensors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[124] L. Ma, J. Stueckler, C. Kerl, and D. Cremers. Multi-view deep learning for consistent semantic mapping with rgb-d cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[125] A. L. Majdik, Y. Albers-Schoenberg, and D. Scaramuzza. MAV urban localization from google street view data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[126] E. Marchand, H. Uchiyama, and F. Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22 (12):2633–2651, 2016.

[127] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[128] M. Mazuran, F. Boniardi, W. Burgard, and G. D. Tipaldi. Relative topometric localization in globally inconsistent maps. In *Robotics Research*, pages 435–451. 2018.

[129] S. McGrail. *Boats of the World: From the Stone Age to medieval times*. Oxford University Press on Demand, 2004.

[130] C. McManus, B. Upcroft, and P. Newmann. Scene signatures: Localised and point-less features for localisation. In *Proceedings of Robotics: Science and Systems*, 2014.

[131] A. I. M. Medina, N. V. D. Wouw, and H. Nijmeijer. Automation of a t-intersection using virtual platoons of cooperative autonomous vehicles. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2015.

[132] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu. Image-based localization using hourglass networks. *arXiv:1703.07971*, 2017.

[133] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu. Relative camera pose estimation using convolutional neural networks. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 675–687. Springer, 2017.

[134] G. A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[135] V. Mohanty, S. Agrawal, S. Datta, A. Ghosh, V. D. Sharma, and D. Chakravarty. Deepvo: A deep learning approach for monocular visual odometry. *arXiv preprint arXiv:1611.06069*, 2016.

[136] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[137] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning (ICML)*, 2010.

[138] T. Naseer and W. Burgard. Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[139] T. Naseer, M. Ruhnke, L. Spinello, C. Stachniss, and W. Burgard. Robust visual slam across seasons. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[140] T. Naseer, W. Burgard, and C. Stachniss. Robust visual localization across seasons. *IEEE Transactions on Robotics*, 34(2):289–302, 2018.

[141] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[142] Nexar. Nexar challenge-1: Using deep learning for traffic light recognition, 2016. URL `https://www.getnexar.com/challenge-1/`.

[143] A. Nicolai, R. Skeele, C. Eriksen, and G. A. Hollinger. Deep learning for laser based odometry estimation. In *RSS workshop Limits and Potentials of Deep Learning in Robotics*, 2016.

[144] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[145] G. Oliveira, A. Valada, C. Bollen, W. Burgard, and T. Brox. Deep learning for human part discovery in images. In *IEEE International Conference on Robotics & Automation (ICRA)*, 2016.

[146] G. Oliveira, N. Radwan, W. Burgard, and T. Brox. Topometric localization with deep learning. In *Proceedings of the International Symposium of Robotics Research*, 2017.

[147] S. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi. Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture. *arXiv preprint arXiv:1802.06338*, 2018.

[148] S. Pellegrini, A. Ess, K. Schindler, and L. V. Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[149] M. Pfeiffer, G. Paolo, H. Sommer, J. Nieto, R. Siegwart, and C. Cadena. A data-driven model for interaction-aware pedestrian motion prediction in object cluttered environments. 2018.

[150] I. Posner, P. Corke, and P. Newman. Using text-spotting to query the world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[151] X. Qu, B. Soheilian, and N. Paparoditis. Vehicle localization using mono-camera and geo-referenced traffic signs. In *Intelligent Vehicles Symposium (IV)*, 2015.

[152] N. Rader, M. Bausano, and J. E. Richards. On the nature of the visual-cliff-avoidance response in human infants. *Child Development*, pages 61–68, 1980.

[153] N. Radwan and W. Burgard. Effective interaction-aware trajectory prediction using temporal convolutional neural networks. In *Proceedings of the Workshop on Crowd Navigation: Current Challenges and New Paradigms for Safe Robot Navigation in*

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

[154] N. Radwan, G. D. Tipaldi, L. Spinello, and W. Burgard. Do you see the bakery? leveraging geo-referenced texts for global localization in public maps. In *IEEE International Conference on Robotics & Automation (ICRA)*, 2016.

[155] N. Radwan, W. Winterhalter, C. Dornhege, and W. Burgard. Why did the robot cross the road? - learning from multi-modal sensor data for autonomous road crossing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[156] N. Radwan, A. Valada, and W. Burgard. Multimodal interaction-aware motion prediction for autonomous street crossing. *arXiv preprint arXiv:1808.06887*, 2018.

[157] N. Radwan, A. Valada, and W. Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters (RA-L)*, 3(4):4407–4414, 2018.

[158] N. Radwan, A. Valada, and W. Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *arXiv preprint arXiv:1804.08366*, 2018.

[159] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, and S. Levine. Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration. *arXiv preprint arXiv:1707.02920*, 2017.

[160] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[161] J. Robinson. Emergency response data transmission system, 2007. US Patent 7,280,038.

[162] J. Roewekaemper, C. Sprunk, G. D. Tipaldi, C. Stachniss, P. Pfaff, and W. Burgard. On the position accuracy of mobile robot localization based on particle filters combined with scan matching. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[163] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision (ECCV)*, 2006.

[164] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*, volume 589. John Wiley & Sons, 2005.

[165] D. B. Rubin. Using the sir algorithm to simulate posterior distributions. *Bayesian statistics*, 3:395–402, 1988.

[166] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, and S. Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. *arXiv preprint arXiv:1806.01482*, 2018.

[167] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[168] T. Sattler, M. Havlena, K. Schindler, and M. Pollefeys. Large-scale location recognition and the geometric burstiness problem. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[169] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *Transactions on Pattern Analysis & Machine Intelligence*, (9):1744–1756, 2017.

[170] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[171] M. Schreiber, F. Pggenhans, and C. Stiller. Detecting symbols on road surface for mapping and localization using OCR. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2014.

[172] G. Schroth, S. Hilsenbeck, R. Huitl, F. Schweiger, and E. Steinbach. Exploiting text-related features for content-based image retrieval. In *International Symposium on Multimedia (ISM)*, 2011.

[173] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1610.02391*, 7, 2016.

[174] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts. *arXiv preprint arXiv:1701.06538*, 2017.

[175] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[176] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[177] G. Singh and J. Košecká. Semantically guided geo-location and modeling in urban environments. *Large-Scale Visual Geo-Localization*, 2016.

[178] H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige. Double window optimisation for constant time visual slam. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[179] L. Sun, Z. Yan, S. M. Mellado, M. Hanheide, and T. Duckett. 3dof pedestrian trajectory prediction learned from long-term autonomous mobile robot deployment data. In *IEEE International Conference on Robotics & Automation (ICRA)*, 2018.

[180] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford. On the performance of convnet features for place recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[181] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[182] M. Teichmann, M. Weber, M. Zöllner, R. Cipolla, and R. Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. *arXiv preprint arXiv:1612.07695*, 2016.

[183] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[184] A. Torii, J. Sivic, and T. Pajdla. Visual localization by linear combination of image descriptors. In *International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011.

[185] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[186] E. Trulls, A. C. Murtra, J. Pérez-Ibarz, G. Ferrer, D. Vasquez, J. M. Mirats-Tur, and A. Sanfeliu. Autonomous navigation for mobile service robots in urban pedestrian environments. *Journal on Field Robotics*, 28(3):329–354, 2011.

[187] S. S. Tsai, H. Chen, D. M. Chen, and B. Girod. Mobile visual search with word-HOG descriptors. In *Data Compression Conference (DCC)*, 2015.

[188] A. Valada and W. Burgard. Deep spatiotemporal models for robust proprioceptive terrain classification. *The International Journal of Robotics Research*, 36(13-14): 1521–1539, 2017.

[189] A. Valada, A. Dhall, and W. Burgard. Convoluted mixture of deep experts for robust semantic segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Workshop, State Estimation and Terrain Perception for All Terrain Mobile Robots*, 2016.

[190] A. Valada, G. Oliveira, T. Brox, and W. Burgard. Towards robust semantic segmentation using deep fusion. In *Robotics: Science and Systems Workshop, Are the Sceptics Right? Limits and Potentials of Deep Learning in Robotics*, 2016.

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

[192] A. Valada, R. Mohan, and W. Burgard. Self-supervised model adaptation for multimodal semantic segmentation. *arXiv preprint arXiv:1808.03833*, 2018.

[193] A. Valada, N. Radwan, and W. Burgard. Deep auxiliary learning for visual localization and odometry. In *IEEE International Conference on Robotics & Automation (ICRA)*, 2018.

[194] A. Valada, N. Radwan, and W. Burgard. Incorporating semantic and geometric priors in deep pose regression. In *Proceedings of the Workshop on Learning and Inference in Robotics: Integrating Structure, Priors and Models at Robotics: Science and Systems (RSS)*, 2018.

[195] J. Valentin, M. Nießner, J. Shotton, A. Fitzgibbon, S. Izadi, and P. Torr. Exploiting uncertainty in regression forests for accurate camera relocalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[196] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[197] A. Vemula, K. Muelling, and J. Oh. Social attention: Modeling attention in human crowds. In *IEEE International Conference on Robotics & Automation (ICRA)*, 2018.

[198] O. Vysotska, T. Naseer, L. Spinello, W. Burgard, and C. Stachniss. Efficient and effective matching of image sequences under substantial appearance changes exploiting gps priors. In *IEEE International Conference on Robotics & Automation (ICRA)*, 2015.

[199] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-based localization using lstms for structured feature correlation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[200] J. Wu, L. Ma, and X. Hu. Delving deeper into convolutional neural networks for camera relocalization. In *IEEE International Conference on Robotics & Automation (ICRA)*, May 2017.

[201] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*, 1994.

[202] Z. Wu, C. Shen, and A. van den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *arXiv preprint arXiv:1611.10080*, 2016.

[203] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[204] Z. Yan, T. Duckett, and N. Bellotto. Online learning for human classification in 3d lidar-based tracking. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[205] X. Yin, X. Wang, X. Du, and Q. Chen. Scale recovery for monocular visual odometry using depth estimated with deep convolutional neural fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[206] B. Yu and I. Lane. Multi-task deep learning for image understanding. In *International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, 2014.

[207] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations (ICLR)*, 2016.

[208] A. R. Zamir and M. Shah. Accurate image localization based on google maps street view. In *European Conference on Computer Vision (ECCV)*, 2010.

[209] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.